

# МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ПО ИНФОРМАТИКЕ

**Практикум**

*2006*

УДК 004.3(075.8)  
ББК 32.973я73  
М54

Авторы-составители: преподаватель кафедры информатики и ИТ УО «ВГУ им. П.М. Машерова» **Н.В. Иванова**; старший преподаватель кафедры информатики и ИТ УО «ВГУ им. П.М. Машерова» **А.А. Чиркина**; преподаватель кафедры информатики и ИТ УО «ВГУ им. П.М. Машерова» **В.В. Шедько**

Рецензенты: заведующий кафедрой прикладной математики и механики УО «ВГУ им. П.М. Машерова» кандидат физико-математических наук, доцент Л.В. Маркова; доцент кафедры автоматизации научных исследований УО «ВГУ им. П.М. Машерова» кандидат технических наук В.И. Жидкевич

Практикум и задания разработаны для студентов заочного отделения, обучающихся по специальности «Математика. Информатика». Материалы могут быть использованы в качестве сборника задач для студентов дневного отделения.

УДК 004.3(075.8)  
ББК 32.973я73

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
Часть 1. ВЫЧИСЛИТЕЛЬНЫЙ ПРАКТИКУМ .....	5
Задание № 1. АРИФМЕТИЧЕСКИЕ ВЫРАЖЕНИЯ .....	6
Задание № 2. ВЕТВЛЕНИЯ .....	7
Задание № 3. ЦИКЛЫ И ОДНОМЕРНЫЕ МАССИВЫ .....	10
Задание № 4. ОБРАБОТКА МАТРИЦ .....	12
Задание № 5. ОБРАБОТКА СИМВОЛЬНЫХ И СТРОКОВЫХ ВЕЛИЧИН .....	14
Задание № 6. ВСПОМОГАТЕЛЬНЫЕ АЛГОРИТМЫ, ПРО- ЦЕДУРЫ И ФУНКЦИИ ПОЛЬЗОВАТЕЛЯ .....	17
Список рекомендуемой литературы .....	17
 2. ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ .....	 18
Структурное программирование .....	18
Решение задач с учетом разрядной сетки ЭВМ .....	28
Методы сортировки .....	32
Моделирование работы исполнителей .....	41
Программирование игровых задач .....	53
Перколяция (просачивание) .....	57
Исследовательские задачи .....	59
Литература .....	67
 3. ПРАКТИКУМ ПО РЕШЕНИЮ ЗАДАЧ В MS EXCEL .....	 68
Составление отчетов, консолидация данных и построение сводных таблиц .....	68
Сортировка, фильтрация, реализация запросов к базе данных с использованием функций категории работы с базой данных ....	72
Построение диаграмм типа «Поверхность» и анимированных графиков .....	80
Создание тестирующей программы .....	84
Решение систем линейных алгебраических уравнений .....	87
Задания для самостоятельной работы .....	103
Литература .....	109
Приложение .....	110

## ВВЕДЕНИЕ

Настоящий практикум разработан для студентов математического факультета специальности «Математика. Информатика». Практикум состоит из трех разделов: вычислительный практикум, практикумы по программированию и по решению задач в MS Excel.

Раздел «Вычислительный практикум» предназначен для формирования структурного стиля программирования и обработки навыков составления программ в среде Turbo Pascal, учебных средах КуМир ИнтАл 2.0. Решение задач этого раздела предполагает знание базовых алгоритмических конструкций и наличие начальных навыков программирования.

Раздел «Практикум по программированию» способствует более углубленному изучению возможностей и ограничений языков программирования высокого уровня и представления данных в ЭВМ. В раздел включены исследовательские и игровые задачи, задачи на моделирование, способствующие развитию творческих способностей в программировании.

В качестве отчета по решению задач из этих разделов студент должен предоставить: описание математической модели решения задачи, листинги программ с комментариями, несколько текстовых вариантов расчетов и их анализ.

Наличие различных по сложности заданий позволяет преподавателю организовать процесс обучения с учетом индивидуальных особенностей студентов.

Раздел «Практикум по решению задач в MS Excel» предназначен для формирования навыков использования компьютерных технологий в решении задач прикладной направленности. Выполнение заданий этого раздела предполагает наличие опыта работы с электронной таблицей пакета MS Office. Отчет по результатам выполнения заданий представляется в электронном виде и должен включить: описание выбранного метода решения, расчетную таблицу с формулами, таблицу с результатами вычислений и их графическую интерпретацию.

Задания практикума могут быть использованы при самостоятельной подготовке студентов.

# 1. ВЫЧИСЛИТЕЛЬНЫЙ ПРАКТИКУМ

## Порядок выполнения работы

1. Ознакомиться с необходимым теоретическим материалом – см. **Список рекомендуемой литературы** в конце этого раздела.
2. Изучить приложения, необходимые для выполнения работы.
3. Ознакомиться с заданием своего варианта работы.
4. Выбрать уровень сложности, соответствующий степени вашей подготовленности (или по усмотрению преподавателя).
5. Подготовить алгоритмы решения заданий своего варианта (в соответствии с уровнем сложности).
6. Записать алгоритмы в виде текстов программ в указанной среде (средах) программирования.
7. Набрать тексты программ, отладить их и сохранить в файлах.
8. Протестировать программы и, если есть ошибки, то исправить их. Получить результат. Проанализировать его.
9. Подготовить отчет.

### Каждый отчет должен содержать следующее:

1. Информационный блок (ФИО, группа, факультет, номер варианта, уровень сложности).
2. Тексты заданий варианта, постановки задач и описания программ (описание логической структуры программы; описание алгоритма решения задачи; описание входных и выходных данных программы; описание подпрограмм; перечень исключений и допустимых значений входных данных; тестовые примеры и результаты расчетов по ним).
3. Листинги программ на заданных языках или в заданных средах (Word, Excel, MathCad, PASCAL, Кумир или ИНТАЛ).

Отчет должен быть представлен в машинописном виде в Word, в который должны быть скопированы результаты из других сред.

### Уровни сложности (дополнительные требования) для заданий 1-4:

1. Ввод исходных данных корректен.
2. Ввод исходных данных корректен и предусмотрен форматированный вывод результата, где это возможно.
3. В программе должна быть предусмотрена проверка входных и выходных данных на корректность и форматированный вывод результата в виде отдельной процедуры или функции, где это возможно.

### Задание № 1. АРИФМЕТИЧЕСКИЕ ВЫРАЖЕНИЯ

Задание должно быть выполнено в средах программирования:

- Word (отчет и блок-схема);
- Excel;
- MathCad;
- PASCAL;
- ИНТАЛ.

Вычислить заданное арифметическое выражение для данных типов: вещественные (**REAL** переменные **a, b**) и целые (**INTEGER**, остальные переменные).

#### Варианты индивидуальных заданий

$$1. \frac{2c - d\sqrt{42}}{c + a - 1}$$

$$2. \frac{4 \cdot \lg(c) - \frac{d}{2} + 23}{a^2 - 1}$$

$$3. \frac{2c + \lg(d) \cdot 51}{d - a - 1}$$

$$4. \frac{\frac{\operatorname{arctg}(2c)}{d} + 2}{d - a^2 - 1}$$

$$5. \frac{\sqrt{\frac{53}{a}} + d - 4a}{1 + ab}$$

$$6. \frac{\lg(4a - 1) + \frac{b}{2}}{bc - 5}$$

$$7. \frac{\sqrt{\frac{25}{c}} - d + 2}{d + a^2 - 1}$$

$$8. \frac{c \cdot \operatorname{tg}(b + 23)}{\frac{a}{2} - 4d - 1}$$

$$9. \frac{2c + \lg\left(\frac{d}{4}\right) + 23}{a^2 - 1}$$

$$10. \frac{\operatorname{arctg}\left(\frac{12}{c}\right) + 73}{a^2 - 1}$$

$$11. \frac{\sqrt{15a + b} - \frac{a}{4}}{ba - 1}$$

$$12. \frac{8\lg(b + 1) - c}{\frac{a}{2} + bc}$$

$$13. \frac{\operatorname{arctg}\left(c - \frac{d}{2}\right)}{2a - 1}$$

$$14. \frac{\frac{c}{d} + \ln\left(\frac{3a}{2}\right)}{c - a + 1}$$

$$15. \frac{42c - \frac{d}{2} + 1}{a^2 - \ln(b - 5)}$$

$$16. \frac{\frac{2c}{a} - d^2}{d + \operatorname{tg}(a - 1)}$$

$$17. \frac{-\frac{25}{a} + c - \operatorname{tg}(b)}{1 + \frac{cb}{2}}$$

$$18. \frac{4a - \ln(b - 1)}{\frac{c}{b} + 18a}$$

$$19. \frac{\frac{\operatorname{arctg}(4b)}{c} - 1}{12c + a - b}$$

$$20. \frac{\operatorname{arctg}(b) + cb - \frac{a}{4}}{ab - 1}$$

$$21. \frac{a + \frac{c}{b} - \sqrt{28}}{4ba + 1}$$

$$22. \frac{\frac{c}{b} - \sqrt{24} + a}{2ac - 1}$$

$$23. \frac{2b - \ln(a+b) \cdot c}{\frac{c}{4} - 1}$$

$$24. \frac{41 - \frac{d}{4} - 1}{\frac{1}{\operatorname{tg}(b+a)} - d}$$

$$25. \frac{a - 4b - 1}{\frac{c}{31} + \operatorname{tg}(ab)}$$

$$26. \frac{\lg\left(\frac{b}{a} + 4\right) \cdot c}{41c - b + 1}$$

$$27. \frac{\lg(21-a) \cdot \frac{c}{4}}{1 + \frac{c}{a} + b}$$

$$28. \frac{c - \frac{\ln(33+b)}{4}}{\frac{ac}{b} - 1}$$

$$29. \frac{2b - 38c}{\operatorname{arctg}(b+a)} + 1$$

$$30. \frac{\operatorname{arctg}\left(\frac{c}{4} + 28\right) \cdot d}{\frac{a}{d} - c - 1}$$

$$31. \frac{\frac{ab}{4} - 1}{\sqrt{41 - b \cdot a} + c}$$

$$32. \frac{1 + a - \frac{b}{2}}{b \cdot \sqrt{24} + \frac{a}{b}}$$

$$33. \frac{\ln(ab+2) \cdot c}{41 - \frac{b}{c} + 1}$$

$$34. \frac{\lg(4b-c) \cdot a}{b + \frac{c}{28} - 1}$$

$$35. \frac{2c + \operatorname{tg}(a-21)}{\frac{cb}{a} + 1}$$

$$36. \frac{\frac{4}{c} + \operatorname{tg}(3a)}{\frac{c}{a} - b - 1}$$

$$37. \frac{81\lg(b-1) - c}{2a + \frac{b}{c}}$$

$$38. \frac{4\ln\left(\frac{a}{b}\right) + 1}{cb - 18 + a}$$

$$39. \frac{\frac{4\ln(b)}{c} + 1}{2c + ac - b}$$

$$40. \frac{\frac{\operatorname{arctg}(b-c)}{b} + \frac{a}{4}}{ab - 1}$$

$$41. \frac{\operatorname{arctg}(a-c) \cdot b + 28}{\frac{4b}{a} + 1}$$

### Задание № 2. ВЕТВЛЕНИЯ

Задание должно быть выполнено в средах программирования:

- Word (отчет и блок-схема);
- Excel;
- MathCad;
- PASCAL;
- КуМмир или ИНТАЛ.

Вычислите заданное целочисленное выражение для данных **a**, **b** целого типа, используя операторы ветвлений. Результат **X** – тоже целочисленный.

### Варианты индивидуальных заданий

$$1. X = \begin{cases} \frac{b}{a} + 1, & \text{если } a > b; \\ -44, & \text{если } a = b; \\ \frac{a^3 + b}{b}, & \text{если } a < b; \end{cases}$$

$$2. X = \begin{cases} \frac{a}{b} + 2, & \text{если } a > b; \\ -100, & \text{если } a = b; \\ \frac{b-100}{a}, & \text{если } a < b; \end{cases}$$

$$3. X = \begin{cases} \frac{a}{b} + 201, & \text{если } a > b; \\ -800, & \text{если } a = b; \\ \frac{a-5}{2}, & \text{если } a < b; \end{cases}$$

$$4. X = \begin{cases} \frac{a}{b} + 1, & \text{если } a > b; \\ -100, & \text{если } a = b; \\ \frac{ab-9}{3}, & \text{если } a < b; \end{cases}$$

$$5. X = \begin{cases} \frac{a}{b} + 1, & \text{если } a > b; \\ -425, & \text{если } a = b; \\ \frac{3a-b}{a}, & \text{если } a < b; \end{cases}$$

$$6. X = \begin{cases} \frac{b^2 - a}{2}, & \text{если } a < b; \\ -b, & \text{если } a = b; \\ \frac{ab-8}{a}, & \text{если } a > b; \end{cases}$$

$$7. X = \begin{cases} ab+5, & \text{если } a < b; \\ -b, & \text{если } a = b; \\ \frac{b-9}{a}, & \text{если } a > b; \end{cases}$$

$$8. X = \begin{cases} ab-1, & \text{если } a > b; \\ 255, & \text{если } a = b; \\ \frac{a-5}{b}, & \text{если } a < b; \end{cases}$$

$$9. X = \begin{cases} \frac{a}{b} + 1, & \text{если } a > b; \\ 2a, & \text{если } a = b; \\ \frac{a-10}{b}, & \text{если } a < b; \end{cases}$$

$$10. X = \begin{cases} ab+8, & \text{если } a > b; \\ 455, & \text{если } a = b; \\ \frac{a-b}{b}, & \text{если } a < b; \end{cases}$$

$$11. X = \begin{cases} \frac{b}{a} + 1, & \text{если } a > b; \\ -271, & \text{если } a = b; \\ \frac{a-b}{b}, & \text{если } a < b; \end{cases}$$

$$12. X = \begin{cases} \frac{a}{b} - 37, & \text{если } a > b; \\ 3, & \text{если } a = b; \\ \frac{a^3 - b}{a}, & \text{если } a < b; \end{cases}$$

$$13. X = \begin{cases} ab-5, & \text{если } a > b; \\ -195, & \text{если } a = b; \\ \frac{a-6}{b}, & \text{если } a < b; \end{cases}$$

$$14. X = \begin{cases} \frac{a}{b} - 7, & \text{если } a > b; \\ -195, & \text{если } a = b; \\ \frac{a^3 + 9}{a}, & \text{если } a < b; \end{cases}$$

$$15. X = \begin{cases} a + \frac{b}{4}, & \text{если } a < b; \\ -155, & \text{если } a = b; \\ \frac{b^3 - 5}{a}, & \text{если } a > b; \end{cases}$$

$$16. X = \begin{cases} \frac{a}{b} - 42, & \text{если } a < b; \\ -11, & \text{если } a = b; \\ \frac{a^3 - 8}{a}, & \text{если } a > b; \end{cases}$$

$$17. X = \begin{cases} \frac{b}{a} - 141, & \text{если } a < b; \\ -131, & \text{если } a = b; \\ \frac{3a-9}{a}, & \text{если } a > b; \end{cases}$$

$$18. X = \begin{cases} 5a+b, & \text{если } a > b; \\ -125, & \text{если } a = b; \\ \frac{a-5}{b}, & \text{если } a < b; \end{cases}$$

$$19. X = \begin{cases} ba+1, & \text{если } a > b; \\ -10, & \text{если } a = b; \\ \frac{a-5}{b}, & \text{если } a < b; \end{cases}$$

$$20. X = \begin{cases} \frac{a}{b} + 31, & \text{если } a > b; \\ -25, & \text{если } a = b; \\ \frac{5a-1}{a}, & \text{если } a < b; \end{cases}$$

$$21. X = \begin{cases} 2a+b, & \text{если } a > b; \\ -2, & \text{если } a = b; \\ \frac{a-5}{b}, & \text{если } a < b; \end{cases}$$



$$22. X = \begin{cases} \frac{b}{a} + 1, & \text{если } a < b; \\ 25, & \text{если } a = b; \\ \frac{a^3 - 5}{b}, & \text{если } a > b; \end{cases}$$

$$23. X = \begin{cases} \frac{b}{a} + 61, & \text{если } a > b; \\ -5, & \text{если } a = b; \\ \frac{b-a}{b}, & \text{если } a < b; \end{cases}$$

$$24. X = \begin{cases} \frac{a}{b} + 1, & \text{если } a > b; \\ -2, & \text{если } a = b; \\ \frac{a-b}{a}, & \text{если } a < b; \end{cases}$$

$$25. X = \begin{cases} \frac{3a-5}{b}, & \text{если } a < b; \\ -4, & \text{если } a = b; \\ \frac{a^3 + b}{a}, & \text{если } a > b; \end{cases}$$

$$26. X = \begin{cases} \frac{b}{a} - 1, & \text{если } a < b; \\ -295, & \text{если } a = b; \\ \frac{a-235}{b}, & \text{если } a > b; \end{cases}$$

$$27. X = \begin{cases} \frac{a}{b} + 5, & \text{если } a < b; \\ -5, & \text{если } a = b; \\ \frac{a^2 - b}{b}, & \text{если } a > b; \end{cases}$$

$$28. X = \begin{cases} \frac{a}{b} + 1, & \text{если } a < b; \\ -1, & \text{если } a = b; \\ \frac{ab-5}{a}, & \text{если } a > b; \end{cases}$$

$$29. X = \begin{cases} \frac{a}{b} - 1, & \text{если } a > b; \\ -25, & \text{если } a = b; \\ \frac{a^3 - 5}{a}, & \text{если } a < b; \end{cases}$$

$$30. X = \begin{cases} ab + 21, & \text{если } a > b; \\ -21, & \text{если } a = b; \\ \frac{3a}{b} + 1, & \text{если } a < b; \end{cases}$$

$$31. X = \begin{cases} ab + 1, & \text{если } a > b; \\ -b, & \text{если } a = b; \\ \frac{a-5}{b}, & \text{если } a < b; \end{cases}$$

$$32. X = \begin{cases} \frac{a}{b} - 1, & \text{если } a < b; \\ 25 - a, & \text{если } a = b; \\ \frac{b-5}{a}, & \text{если } a > b; \end{cases}$$

$$33. X = \begin{cases} \frac{b}{a} + 2, & \text{если } a > b; \\ -11, & \text{если } a = b; \\ \frac{a-8}{b}, & \text{если } a < b; \end{cases}$$

$$34. X = \begin{cases} \frac{a}{b} + 2, & \text{если } a > b; \\ 8, & \text{если } a = b; \\ \frac{b-9}{a}, & \text{если } a < b; \end{cases}$$

$$35. X = \begin{cases} ab + 5, & \text{если } a < b; \\ -5, & \text{если } a = b; \\ \frac{b-a}{b}, & \text{если } a > b; \end{cases}$$

$$36. X = \begin{cases} \frac{a}{b} + 1, & \text{если } a < b; \\ -71, & \text{если } a = b; \\ \frac{a-b}{a}, & \text{если } a > b; \end{cases}$$

$$37. X = \begin{cases} \frac{b}{a} - 7, & \text{если } a > b; \\ 43, & \text{если } a = b; \\ \frac{a^3 - b}{b}, & \text{если } a < b; \end{cases}$$

$$38. X = \begin{cases} -5 + \frac{a}{b}, & \text{если } a > b; \\ 45, & \text{если } a = b; \\ \frac{3a-6}{b}, & \text{если } a < b; \end{cases}$$

$$39. X = \begin{cases} \frac{a}{b} + 7, & \text{если } a > b; \\ -125, & \text{если } a = b; \\ \frac{3b+9}{a}, & \text{если } a < b; \end{cases}$$

$$40. X = \begin{cases} \frac{ab}{4}, & \text{если } a < b; \\ -55, & \text{если } a = b; \\ \frac{b-5}{a}, & \text{если } a > b; \end{cases}$$

$$41. X = \begin{cases} \frac{a}{b} - 1, & \text{если } a > b; \\ -414, & \text{если } a = b; \\ \frac{a^3 - b}{a}, & \text{если } a < b; \end{cases}$$

### Задание № 3. ЦИКЛЫ И ОДНОМЕРНЫЕ МАССИВЫ

Задание должно быть выполнено в средах программирования:

- Word (отчет и блок-схема);
- Excel;
- MathCad;
- PASCAL;
- Кумир или ИНТАЛ.

Задав одномерный массив целочисленных данных **A** в одном из заданных типов: **BYTE**, **WORD**, **SHORTINT**, **INTEGER** или **LONGINT**, сделать обработку массива, как указано в варианте. Длина массива  $N \leq 20$ . Исходные данные задать самостоятельно, учитывая формат элементов массива **A**. В программе должны быть предусмотрены процедуры ввода-вывода элементов массива **A** и его обработки. Тип результата определяется из контекста задачи.

#### Варианты индивидуальных заданий

Тип **BYTE** варианты: 2, 7, 12, 17, 22, 27, 32, 37.

Тип **WORD** варианты: 3, 8, 13, 18, 23, 28, 33, 38.

Тип **SHORTINT** варианты: 4, 9, 14, 19, 24, 29, 34, 39.

Тип **INTEGER** варианты: 5, 10, 15, 20, 25, 30, 35, 40.

Тип **LONGINT** варианты: 1, 6, 11, 16, 21, 26, 31, 36, 41.

1. Найти, сколько элементов массива  $A=\{a[i]\}$  удовлетворяют условию:  $a[i] \leq i^2$
2. Найти произведение элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $(a[i])^2 \leq i$
3. Найти, сколько элементов массива  $A=\{a[i]\}$  удовлетворяют условию:  $i \leq a[i] \leq i^2$
4. Найти сумму всех элементов массива  $A=\{a[i]\}$ , имеющих четные индексы
5. Найти, сколько элементов массива  $A=\{a[i]\}$  удовлетворяют условию:  $a[i-1] \leq a[i] \leq a[i+1]$
6. Найти сумму квадратов индексов всех положительных элементов массива  $A=\{a[i]\}$
7. Найти среднее арифметическое элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $a[i] \geq c$
8. Найти максимальную сумму индексов, для положительных, отрицательных и нулевых элементов в массиве  $A=\{a[i]\}$
9. Найти сумму квадратов индексов элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $a[i] \geq c$
10. Найти среднее арифметическое индексов последних **L** отрицательных элементов в массиве  $A=\{a[i]\}$

11. Найти сумму первых  $K$  элементов массива  $A=\{a[i]\}$ , имеющих четные индексы
12. Найти максимальное количество одинаковых элементов массива  $A=\{a[i]\}$ , среди имеющих нечетные индексы
13. Найти сумму индексов элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $c \leq a[i] \leq d$
14. Найти среднее арифметическое последних  $L$  положительных элементов в массиве  $A=\{a[i]\}$
15. Найти произведение индексов положительных элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $c \leq a[i] \leq d$
16. Найти произведение индексов отрицательных элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $c \leq a[i] \leq d$
17. Найти сумму индексов кубов всех положительных элементов массива  $A=\{a[i]\}$
18. Найти сумму квадратов индексов всех отрицательных элементов массива  $A=\{a[i]\}$
19. Найти наибольший из квадратов всех положительных элементов массива  $A=\{a[i]\}$
20. Найти наименьший из квадратов всех отрицательных элементов массива  $A=\{a[i]\}$
21. Найти, сколько элементов массива  $A=\{a[i]\}$  удовлетворяют условию:  $c \leq a[i] \leq d$
22. Найти произведение элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $c \leq a[i] \leq d$
23. Найти, сколько отрицательных элементов массива  $A=\{a[i]\}$  удовлетворяют условию:  $c \leq a[i] \leq d$
24. Найти сумму кубов всех отрицательных элементов массива  $A=\{a[i]\}$
25. Найти, сколько положительных элементов массива  $A=\{a[i]\}$  удовлетворяют условию:  $a[i-1] \leq a[i] \leq a[i+1]$
26. Найти сумму квадратов всех положительных элементов массива  $A=\{a[i]\}$
27. Найти произведение квадратов элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $a[i] \geq c$
28. Найти, сколько положительных, отрицательных и нулевых элементов в массиве  $A=\{a[i]\}$
29. Найти произведение квадратов элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $a[i] \geq c$
30. Найти произведение последних  $L$  отрицательных элементов в массиве  $A=\{a[i]\}$
31. Найти сумму первых  $K$  элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $c \leq a[i] \leq d$
32. Найти количество одинаковых элементов массива  $A=\{a[i]\}$

33. Найти сумму элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $c \leq a[i] \leq d$
34. Найти произведение последних  $L$  положительных элементов в массиве  $A=\{a[i]\}$
35. Найти произведение положительных элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $c \leq a[i] \leq d$
36. Найти произведение отрицательных элементов массива  $A=\{a[i]\}$ , удовлетворяющих условию:  $c \leq a[i] \leq d$
37. Найти сумму кубов всех положительных элементов массива  $A=\{a[i]\}$
38. Найти сумму квадратов всех отрицательных элементов массива  $A=\{a[i]\}$
39. Найти произведение квадратов всех положительных элементов массива  $A=\{a[i]\}$
40. Найти произведение квадратов всех отрицательных элементов массива  $A=\{a[i]\}$
41. Найти сумму первых  $K$  отрицательных элементов массива  $A=\{a[i]\}$

#### **Задание № 4. ОБРАБОТКА МАТРИЦ**

Задание должно быть выполнено в средах программирования:

- Word (отчет и блок-схема);
- Excel;
- MathCad;
- PASCAL;
- КуМир или ИНТАЛ.

Задать квадратную матрицу  $A$  размером  $N \times N$  ( $N \leq 10$ ), состоящую из действительных элементов, реализовать ее обработку, как указано в варианте. Исходные данные задать самостоятельно, учитывая специфику задачи. Исходные данные можно задать заполнением матрицы случайными числами, предусмотрев ее вывод до обработки.

#### **Варианты индивидуальных заданий**

1. Сформировать вектор из суммы элементов строк и найти их среднее арифметическое.
2. Сформировать вектор из суммы элементов столбцов и найти их среднее арифметическое.
3. Сформировать вектор из произведения элементов строк и найти их среднее арифметическое.
4. Сформировать вектор из произведения элементов столбцов и найти их среднее арифметическое.
5. Сформировать вектор из наименьших значений элементов строк и найти их среднее арифметическое.
6. Сформировать вектор из наименьших значений элементов столбцов и найти их среднее арифметическое.

7. Сформировать вектор из наибольших значений элементов строк и найти их среднее арифметическое.
8. Сформировать вектор из наибольших значений элементов столбцов и найти их среднее арифметическое.
9. Сформировать вектор из разностей наибольших и наименьших значений элементов строк.
10. Сформировать вектор из разностей наибольших и наименьших значений элементов столбцов.
11. Сформировать вектор из суммы наибольших и наименьших значений элементов строк.
12. Сформировать вектор из суммы наибольших и наименьших значений элементов столбцов.
13. Найти сумму элементов строки, в которой расположен наименьший элемент.
14. Найти сумму элементов строки, в которой расположен наибольший элемент.
15. Найти сумму элементов столбца, в котором расположен наименьший элемент.
16. Найти среднее арифметическое элементов каждого из четных столбцов этой матрицы.
17. Найти среднее арифметическое элементов каждого из нечетных столбцов этой матрицы.
18. Найти среднее арифметическое элементов каждой из строк матрицы.
19. Найти среднее арифметическое элементов каждой из четных строк этой матрицы.
20. Найти среднее арифметическое элементов каждой из нечетных строк этой матрицы.
21. Найти среднее арифметическое из всех отрицательных элементов этой матрицы
22. Найти среднее арифметическое из всех положительных элементов этой матрицы.
23. Найти характеристику каждой ее строки (сумму положительных четных элементов в каждой строке).
24. Найти характеристику каждого ее столбца (сумму модулей отрицательных нечетных элементов в каждом столбце).
25. Найти сумму и произведение всех ее положительных элементов.
26. Найти сумму и произведение всех ее отрицательных элементов.
27. Найти сумму всех ее положительных и произведение всех ее отрицательных элементов.
28. Найти сумму всех ее отрицательных и произведение всех ее положительных элементов.
29. Найти сумму всех ее элементов и заменить ею все диагональные элементы этой матрицы.
30. Найти произведение всех ее элементов и заменить им все диагональные элементы этой матрицы.

31. Найти сумму всех ее положительных элементов и заменить ею все диагональные элементы этой матрицы.
32. Найти произведение всех ее положительных элементов и заменить им все диагональные элементы этой матрицы.
33. Получить новую матрицу путем деления всех элементов данной матрицы на ее наибольший по модулю элемент.
34. Получить новую матрицу путем деления всех элементов данной матрицы на ее наименьший по модулю элемент.
35. Получить новую матрицу путем умножения всех элементов данной матрицы на ее наибольший по модулю элемент.
36. Получить новую матрицу путем умножения всех элементов данной матрицы на ее наименьший по модулю элемент.
37. Получить новую матрицу путем сложения всех элементов данной матрицы с ее наибольшим по модулю элементом.
38. Получить новую матрицу путем сложения всех элементов данной матрицы с ее наименьшим по модулю элементом.
39. Получить новую матрицу путем вычитания всех элементов данной матрицы из ее наибольшего по модулю элемента.
40. Получить новую матрицу путем вычитания всех элементов данной матрицы из ее наименьшего по модулю элемента.
41. Заменить нулями все ее элементы, расположенные на главной диагонали и выше нее.

### **Задание № 5. ОБРАБОТКА СИМВОЛЬНЫХ И СТРОКОВЫХ ВЕЛИЧИН**

Задание должно быть выполнено в средах программирования:

- Word (отчет);
- PASCAL;
- ИНТАЛ.

Для вводимой строки длиной до  $N$  символов ( $N \leq 255$ ), состоящей из слов разделенных одним или несколькими пробелами, реализовать ее обработку, как указано в варианте. Уровни сложности (дополнительные требования) для задания 5:

1. Решить для исходной строки, без знаков препинания, набранной буквами одного алфавита, одного регистра.
2. Решить для исходной строки, без знаков препинания, набранной буквами одного алфавита, в разных регистрах.
3. Решить для исходной строки, без знаков препинания, набранной буквами разных алфавитов, в разных регистрах.
4. Решить для исходной строки, со знаками препинания, набранной буквами разных алфавитов, в разных регистрах.
5. Решить для исходной строки, состоящей из любых символов.

### Варианты индивидуальных заданий

1. Найти длину слова, состоящего из наибольшего количества различных букв.
2. Найти первое вхождение слова, состоящего из наименьшего количества различных букв.
3. Найти первое вхождение самого длинного слова, начинающееся и заканчивающееся различными буквами.
4. Найти количество слов одинаково читающихся справа налево и наоборот.
5. Найти букву(ы), наиболее часто встречающуюся(иеся) в строке.
6. Найти букву(ы), которая встречается(ются) в наибольшем количестве слов строки.
7. Найти количество слов начинающихся и заканчивающихся одинаковой буквой.
8. Найти первое вхождение самого длинного слова, одинаково читающееся справа налево и наоборот.
9. Найти первое вхождение слова, в котором одна из букв повторяется наибольшее количество раз.
10. Найти первое вхождение слова, состоящего из наибольшего количества различных букв.
11. Найти наибольшую длину слова начинающегося и заканчивающегося одинаковой буквой.
12. Подсчитать количество различных слов в предложении.
13. Подсчитать количество слов в предложении, повторяющихся более одного раза.
14. Найти первое вхождение слова, повторяющегося в строке наибольшее число раз.
15. Определить, является ли данное предложение «перевертышем» после удаления из него всех пробелов.
16. Подсчитать количество пятибуквенных слов в предложении.
17. Подсчитать количество слов в предложении, начинающихся и заканчивающихся одинаковой буквой.
18. Подсчитать наибольшее количество слов в предложении, начинающихся одинаковой буквой.
19. Подсчитать количество слов в предложении, которые начинаются и заканчиваются различными буквами.
20. Подсчитать количество слов в предложении, начинающихся с различных букв.
21. Подсчитать количество слов в предложении, заканчивающихся одинаковой буквой.

22. Найти букву(ы), которой(ыми) заканчивается наибольшее количество слов в строке.
23. Найти букву(ы), которая(ые) наименьшее количество раз встречается(ются) в строке.
24. Найти минимальную длину слова, в котором наибольшее количество гласных букв.
25. Найти первое вхождение слова, в котором наименьшее количество различных согласных букв.
26. Найти первое вхождение самого длинного слова, начинающееся и заканчивающееся согласными буквами.
27. Найти количество слов, содержащих одинаковое количество согласных и гласных букв.
28. Найти букву(ы), с которой наиболее часто начинаются слова в строке.
29. Найти гласную букву(ы), которая(ые) встречается(ются) в наибольшем количестве слов строки.
30. Найти количество слов, начинающихся с гласной и заканчивающихся согласной буквой.
31. Найти последнее вхождение самого длинного слова, состоящего из всех различных букв.
32. Найти первое вхождение слова, в котором каждая из букв повторяется не менее двух раз.
33. Найти первое вхождение слова, состоящего из наименьшего количества различных букв.
34. Найти наибольшую длину слова, начинающегося и заканчивающегося одинаковой согласной буквой.
35. Подсчитать наибольшее количество различных букв в слове в заданном тексте.
36. Подсчитать количество слов в предложении, в которых более одной буквы повторяющейся более одного раза.
37. Подсчитать количество различных букв, с которых начинаются слова в предложении.
38. Подсчитать количество слов в предложении, начинающихся буквой, повторяющейся в этом слове.
39. Подсчитать количество слов в предложении, в которых встречаются идущие подряд одинаковые буквы.
40. Подсчитать сколько раз в строке следующее слово начинается с той же буквы, которой заканчивается предыдущее слово.
41. Подсчитать количество слов в предложении, начинающихся, заканчивающихся и содержащих в середине одну и ту же букву.



### **Задание № 6. ВСПОМОГАТЕЛЬНЫЕ АЛГОРИТМЫ, ПРОЦЕДУРЫ И ФУНКЦИИ ПОЛЬЗОВАТЕЛЯ**

Задание должно быть выполнено в средах программирования:

- Word (отчет);
  - PASCAL;
  - ИНТАЛ.
1. Оформите вычисление значения  $X$  в виде функции пользователя для своего варианта из задания № 2.
  2. Оформите решение для своего варианта из задания № 3 при помощи процедуры (вспомогательного алгоритма) пользователя без аргументов.
  3. Оформите решение для своего варианта из задания № 4 при помощи процедуры (вспомогательного алгоритма) пользователя с параметрами.
  4. Оформите решение для своего варианта из задания № 5 при помощи функции или процедуры пользователя.
  - 5(\*). Оформите решение для своего варианта из задания № 3 или № 5 не используя циклов, заменив их рекурсией.

#### **Литература**

1. Абрамов С.А., Зима Е.В. Начала информатики. – М., 1989.
2. Джонс Ж., Харроу К. Решение задач в системе Турбо Паскаль. – М., 1991.
3. Прайс Д. Программирование на языке Паскаль: Практическое руководство. – М., 1987.
4. Фаронов В.В. Турбо Паскаль 7.0. Начальный курс: Учебное пособие. – М., 1997.
5. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. – М., 1988.
6. Офицеров Д.В. Долгий А.Б. Старых В.А. Программирование на персональных ЭВМ. – Мн., 1993.
7. Пильщиков В.Н. Сборник упражнений по языку Паскаль. – М., 1989.
8. Фаронов В.В. Турбо Паскаль 7.0. Практика программирования: Учебное пособие. – М., 1997.
9. Встроенные справочные системы сред ИНТАЛ, Pascal, MathCad, Excel.

## 2. ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ

### *Структурное программирование*

Структурное программирование – это определенный способ организации программы, который облегчает написание программы, ускоряет его, упрощает отладку программы и открывает возможности для дальнейших модификаций программы.

Возникло структурное программирование в конце пятидесятих годов прошлого века, но лишь в семидесятых годах Э. Дейкстра сумел убедить программистов ведущих фирм в необходимости структурированной организации программ. Основной признак структурированной программы кратко можно описать следующим образом.

Структурированная программа целиком состоит из блок-программы и блоков (подпрограмм), к которым блок-программа обращается. Блок-программа должна быть обозримой, по возможности не должна содержать вычислений и состоит лишь из операторов обращения (вызова) к подпрограммам. Каждый блок имеет единственный вход и единственный выход. Более подробно о структурном программировании можно прочитать, например, в лекциях по программированию, подготовленных А.Л. Брудно и Л.И. Каплан для подготовки участников Московских олимпиад по программированию [3].

В заданиях, предложенных ниже, приведены тексты программ на языке Pascal. Эти программы не отвечают требованиям структурного программирования. Задания:

1. Наберите и выполните программу. Разберитесь, что делает эта программа, сформулируйте условие задачи, решение которой приведено в программе.
2. Составьте новую программу, которая отвечает требованиям структурного программирования и решает задачу, условие которой вы сформулировали в п. 1. Составленная программа должна работать так же, как и программа, текст которой приведен в задании.

#### **Задание 1.**

```

program v1;
uses crt;
label m1,m2,m3;
var i,n:integer;
a:array[1..10] of real;
BEGIN
clrscr; randomize;
for i:=1 to 10 do
begin
a[i]:=random(20); a[i]:=a[i]-10
end;

```

```

for i:=1 to 10 do write(a[i]:8:2);
n:=0;
for i:=1 to 10 do
  begin
    if a[i]<0 then begin a[i]:=2*a[i]; n:=n+1 end
      else if a[i]=0 then goto m1
        else if a[i]<5 then begin a[i]:=1/a[i]; n:=n+1 end
          else if a[i]<10 then goto m1;
    m2: if n=2 then goto m3;
    end;
m1: a[i]:=333; n:=n+1; goto m2;
m3: for i:=1 to 10 do write(a[i]:8:2);
END.

```

### Задание 2.

```

program v2;
uses crt;
label m1,m2,m3;
var i,n,p:integer;
a:array[1..20] of integer;
BEGIN
clrscr; randomize;
for i:=1 to 20 do
  begin
    a[i]:=random(20); a[i]:=a[i]-10;
    write(a[i]:8) end;
n:=0; p:=1;
for i:=1 to 20 do begin
  if (a[i]>=-3) and (a[i]<=-1) then goto m1
    else if (a[i]>=1) and (a[i]<=3) then goto m1
      else goto m2;
m1:p:=p*a[i];
n:=n+1;
if n=5 then goto m3;
m2: end;
m3: writeln(p);
END.

```

### Задание 3.

```

program v3;
uses crt;
label m1;
const n=4;
var i,j,s,k:integer;
a:array[1..20] of integer;
b:array[1..5] of real;
BEGIN
clrscr; randomize;
for i:=1 to 20 do begin
  a[i]:=random(300);
  a[i]:=a[i]-100

```

```

end;
k:=1;
for i:=1 to 20 do begin
  s:=0;
  for j:=i to i+n do s:=s+a[i];
  b[k]:=s/n;
  if b[k]<0 then goto m1;
  i:=i+n; k:=k+1;
end;
m1: for i:=1 to k-1 do write(b[i]:8:2);
END.

```

#### Задание 4.

```

program v4;
uses crt;
label m1,m2,m3;
var i,n,p:integer;
a:array[1..20] of integer;
BEGIN
clrscr; randomize;
for i:=1 to 20 do
  begin
  a[i]:=random(15); a[i]:=a[i]-10;
  write(a[i]:8)
  end;
n:=0;
for i:=1 to 20 do
  begin
  if a[i]>-2 then if a[i]<0 then goto m1;
  if a[i]>0 then if a[i]<2 then goto m1;
m2: end;
goto m3;
m1: n:=n+1;
if n=3 then begin writeln('Да, имеется');
  goto m3 end;
goto m2;
m3: END.

```

#### Задание 5.

```

program v5;
uses crt;
label m1,m2,m3;
const a:array[1..10] of real=(2, 0.25, 0.5, 3, 8, 0.125, 7, 4,
0.5, 1);
var i,j,n:integer;
BEGIN
clrscr;
for i:=1 to 9 do begin
for j:=i to 10 do begin
  if a[i]*a[j]=1 then n:=n+1;
  if n=4 then goto m1;

```

```

end; end;
writeln('Нет, не имеется');
goto m2;
m1: writeln('Да, имеется');
m2: END.

```

### Задание 6.

```

program v6;
uses crt;
label m1,m2,m3;
const a:array[1..10] of real=(2, 0.25, 0.5, 3, 8, 0.125, 7, 4,
0.5, 1);
var i,j,n:integer;
BEGIN
clrscr;
for i:=1 to 9 do begin
for j:=i to 10 do begin
if a[i]*a[j]=1 then n:=n+1;
if n=4 then goto m1;
end; end;
writeln('Нет, не имеется');
goto m2;
m1: writeln('Да, имеется');
m2: END.

```

### Задание 7.

```

program v7;
uses crt;
label m1,m2,m3;
const a:array[1..10] of real=(2, 0.25, 0.5, 3, 8, 0.125, 7, 4,
0.5, 1);
var i,j,n:integer;
BEGIN
clrscr;
for i:=1 to 9 do begin
for j:=i to 10 do begin
if a[i]*a[j]=1 then n:=n+1;
if n=4 then goto m1;
end; end;
writeln('Нет, не имеется');
goto m2;
m1: writeln('Да, имеется');
m2: END.

```

### Задание 8.

```

program v8;
uses crt;
label m1,m2,m3;
const a:array[1..10] of real=(2, 0.25, 0.5, 3, 8, 0.125, 7, 4,
0.5, 1);
var i,j,n:integer;

```

```

BEGIN
clrscr;
for i:=1 to 9 do begin
for j:=i to 10 do begin
  if a[i]*a[j]=1 then n:=n+1;
  if n=4 then goto m1;
end; end;
writeln('Нет, не имеется');
goto m2;
m1: writeln('Да, имеется');
m2: END.

```

### Задание 9.

```

program v9;
uses crt;
label m1,m2,m3;
const a:array[1..10] of real=(2, 0.25, 0.5, 3, 8, 0.125, 7, 4,
0.5, 1);
var i,j,n:integer;
BEGIN
clrscr;
for i:=1 to 9 do begin
for j:=i to 10 do begin
  if a[i]*a[j]=1 then n:=n+1;
  if n=4 then goto m1;
end; end;
writeln('Нет, не имеется');
goto m2;
m1: writeln('Да, имеется');
m2: END.

```

### Задание 10.

```

program v10;
uses crt;
label m1,m2,m3;
const a:array[1..10] of real=(2, 0.25, 0.5, 3, 8, 0.125, 7, 4,
0.5, 1);
var i,j,n:integer;
BEGIN
clrscr;
for i:=1 to 9 do begin
for j:=i to 10 do begin
  if a[i]*a[j]=1 then n:=n+1;
  if n=4 then goto m1;
end; end;
writeln('Нет, не имеется');
goto m2;
m1: writeln('Да, имеется');
m2: END.

```

**Задание 11.**

```

program v11;
const n = 15;
a : array [1..n] of integer = (-12, 27, 8, 19, 22, -9, 19, 0,
-2, 21, -17, 24, 4, 0, 18);
label m1, m2, m3;
var i : 1..n;
    m : integer;
    k : boolean;
BEGIN
    k:=false; goto m1;
m2: for i:=1 to n do
    if i=1 then m:=a[i] else
        if a[i]>m then m:=a[i];
    for i:=1 to n do
        if a[i]<0 then a[i]:=-a[i] else
            if (a[i]<>0) and not(odd(a[i])) then a[i]:=a[i]-m else
                if (a[i]<>0) and odd(a[i]) then a[i]:=a[i] div m;
        k:=true;
m1:  for i:=1 to n do
        write(a[i]:5);
        writeln;
        if k then goto m3 else goto m2;
m3: END.

```

**Задание 12.**

```

program v12;
const n = 12;
a : array [1..n] of integer = (18, -9, 27, -25, -3, 0, -2,
40, 27, 32, 3, -24);
label m1, m2, m3;
var i : 1..n;
    m : integer;
    f : boolean;
BEGIN
    f:=false; goto m3;
m2: for i:=n downto 1 do
    if i=n then m:=a[i] else
        if a[i]<m then m:=a[i];
    i:=n; f:=true;
    while (i>0) and f do
        if a[i]=m then f:=false else begin
            a[i]:=a[i]+m;
            i:=i-1 end;
        f:=true;
m3: for i:=1 to n do
    write(a[i]:5);
    writeln;
    if f then goto m1 else goto m2;
m1: END.

```

**Задание 13.**

```

program v13;
uses crt;
label m1, m2, m3;
const n = 12;
var a, b, c : array [1..n] of integer;
    i : 1..n;
    f : boolean;
BEGIN
  for i:=1 to n do begin
    write('a[' ,i, ']= '); readln(a[i]);
    write('b[' ,i, ']= '); readln(b[i]);
  end;
  writeln; f:=true; goto m2;
m1: for i:=1 to n do
  if (a[i]>b[i]) and (b[i]<>0) and ((a[i] div b[i]) mod 2=0)
then c[i]:=a[i] div b[i] mod 2 else
  if (a[i]>b[i]) and (b[i]<>0) and ((a[i] div b[i]) mod
2=1) then c[i]:=a[i]+b[i] else
  if (a[i]<b[i]) and (a[i]<>0) and ((b[i] div a[i]) mod
2=1) then c[i]:=b[i] div a[i] mod 2 else
  if (a[i]<b[i]) and (a[i]<>0) and ((b[i] div a[i])
mod 2=0) then c[i]:=a[i]*b[i] else
  if (a[i]<>0) and (b[i]<>0) then c[i]:=a[i] else
c[i]:=0;
  f:=not(f); clrscr;
m2: for i:=1 to n do write(a[i]:5); writeln;
  for i:=1 to n do write(b[i]:5); writeln;
  if not(f) then goto m3 else goto m1;
m3: for i:=1 to n do write(c[i]:5); writeln; readkey
END.

```

**Задание 14.**

```

program v14;
uses crt;
label m1, m2, m3;
const n = 12;
var a, b, h : array [1..n] of integer;
    i : 1..n;
    f : boolean;
BEGIN
  for i:=1 to n do begin
    write('a[' ,i, ']= '); readln(a[i]);
    write('b[' ,i, ']= '); readln(b[i]);
  end;
  writeln; f:=i<>0; clrscr;
  for i:=1 to n do write(a[i]:5); writeln;
  for i:=1 to n do write(b[i]:5); writeln;
  if not(f) then goto m3 else goto m1;
m3: for i:=1 to n do write(h[i]:5); writeln; readkey; goto m2;

```



```

m1: for i:=1 to n do
    if (a[i]>b[i]) and (b[i]<>0) and ((a[i] mod b[i])<>0) then
h[i]:=a[i] mod b[i] else
    if (a[i]>b[i]) and (b[i]<>0) and ((a[i] mod b[i])=0)
then h[i]:=a[i] div b[i] else
    if (a[i]<b[i]) and (a[i]<>0) and ((b[i] mod a[i])<>0)
then h[i]:=b[i] mod a[i] else
    if (a[i]<b[i]) and (a[i]<>0) and ((b[i] mod a[i])=0)
then h[i]:=b[i] div a[i] else
    if (a[i]<>0) and (b[i]<>0) then h[i]:=a[i] else
h[i]:=0;
    goto m3;
m2: END.

```

### Задание 15.

```

program v15;
label m1, m2, m3, m4, m5;
const n=10;
  v : array [1..n] of real = (0.23, 5.07, -12.1, -2.09, 4.28,
-5.0, 11.05, -3.03, 4.17, -0.16);
var i, k : integer;
    os, s : real;
    b : boolean;
BEGIN
  k:=0;
m2: case k of
    0, 3 : goto m1;
    1 : goto m3;
    2 : goto m4;
    4 : goto m5 end;
m4: s:=0;
    for i:=1 to n do s:=sqr(v[i]-os)+s;
    for i:=1 to n do
    if v[i]<0 then v[i]:=s+v[i] else v[i]:=v[i]-s;
    k:=k+1; goto m2;
m3: os:=0;
    for i:=1 to n do os:=os+v[i];
    os:=os/n; k:=k+1; goto m2;
m1: for i:=1 to n do write(v[i]:8:2);
    writeln; k:=k+1; goto m2;
m5: END.

```

### Задание 16.

```

program v16;
label m1, m2, m3, m4, m5;
const n=13;
var a : array [1..n] of boolean;
    i, k, m : byte;
    f : boolean;
BEGIN
  randomize; goto m2;

```

```

m3: i:=1; m:=1; f:=i<n;
m4: if f then goto m5 else begin
      writeln('k= ',m); goto m1 end;
m2: for i:=1 to n do begin
      a[i]:=random>=0.5; write(a[i]:6)
      end;
      writeln; goto m3;
m5: k:=1;
      while (i<n) and (a[i]=a[i+1]) do begin
        k:=k+1; i:=i+1 end;
      i:=i+1;
      if k>m then m:=k; f:=i<n; goto m4;
m1: END.

```

### Задание 17.

```

program v17;
uses crt;
label m1, m2, m3, m4, m5;
const n=20;
type mas = array [1..n] of integer;
var a, b : mas;
    i, k, j : byte;
    h : integer;
    f : boolean;
BEGIN
  randomize; clrscr; goto m5;
m2: if j=h then begin
      k:=k+1; b[k]:=a[i]
      end; goto m3;
m5: for i:=1 to n do begin
      a[i]:=random(50)-25;
      write(a[i]:4) end;
      writeln; k:=0;
      for i:=1 to n do
        if (abs(a[i])>9) and (a[i]<100) then begin
          h:=abs(a[i]); j:=1;
m1: j:=j+1; f:=(j<h) and (h mod j<>0);
          if f then goto m1 else goto m2;
m3: end;
          if k>0 then for i:=1 to k do write(b[i]:4);
          writeln;
END.

```

### Задание 18.

```

program v18;
const n=10;
label m1, m2, m3;
var i, j, k : integer;
    a, b : array [1..n] of integer;
BEGIN
  for i:=1 to n do begin

```

```

m3: writeln('a[' ,i,'] = '); readln(a[i]);
    writeln('b[' ,i,'] = '); readln(b[i]);
    if (a[i]<1) or (b[i]<1) then goto m3
end; k:=1;
m2: for j:=1 to a[k] do begin
    i:=j*b[k] div a[k];
    if i*a[k]=j*b[k] then goto m1
end;
m1: writeln(a[k], ' / ', b[k], ' = ', j, ' / ', i);
    k:=k+1;
    if k<=n then goto m2;
END.

```

### Задание 19.

```

program v19;
label m1, m2;
var k, q : integer; f : boolean;
    a, b, i, j, n : longint;
    r : array [1..10] of longint;
BEGIN
    write('n= '); readln(n); q:=0;
    for i:=1 to n do begin
        k:=0; a:=i*i; b:=i;
m1: b:=b div 10; k:=k+1;
        if b>0 then goto m1; b:=i;
        for j:=1 to k do begin
            if a mod 10 <> b mod 10 then goto m2;
            a:=a div 10; b:= b div 10
        end;
        f:=true; q:=q+1; r[q]:=i*i;
m2: end;
        if f then for k:=1 to q do write(r[k]:10) else write('Нет');
    writeln
END.

```

### Задание 20.

```

program v20;
const n=500;
label m1, m2, m3;
var i, j, k, q, ii : integer;
    m : longint;
    p : array [1..n] of integer;
BEGIN
m3: write('M= '); readln(m);
    if (m>3001) or (m<2) then goto m3;
    if m>=2 then write(2:8);
    if m>=3 then write(3:8);
    k:=1; p[k]:=3; i:=5;
    while i<=m do begin
        for j:=1 to k do begin
            q:=p[j];

```

```

    if sqr(q)>i then goto m1;
    if i mod q = 0 then goto m2 end;
  if k=n then begin
    for ii:=2 to k do write(p[ii]:8); writeln; i:=m+1;
    writeln('Остальные числа не могут быть обработаны
корректно');
    halt end
  else
m1: if k<n then begin
    k:=k+1; p[k]:=i end;
m2: i:=i+2 end;
    if k>1 then for i:=2 to k do write(p[i]:8); writeln
END.

```

### **Решение задач с учетом разрядной сетки ЭВМ**

В программировании часто приходится обрабатывать числа, выходящие за пределы допустимых разрядной сеткой ЭВМ значений. Для решения следующих задач остановимся на рассмотрении целых чисел. Для обработки этих чисел в программе необходимо предусмотреть возможность формирования очередного числа. Если число может быть получено – оно обрабатывается соответствующим образом и программа продолжает работу. Завершение работы программы должно осуществляться корректно, т.е. не должно быть прерываний типа «Error 215: Runtime error». Программа должна соответствовать принципам структурного программирования.

1. По аналогии с определением факториала дадим определение «терминала» для целых неотрицательных чисел  $n$ :  $n? = 1 + 2 + 3 + \dots + n = \sum_{k=1}^n k$ . Это аналог факториала, только здесь мы производим сложение, а не умножение. Написать программу вычисления следующей суммы:  $\sum_{n=1}^M n?$ , где  $M$  – максимально возможное число, при котором данная сумма «умещается» в памяти ЭВМ.
2. Заданы первый элемент и разность арифметической прогрессии, а также следующие условия на входные данные: первый элемент и разность прогрессии может лежать в диапазоне от  $-maxint$  до  $+maxint$ ; число элементов прогрессии должно лежать в диапазоне  $1 \dots 200$ . Выдать результат в виде таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.
3. Заданы первый элемент и знаменатель геометрической прогрессии, а также следующие условия на входные данные: первый элемент и знаменатель прогрессии может лежать в диапазоне от  $-maxint$  до  $+maxint$ ; число элементов прогрессии должно лежать в диапазоне  $1 \dots 200$ . Выдать результат в виде таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.

4. Разработать программу вычисления суммы элементов геометрической прогрессии с заданными первым элементом и знаменателем, причем количество элементов в сумме должно быть максимально возможным, при котором сумма «умещается» в памяти ЭВМ.
5. Составить программу, выводящую сумму натуральных простых чисел вида  $4k + 1$ , и количество слагаемых в этой сумме. Результат вывести в виде линейной таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.
6. Найти все целые числа, которые делятся на 3, 7 и 19. Выдать результат в виде линейной таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.
7. Найти все числа, «умещающиеся» в памяти ЭВМ, которые делятся без остатка на все без исключения целые числа от 1 до 10. Выдать результат в виде линейной таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.
8. Найти произведение следующего вида:  $(a - n)(a - 2n)(a - 3n) \dots$ , вычисления производить до тех пор, пока данное произведение «умещается» в памяти ЭВМ. Значения  $a$  и  $n$  вводятся с клавиатуры. Вывести все полученные числа.
9. Выдать в виде таблицы квадраты натуральных чисел, причем размер таблицы настолько велик, насколько позволяет процессор.
10. Выдать в виде таблицы кубы целых чисел, причем размер таблицы настолько велик, насколько позволяет процессор.
11. Задана последовательность квадратов чисел Фибоначчи. Найти максимальный элемент этой последовательности, «умещающийся» в разрядную сетку ЭВМ.
12. Найти сумму чисел Фибоначчи, причем количество элементов в сумме должно быть максимально возможными, при котором данная сумма «умещается» в памяти ЭВМ.
13. Найти максимальное четное число Фибоначчи, «умещающееся» в памяти ЭВМ. Выводить все найденные четные числа Фибоначчи.
14. Найдите все числа Фибоначчи, «умещающиеся» в разрядную сетку ЭВМ, являющиеся простыми числами. Выводить все найденные простые числа Фибоначчи.
15. Последовательность задана следующим образом:  $a_0 = 0$ ,  $a_1 = 1$ ,  $a_{n+2} = a_{n+1} + a_n + f(n)$ , где  $f(n) = n^5$ . Напишите программу вычисления элементов этой последовательности и выдачи результата в виде таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.

16. Видный французский физик и популяризатор науки М. Мерсенн заметил, что многие простые числа имеют вид  $2^p - 1$ , здесь  $p$  также является простым числом. Все простые числа упомянутого вида называются числами Мерсенна. Составить программу для нахождения всех чисел Мерсенна, «умещающихся» в памяти ЭВМ.
17. Задана последовательность следующего вида:  $x; 2^2 \cdot x^2; 3^2 \cdot x^3; \dots; n^2 \cdot x^n; \dots$ , где  $x$  – целое. Напишите программу вычисления элементов последовательности и выдачи результата в виде таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.
18. Задана последовательность следующего вида:  $1; -x; x^2; -x^3; \dots; (-1)^{n+1} x^{n-1}; \dots$ , где  $x$  – целое. Напишите программу вычисления элементов последовательности и выдачи результата в виде таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.
19. Задана последовательность следующего вида:  $2 \cdot x^2; 4 \cdot x^4; 8 \cdot x^6; \dots; 2^n \cdot x^{2n}; \dots$ , где  $x$  – целое. Напишите программу вычисления элементов последовательности и выдачи результата в виде таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.
20. Задана последовательность следующего вида:  $3x; 3^2 \cdot x^3; 3^3 \cdot x^5; \dots; 3^n \cdot x^{2n-1}; \dots$ , где  $x$  – целое. Напишите программу вычисления элементов последовательности и выдачи результата в виде таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.
21. Задана последовательность следующего вида:  $x; 2^2 \cdot x^3; 2^4 \cdot x^5; 2^6 \cdot x^7; \dots; 2^{2n-2} \cdot x^{2n-1}; \dots$ , где  $x$  – целое. Напишите программу вычисления элементов последовательности и выдачи результата в виде таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.
22. Задана последовательность следующего вида:  $x + 2!x^2 + 3!x^3 + \dots + n!x^n$ , где  $x$  – целое. Напишите программу вычисления элементов последовательности и выдачи результата в виде таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.
23. Автоморфным называется такое число, которое равно последним цифрам своего квадрата. Например:  $5^2 = 25; 25^2 = 625$ . Составить программу для нахождения всех автоморфных чисел, «умещающихся» в памяти ЭВМ.
24. Вывести все «особые» натуральные числа  $k$ , «умещающиеся» в памяти ЭВМ. Число  $k$  считать «особым», если для него существуют натуральные числа  $n$  и  $m$  такие, что  $k = n^3 + m^2$ .
25. Обозначим через  $S(n)$  сумму цифр числа  $n$ . Выведите на экран элементы последовательности  $S(a), S(a^2), S(a^3), \dots$  до тех пор, пока число  $n = a^k$ , ( $a, k \in \mathbb{N}$ ), «умещается» в памяти ЭВМ.

26. Найти все числа, «умещающиеся» в памяти ЭВМ, следующего вида: число, деленное на сумму его цифр равно сумме кубов тех же цифр.
27. Найти все числа, «умещающиеся» в памяти ЭВМ, следующего вида: если из суммы квадратов цифр числа вычесть произведение этих же цифр, то получится опять это число (например,  $(3^2+7^2) - 3 \times 7=37$ ).
28. Составить программу, которая выводит все целые числа  $a, b, c$ , такие, что  $1 < a < b < c$  и число  $(a-1)(b-1)(c-1)$  является делителем числа  $(abc-1)$ . Результат вывести в виде линейной таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.
29. Число, состоящее из  $n$  ( $n > 1$ ) цифр, называется числом Армстронга, если сумма его цифр, возведенных в  $n$ -ю степень, равна самому этому числу. Например:  $153 = 1^3+5^3+3^3$ ;  $1634 = 1^4+6^4+3^4+4^4$ . Составить программу, которая будет находить все  $n$ -значные числа Армстронга, «умещающиеся» в памяти ЭВМ.
30. Иногда возникает необходимость построить такой прямоугольный треугольник, у которого оба катета и гипотенуза выражаются целыми числами. Целые числа, пригодные для этой цели называют пифагоровыми. Найти все пифагоровы тройки чисел, «умещающиеся» в памяти ЭВМ.
31. Составить программу-генератор простых чисел, в основу которого положить формулу:  $\frac{2^{2x+1} + 1}{3}$  при  $1 \leq x \leq 36$ . Найти все простые числа, которые можно получить с помощью этого генератора, «умещающиеся» в памяти ЭВМ.
32. Составить программу-генератор простых чисел, в основу которого положить формулу:  $F = n^2 + n + 41$  при  $n$  кратном 41. Из найденных чисел сформировать таблицу, размер которой настолько велик, насколько позволяет процессор.
33. Найти все элементы последовательности  $(a_n)$ , «умещающиеся» в памяти ЭВМ, если последовательность образована следующим образом:
- $$a_1 = 1,$$
- $$a_n = 1 + \sum_d^{\frac{n}{d}} (-1)^d a_d.$$
- Сумма  $\sum$  берется по всем делителям числа  $n$  (включая  $d=1$  и  $d=n$ ).
34. Между числом 3 и неизвестным числом вставлено еще одно число так, что все три числа образуют арифметическую прогрессию. Если средний член этой прогрессии уменьшить на 6, то получится геометрическая прогрессия. Найдите разность арифметической и знаменатель геометрической прогрессий. Составьте программу, которая вычисляет сумму арифметической прогрессии и выводит в виде линейной таблицы эле-

менты геометрической прогрессии, начиная с  $a_1=3$ , причем размер таблицы настолько велик, насколько позволяет процессор.

35. Если между цифрами числа 16, которое является полным квадратом, вписать число 15, получим 1156 – число, также являющееся полным квадратом. Повторив эти действия с числом 1156, получим 111556 – также полный квадрат. Составить программу, выводящую натуральные числа, квадраты которых получаются описанным выше способом, и их квадраты в виде линейной таблицы, причем размер таблицы настолько велик, насколько позволяет процессор.

### **Методы сортировки**

Задачи упорядочивания (сортировки) данных имеют большое практическое значение и возникают, например, при оформлении статических сводок, справочных материалов. Процесс упорядочивания при больших объемах данных трудоемок, поэтому приходится прибегать к компьютерам. Проблема описания и исследования различных методов сортировки представляет собой классическую проблему программирования. Разработчики программного обеспечения предпочитают хранить информацию в отсортированном виде, чтобы поиск нужных данных можно было осуществлять посредством бинарного или других методов, отличных от полного перебора. В программировании рассматривают сортировку массивов и файлов. При решении задачи сортировки обычно выдвигается требование минимальности использования дополнительной памяти. Из этого, как правило, следует недопустимость применения дополнительных массивов.

Под сортировкой, или упорядочиванием списка объектов будем понимать расположение этих объектов по возрастанию или убыванию согласно определенному линейному отношению порядка, такому как отношение « $\leq$ » для чисел [1].

Существует множество разнообразных методов сортировки. Все методы сортировки можно разделить на две большие группы: прямые методы сортировки и улучшенные методы сортировки. Прямые методы сортировки по принципу, лежащему в основе метода, в свою очередь разделяются на три подгруппы:

- сортировка обменом («пузырьковая» сортировка);
- сортировка выбором;
- сортировка вставками.

Улучшенные методы сортировки основываются на тех же принципах, что и прямые, но используют некоторые оригинальные идеи, ускоряющие процесс сортировки. На практике прямые методы используются редко из-за низкого быстродействия. Однако они хорошо отражают суть основанных на них улучшенных методов. Если длина массива невелика, то неко-



торые из прямых методов могут превзойти улучшенные методы.

При программировании алгоритмов сортировки большое внимание уделяется эффективности самого алгоритма. Выделяют следующие критерии оценки эффективности алгоритма сортировки:

- количество шагов алгоритма, необходимых для упорядочивания  $n$  элементов;
- количество сравнений между значениями элементов, выполняемых при сортировке массива из  $n$  элементов.

Рассмотрим один из методов сортировки обменами – метод пузырька. Принцип метода состоит в следующем. Пусть дан некоторый числовой массив  $X = \{x_1, x_2, \dots, x_n\}$ . Его надо упорядочить по неубыванию значений элементов, т.е. расположить элементы таким образом, чтобы выполнялись условия:  $x_1 \leq x_2 \leq \dots \leq x_n$ .

Будем поочередно слева направо сравнивать пары соседних элементов. Находим такое  $i$ , что  $x_i > x_{i+1}$ . Меняем  $x_i$  и  $x_{i+1}$  местами. Просмотр продолжаем дальше с  $x_{i+1}$  элемента. В результате одного такого прохода наибольшее число передвинется на последнее место (всплыл первый, самый большой, «пузырек»). В результате первого прохода по массиву максимальный элемент становится на свое, последнее, место. Второй проход обменов выполняется до  $n-1$  элемента и т.д. Всего требуется  $n-1$  проход по массиву. Реализацию этого метода можно проиллюстрировать следующим примером. Пусть дан массив  $X = \{2, -7, 0, 18, -16, 5\}$ .

	2	-7	0	18	-16	5
1)	-7	0	2	-16	5	18
2)	-7	0	-16	2	5	
3)	-7	-16	0	2		
4)	-16	-7	0			
5)	-16	-7				

Поскольку в исходном массиве число элементов  $n = 6$ , количество проходов массива  $n - 1 = 5$ . Массив будет упорядочен после просмотра, в котором участвовали только первый и второй элементы.

```

for i:=1 to n-1 do
  for j:=1 to n-i do
    if a[j]>a[j+1] then begin {обмен}
      c:=a[j]; a[j]:=a[j+1]; a[j+1]:=c
    end;
  
```

Недостаток этого метода сортировки: массив может быть уже отсортирован, а процесс прохода по массиву продолжается.

Еще один из прямых методов сортировки – сортировка вставками. Принцип метода состоит в следующем. Массив разделяется на две части: отсор-

тированную и неотсортированную. Элементы из неотсортированной части поочередно выбираются и вставляются в отсортированную часть так, чтобы не нарушить в ней упорядоченность элементов. В начале работы алгоритма в качестве отсортированной части массива принимают только один первый элемент, а в качестве неотсортированной части – все остальные элементы. Таким образом, алгоритм будет состоять из  $n-1$  прохода, каждый из которых будет включать следующие действия:

- взятие очередного  $i$ -го элемента из неотсортированной части и хранение его в дополнительной переменной;
- поиск позиции  $j$  в отсортированной части массива, не нарушающей упорядоченности и сдвиг элементов массива от  $i-1$  до  $j-1$  вправо с целью освобождения найденной позиции вставки;
- вставка взятого элемента в найденную  $j$ -ю позицию.

```

program sort_vsta;
const n=10; a:array [1..n] of integer=(-6, 9, 8, 4, 6, -1, -4, 0, 2, -9);
var k,i,j,c:integer;
begin
  for i:=2 to n do begin
    c:=a[i]; j:=i;
    while (j>1) and (c<a[j-1]) do begin
      a[j]:=a[j-1];
      j:=j-1;
    end;
    a[j]:=c;
  end;
end.

```

Выполнение этой программы можно проиллюстрировать следующим образом.

	-6	9	8	4	6	-1	-4	0	2	-9
1)	-6	9								
2)	-6	8	9							
3)	-6	4	8	9						
4)	-6	4	6	8	9					
5)	-6	-1	4	6	8	9				
6)	-6	-4	-1	4	6	8	9			
7)	-6	-4	-1	0	4	6	8	9		
8)	-6	-4	-1	0	2	4	6	8	9	
9)	-9	-6	-4	-1	0	2	4	6	8	9

Составьте программы, реализующие описанные в заданиях методы сортировок.

### **Задание 1.**

Метод пузырька – лишь один из алгоритмов сортировки обменами. Есть и другие алгоритмы, которые относятся к этой группе сортировок. Вот один из них:

последовательным просмотром чисел  $a_1, a_2, \dots, a_n$  найти наименьшее  $i$  такое, что  $a_i > a_{i+1}$ . Поменять  $a_i > a_{i+1}$  местами и возобновить просмотр с начала массива. Когда не удастся найти такое  $i$ , массив будет упорядочен нужным образом. Написать программу, реализующую данный алгоритм.

Решить задачу неизменным, стандартным методом пузырька.

Исследовать число сравнений и число перемещений (т.е. перестановок элементов с одного места на другое) в процессе применения каждого из реализованных методов сортировки.

В задаче варианта предусмотреть диалог с пользователем и по его выбору организовать сортировку по возрастанию или убыванию элементов.

### **Задание 2.**

Одним из методов сортировки обменами является метод пузырька. Его можно усовершенствовать следующим образом: последовательным просмотром чисел  $a_1, a_2, \dots, a_n$  в границах от 1 до  $n$  (т.е. слева направо) сравниваем между собой пары и переставляем элементы в неупорядоченных парах. Сужаем границы и просматриваем пары в диапазоне от  $n$  до 2 (направление – справа налево), затем от 2 до  $n - 1$ , от  $n - 1$  до 3, и т.д., причем направление просмотра массива каждый раз меняется. Описанный метод называют шейкерной сортировкой.

Написать процедуру, реализующую данный метод.

Решить задачу неизменным, стандартным методом пузырька и реализовать его во второй процедуре.

Исследовать число сравнений и число перемещений (т.е. перестановок элементов с одного места на другое) в процессе применения каждого из реализованных методов сортировки.

В основной программе предусмотреть диалог с пользователем и по его выбору организовать сортировку по возрастанию или убыванию элементов.

### **Задание 3.**

Метод упорядочивания бинарными вставками основан на методе сортировки простыми вставками. Место, на которое надо вставить некоторый элемент  $a_i$ , в уже упорядоченную последовательность  $a_1, a_2, \dots, a_{i-1}$ , определяется методом дихотомии (деления пополам). Принцип бинарного (двоичного) поиска в упорядоченном массиве: исходный массив делится пополам и для сравнения выбирается средний элемент. Если он совпадает с искомым, то поиск заканчивается. Если же средний элемент меньше искомого, то все элементы левее его тоже будут меньшими. Их можно исключить

из дальнейшего поиска, оставив правую часть упорядоченного массива. На втором этапе выполняются аналогичные действия над оставшейся половиной массива. Так продолжается до тех пор, пока или совпадающий элемент будет найден, или длина зоны поиска станет  $= 0$ . В последнем случае искомым элемент найден не будет.

Написать процедуру, реализующую данный метод.

Решить задачу неизменным, стандартным методом простых вставок и оформить решение во второй процедуре.

Исследовать число сравнений и число перемещений (т.е. перестановок элементов с одного места на другое) в процессе применения каждого из реализованных методов сортировки. Доказать, что метод бинарных вставок требует  $n \cdot \log_2 n$  сравнений элементов.

В основной программе предусмотреть диалог с пользователем и по его выбору организовать сортировку по возрастанию или убыванию элементов.

#### **Задание 4.**

Стандартный метод сортировки выбором по неубыванию заключается в следующем: среди всех элементов массива отыскивается максимальный элемент и меняется местами с последним элементом массива; среди всех элементов, кроме последнего (он уже находится на своем окончательном месте), снова отыскивается наибольший и меняется местами с элементом, стоящим на предпоследнем месте. Этот процесс продолжается до тех пор, пока не будет обработано  $n - 1$  элементов, последний элемент останется на своем месте.

Метод сортировки выбором можно изменить следующим образом. За один просмотр массива определяется наименьший и наибольший из них. Наибольший меняется местом с последним элементом массива, наименьший – с первым. Затем границы массива сужаются и просмотр возобновляется для  $n - 2$  элементов (исключая первый и последний).

Составить процедуры сортировки массива по возрастанию описанными методами.

Исследовать число сравнений и число перемещений (т.е. перестановок элементов с одного места на другое) в процессе применения каждого из реализованных методов сортировки.

В программе предусмотреть диалог с пользователем и по его выбору организовать сортировку по возрастанию или убыванию элементов.

#### **Задание 5.**

Улучшение метода, основанного на обмене, приводит к самому лучшему из известных в данный момент методов – методу быстрой сортировки Хоара. Изобретатель Ч. Хоар исходил из того, что для достижения наилучшей эффективности сначала лучше производить перестановки на большие расстояния.

Суть метода. Пусть  $A(k)$  ( $k=1..n$ ) – одномерный массив и  $x$  – какой-нибудь элемент из него. Будем просматривать слева наш массив до тех пор, пока не обнаружим элемент  $a_i > x$ , затем будем просматривать массив справа, пока не встретим  $a_j < x$ . Поменяем эти два элемента и продолжим процесс просмотра и обмена, пока оба просмотра не встретятся где-то в середине массива. В результате массив окажется разбитым на левую часть с  $a_i \leq x$  и правую часть  $a_j \geq x$ . Аналогично поступим с частями AL и AR и т.д., пока каждая из частей не будет содержать ровно один элемент. Легко понять, что в результате получится отсортированный массив.

Составить программу сортировки массива методом Хоара. Исследовать число сравнений и число перемещений (т.е. перестановок элементов с одного места на другое) в процессе применения реализованного метода сортировки.

В программе предусмотреть диалог с пользователем и по его выбору организовать направление сортировки (по возрастанию или убыванию элементов).

#### **Задание 6.**

Сортировка методом Фон-Неймана. Процесс сортировки осуществляется следующим образом. Завести два массива A и B и записать исходные числа в A. Упорядочить пары соседних чисел ( $A_1$  и  $A_2$ ,  $A_3$  и  $A_4$  и т.д.) и записать их в B; взять из B по две соседние упорядоченные пары и, слив их в упорядоченные четверки, снова записать в A; затем соседние четверки слить в упорядоченные восьмерки и т.д. до тех пор, пока весь массив не будет отсортирован. Учесть, что, в конце концов, числа должны оказаться в A.

Исследовать число сравнений и число перемещений (т.е. перестановок элементов с одного места на другое) в процессе применения реализованного метода сортировки.

В программе предусмотреть диалог с пользователем и по его выбору организовать направление сортировки (по возрастанию или убыванию элементов).

#### **Задание 7.**

Задан массив вещественных чисел из  $n$  элементов. Отсортировать массив методом выбора, используя для поиска максимального и минимального элементов следующий алгоритм.

Требуется найти  $\max(a_1, a_2, \dots, a_n)$  и  $\min(a_1, a_2, \dots, a_n)$  следующим образом. Берем два первых элемента и среди них ищем максимальный:  $m1 = \max(a_1, a_2)$ . Затем ищем максимум среди 3-го и 4-го элементов  $m2 = \max(a_3, a_4)$ . Далее находим максимальный из этих двух максимумов:  $m1 = \max(m1, m2)$ . В результате в  $m1$  находится максимальный из первых четырех элементов массива. Берем следующие два элемента, ищем среди них максимальный:  $m2 = \max(a_5, a_6)$  и снова находим  $m1 = \max(m1, m2)$ . Теперь в  $m1$  находится максимальный из первых шести элементов массива. Процесс продолжаем до тех пор, пока не закончится массив и в  $m1$  не ока-

жется максимальный элемент (предусмотреть случай нечетного количества элементов в массиве).

Минимальный элемент ищется аналогично.

Составить процедуру, реализующую данный алгоритм. Используя написанную процедуру, упорядочить элементы массива по возрастанию их значений следующим образом: найденные максимумы перемещать вправо, а минимумы – влево.

Исследовать число сравнений и число перемещений (т.е. перестановок элементов с одного места на другое) в процессе применения реализованного метода сортировки.

### Задание 8.

Отсортировать линейный массив методом Шелла. Суть метода в следующем. Рассмотрим массив из  $n$  элементов. Сначала отдельно группируются и сортируются элементы, отстоящие на расстояние  $[n/2]$ . Каждая из групп будет состоять из 2 элементов. После первого прохода элементы перегруппировываются – теперь каждый элемент группы отстоит от другого на  $[n/4]$  и вновь сортируются и т.д. Каждая  $i$ -я сортировка объединяет две группы, уже отсортированные  $2i$ -сортировкой. Сортировка групп выполняется применением сортировки вставками.

Говоря о сортировке Шелла, мы выбрали простой ее вариант, когда повторные перемещения уменьшаются вдвое по сравнению с предыдущими.

Кнут рекомендовал использовать последовательности:

$$(1) \quad X_n = 3X_{n-1} + 1, \quad X_1 = 1 \quad (1, 4, 13, 40, 121, \dots)$$

$$(2) \quad X_n = 2X_{n-1} + 1, \quad X_1 = 1 \quad (1, 3, 7, 15, 31, \dots) \text{ для выбора шага сортировки.}$$

Предположим, что надо отсортировать 1000 значений. Минимальное значение  $X_i \geq 1000$  в (1) будет  $X_7 = 1093$ . Рассматриваются значения  $X_1 = 1$ ,  $X_2 = 4$ ,  $X_3 = 13$ ,  $X_4 = 40$ ,  $X_5 = X_{7-2} = 121$ .

На первом этапе сортируются вставками  $X_5$  последовательностей, образованные 7-8 элементами каждая, индексы которых отстоят от друг от друга на  $X_5 = 121$ :

$$X_1, X_{122}, X_{243}, \dots, X_{969};$$

$$X_2, X_{123}, X_{244}, \dots, X_{970};$$

121 последовательность

...

$$X_{121}, X_{242}, X_{363}, \dots, X_{968}.$$

На втором этапе будем сортировать  $X_4$  последовательности, индексы которых отстоят друг от друга на  $X_4 = 40$

$$X_1, X_{41}, X_{81}, \dots$$

$$X_2, X_{42}, X_{82}, \dots$$

40 последовательностей

...

$$X_{40}, X_{80}, X_{120}, \dots$$

Дальнейшие этапы – рассматриваются  $X_3$  последовательностей элементов, затем  $X_2$  последовательностей элементов и, наконец,  $X_1 = 1$ : одна последовательность, образованная из 1000 исходных значений, которая уже прак-

тически отсортирована благодаря предыдущим этапам. Остается только переставить пары соседей для окончательного упорядочивания.

Составить программу, реализующую данный алгоритм. Исследовать число сравнений и число перемещений (т.е. перестановок элементов с одного места на другое) в процессе применения реализованного метода сортировки.

### **Задание 9.**

Алгоритм сортировки обменов имеет свои достоинства. Пусть имеется текст, слова в котором разделены одним или несколькими пробелами. Требуется переставить их в лексикографическом порядке (т.е. по алфавиту). Так как разные слова могут иметь разную длину, то без больших затруднений можно менять местами только слова, стоящие рядом. Но алгоритм сортировки обменов и предписывает только такие обмены. Написать программу решения этой задачи, используя метод пузырька.

Написать процедуру, реализующую этот метод.

Написать процедуру решения этой задачи, использующую метод сортировки поиском последовательных минимумов.

Исследовать число сравнений и число перемещений (т.е. перестановок элементов с одного места на другое) в процессе применения каждого из реализованных методов сортировки.

В программе предусмотреть диалог с пользователем, позволяющий определить направление сортировки: по алфавиту или в порядке, обратном алфавитному.

### **Задание 10.**

Сортировка слиянием – один из примеров рекурсивных сортировок. В основе этого метода лежит замечание, согласно которому слияние двух отсортированных списков данных выполняется быстро. Список из одного элемента уже отсортирован, поэтому сортировка слиянием разбивает весь исходный список на одноэлементные куски, а затем постепенно сливает их. Поэтому вся деятельность заключается в слиянии двух списков.

Иными словами, сортировку слиянием можно представить в виде рекурсивного алгоритма, выполняющего работу, двигаясь вверх по рекурсии. Исходный список разбивается пополам до тех пор, пока номер первого элемента куска меньше номера последнего элемента в нем. Если же в очередном куске это условие не выполняется, это означает, что мы добрались до списка из одного элемента, который сам по себе уже отсортирован. Затем сливаем два списка длиной в один элемент в отсортированный список длины два. При возврате на следующий уровень два списка длины два сливаются в один отсортированный список длины 4. Этот процесс продолжается, пока мы не доберемся до исходного вызова, при котором две отсортированные половины списка сливаются в общий отсортированный список.

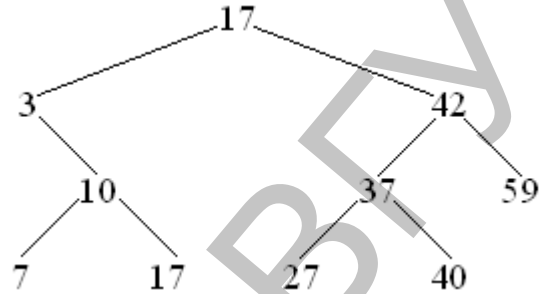
Составить программу, реализующую алгоритм сортировки слиянием. Исследовать число сравнений в процессе применения реализованного метода сортировки.

### Задание 11.

Заданный целочисленный массив упорядочить методом древовидной сортировки.

Рассмотрим данный метод на примере. Пусть следующие значения надо упорядочить по возрастанию: 17, 42, 37, 3, 27, 10, 40, 17, 59, 7,...

Строим двоичный лес, корень которого будет первым значением в упорядочении, здесь это 17. Следующее значение 42, которое больше 17, вставляется в лес как правый сын 17. Значение 37 больше 17, оно является правым потомком 17, но 37 меньше 42, таким образом, оно становится левым сыном 42 и т.д. Таким образом, шаг за шагом строим лес до получения следующего графа (второе значение 17 мы вставили слева от корня, но могли бы точно так же вставить его справа как правого сына 27):



Затем достаточно, начиная с корня, записать каждое значение, слева от него – его левого сына, а справа – правого сына. Следовательно, получаем отсортированный массив.

$$\begin{array}{ccccccc}
 & & (3) & 17 & (42) & & \\
 & & (3 & 10) & 17 & (37 & 42 & 59) \\
 (3 & (7 & 10 & 17)) & 17 & ((27 & 37 & 40) & 42 & (59))
 \end{array}$$

Составить программу, реализующую данный алгоритм. Исследовать число сравнений и число переформировок дерева в применении реализованного метода сортировки.

### Задание 12.

Пирамидальная сортировка. Алгоритм сортировки массивов, предложенный Дж. Уильямсом и развитый Р. Флойдом носит название пирамидального. Он базируется на представлении массива  $x_i$  ( $i = 1..n$ ) в виде бинарного дерева Д.

Сначала введем определение пирамидальной последовательности. Пирамидой называется непустая последовательность элементов массива  $x_i$  ( $i = 1..n$ ) вида  $x_p, x_{p+1}, \dots, x_q$ , ( $1 \leq p \leq q \leq n$ ), для которой выполнено одно из условий:

- $2p > q$ ;
- $2p = q$  и  $x_p \geq x_q$ ;
- $2p < q$  и  $x_j \geq x_{2j}$ , ( $p \leq j \leq q/2$ ),  $x_j \geq x_{2j+1}$ , ( $p \leq j \leq (q-1)/2$ ).

Из определения вытекают следующие свойства:

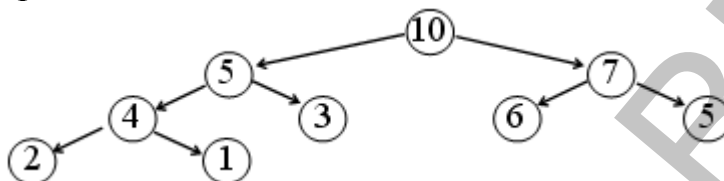


Свойство 1. Для любой последовательности  $x_i$  ( $i = 1..n$ ) подпоследовательность  $x_{n/2+1}, x_{n/2+2}, \dots, x_n$  является пирамидой.

Свойство 2. Если последовательность  $x_i$  ( $i = 1..n$ ) пирамида, то  $x_1 \geq \max(x_j)$ ,  $2 \leq j \leq n$ .

Свойство 3. Если последовательность  $x_i$  ( $i = 1..n$ ) пирамида и представлена в виде бинарного дерева  $D$ , то значение любого узла в  $D$  будет не меньше значений его левого и правого потомков.

Пример. Последовательность 10, 5, 7, 4, 3, 6, 5, 2, 1 является пирамидой. Это видно из дерева:



Перейдем к описанию алгоритма. Реализуется алгоритм в два этапа: построение пирамиды, сортировка пирамиды.

Этап 1. Как отмечалось (свойство 1) «хвост» является пирамидой. Попробуем его нарастить, добавляя к ней оставшиеся элементы массива.

Пусть  $x_{j+1}, \dots, x_n$  – пирамида. Рассмотрим  $x_j, x_{j+1}, \dots, x_n$ . «Просеем»  $x_j$  по соответствующей ветке двоичного дерева. Рассмотрим и его два потомка  $x_{2j}$  и  $x_{2j+1}$ .

Если  $x_j \geq x_{2j}$  и  $x_j \geq x_{2j+1}$ , то вычисления прекращаем. Пополненная последовательность – пирамида. В противном случае обмениваем значения  $x_j$  и  $\max(x_{2j}, x_{2j+1})$  в соответствующих позициях  $D$  и «опустившийся» элемент продолжаем просеивать тем же самым способом. В конце концов получаем пирамиду. Продолжая наращивание, мы превратим весь массив в пирамиду. При этом  $x_1 \geq \max(x_j)$ ,  $2 \leq j \leq n$ . Объявляем полученную пирамиду  $X$  текущей и переходим ко второму этапу.

Этап 2. (Сортировка пирамиды). В текущей пирамиде  $x_1 \geq \max(x_j)$ ,  $2 \leq j \leq n$  обменяем значениями концевые элементы  $X$  и укоротим  $X$  справа на 1. Полученная последовательность может не быть пирамидой. Применим к ней процесс «просеивания» для элемента  $x_1$ , описанный в этапе 1. Преобразованная последовательность  $X$  станет пирамидой. Повторяя этап 2,  $n - 1$  раз отсортируем  $X$  по неубыванию.

Составить программу, реализующую данный алгоритм. Исследовать число сравнений в применении реализованного метода сортировки.

### **Моделирование работы исполнителей**

Требования к выполнению заданий следующие: необходимо написать программу, реализующую работу исполнителя. Организовать работу исполнителя в той среде, в которой он может выполнять команды. В системе команд исполнителя написать алгоритмы решения предложенных заданий и проиллюстрировать их выполнение.

Приведем пример программы, реализующей работу исполнителя Черепаха. Система команд исполнителя:

- ВПЕРЕД ( $n$ ) – Черепаха перемещается вперед на  $n$  шагов.
- ПОВОРОТ ( $\alpha$ ) – Черепаха остается на месте, только поворачивается на угол  $\alpha$ .
- РИСУЙ – При перемещении Черепаха оставляет прорисованную линию.
- НЕ РИСУЙ – При перемещении Черепаха не оставляет прорисованной линии.

В программе приняты следующие допущения:

Черепаха установлена в середине экрана.

Черепаха невидима.

Перо поднято.

Так как при команде НЕ РИСУЙ выполняется прорисовка линий цветом фона, то уже нарисованные линии «стираются», т.е. в режиме команды НЕ РИСУЙ Черепаха может работать как ластик.

В команде ПОВОРОТ значения угла определяется в градусной мере.

Не установлены границы поля Черепахи, т.е. выход за пределы экрана не фиксируется (и ошибкой не считается).

```

Program Turtle;
uses Graph, Crt;
Var
  b : real; { Единственная глобальная переменная для запоми-
            нания текущего направления }
procedure InitTurtle; { Инициализация рабочего поля Че-
репахи }
var gd, gm : integer;
begin
  gd:=detect;
  InitGraph(gd, gm, 'c:\Вр\bgi'); {графический режим экрана }
  ClearDevice;
  SetColor(0); { установка цвета фона }
  lineto(320,199); {установка Черепахи в центр экрана }
  b:=0 {установка начального направления }
end;
procedure PenDown; { Команда РИСУЙ }
begin
  SetColor(3)
end;
procedure PenUp; { Команда НЕ РИСУЙ }
begin
  SetColor(0)
end;
procedure Turn (a:real); { Команда ПОВОРОТ (a) }
begin
  b:=b-a { положительное направление – по часовой
стрелке }
end;

```

```

procedure Forwd (n:integer); { Команда ВПЕРЕД (n) }
var x, y : integer;
begin
  x:=round(cos(b*pi/180)*n*64/30);{вычисление смещения по x}
  y:=round(sin(b*pi/180)*n*64/40);{вычисление смещения по y}
  LineRel(x,y)           {сместиться на вектор (x,y)}
end;
{* РИСОВАНИЕ КАРТИНКИ *}
procedure Figure(a:integer);      { Рисование пятиугольника со
стороной a }
var i : integer;
begin
  PenDown;
  for i:=1 to 4 do begin
    Forwd(a);
    Turn(72)
  end;
  Forwd(a);
  PenUp
end;
procedure Star(a:integer);      { Рисование звезды }
var i : integer;
begin
  PenDown;
  for i:=1 to 5 do begin
    Forwd(a);
    Turn(-144);
  end;
  PenUp
end;
end;
end;
Begin
  InitTurtle;      { Инициализация рабочего поля Черепахи }
  Star(50);        { Звезда }
  Turn(-72);      { Поворот }
  Figure(31);     { Пятиугольник }
  readkey;        {Задержать картинку на экране до нажатия клавиши}
  closegraph
End.

```

Результатом работы этой программы будет нарисованная звезда, вписанная в правильный пятиугольник. Предложенную модель можно усложнить:

- ввести ограничение на угол поворота;
- предусмотреть возможность смены цвета для рисования линий;
- сделать Черепашку видимой;
- написать меню выбора режима работы;
- создать библиотеку картинок-заданий;
- написать меню выбора картинок;

- предусмотреть масштабирование изображений;
- предусмотреть ограничения размеров рабочего поля Черепахи.

### **Задание 1. Исполнитель – Робот**

Робот стоит перед западным концом идущего на восток клетчатого коридора неизвестной длины и шириной в одну клетку. Некоторые клетки коридора закрашены. Программой для Робота является последовательность занумерованных команд из следующего набора:

- ШАГ НА ВОСТОК – смещение на одну клетку;
- ШАГ НА ЗАПАД – смещение на одну клетку;
- ЗАКРАСИТЬ КЛЕТКУ;
- КЛЕТКА ЗАКРАШЕНА? (N) – переход к команде с номером N при выполнении этого условия;
- КЛЕТКА НЕ ЗАКРАШЕНА? (N) – переход к команде с номером N при выполнении этого условия;
- КОНЕЦ КОРИДОРА? – остановка Робота при выполнении этого условия.

Составить программу для Робота, после выполнения которой он дополнительно закрасит все те клетки внутри коридора, которые граничат с клетками, закрашенными до начала исполнения программы, и остановится у восточного конца коридора. Разрешается использовать процедуры, составленные из указанных команд.

### **Задание 2. Исполнитель – Робот**

Из пункта А в пункт В ведет прямой туннель с двухрядным движением. Был изобретен Робот, который может двигаться на север, юг, запад, восток на ширину ряда туннеля (она равна 1 м). Пункт В на N метров севернее и на K метров восточнее пункта А, причем  $N > K > 0$ . Напишите программу для Робота, по которой он пройдет из центра А центр В, не натолкнувшись на стены туннеля. (Считать, что центры пунктов А и В расположены на средней линии туннеля).

При написании программы разрешается использовать команды Робота (шаг на запад, шаг на север, шаг на восток, шаг на юг), все управляющие конструкции языка программирования, а также арифметические операции (+, −, \*, /) и команду присваивания.

### **Задание 3. Исполнитель – Робот-манипулятор**

На координатной плоскости задан Робот-манипулятор, состоящий из двух звеньев длиной  $L_1$  и  $L_2$ . Первое звено шарнирно закреплено в начале координат, второе шарнирно соединено с первым. Робот управляется путем поворота звеньев в первом и втором шарнирах на  $N_1$  и  $N_2$  градусов. Робот предназначен для высверливания отверстий в деталях. Система команд Робота-манипулятора:

- повернуть первое звено на  $N_1$  градусов;
- повернуть второе звено на  $N_2$  градусов.

При написании программы разрешается использовать команды Робота-манипулятора, все управляющие конструкции языка программирования, а также арифметические операции (+, −, \*, /) и команду присваивания.

Составить программу, с помощью которой Робот просверлит в пластине три ряда, расположенных по окружности, по  $K$  отверстий в каждом.

#### **Задание 4. Исполнитель – Робот-кладовщик**

Робот-кладовщик предназначен для поиска нужных деталей. Он может перемещаться перед прямоугольным стеллажом, в ячейках которого находятся ящики с однотипными деталями  $N$  видов.

Список допустимых действий Робота-кладовщика:

- запросить наименование детали;
- переместиться к соседней справа (слева, снизу, сверху) ячейке стеллажа;
- взять из ячейки деталь и поместить ее в грузовой отсек;
- доставить детали к рабочему месту;
- проверить, находится ли в ячейке нужная деталь;
- проверить, является ли ячейка, напротив которой Робот находится, граничной справа (слева, снизу, сверху).

Робот-кладовщик выдает сообщение об отказе в том случае, если в ячейке закончились детали или запрошена несуществующая деталь.

Задание. Доставить к рабочему месту по  $k$  деталей каждого вида.

#### **Задание 5. Исполнитель – Робот-упаковщик**

Робот-упаковщик предназначен для раскладывания по ячейкам стеллажа поступающих из лотка деталей. Он может перемещаться перед стеллажом, состоящим из трех рядов ячеек. Первый ряд предназначен для деталей вида 1, причем в каждую ячейку первого ряда вмещается  $n_1$  деталей; второй – для вида 2, в ячейку вмещается  $n_2$  деталей; третий – для вида 3, в ячейку вмещается  $n_3$  деталей.

Список допустимых действий Робота-упаковщика:

- взять с лотка деталь и определить ее вид;
- переместиться к соседней справа (слева, снизу, сверху) ячейке стеллажа;
- проверить, сколько деталей находится в ячейке;
- проверить, является ли ячейка, напротив которой Робот находится, граничной справа (слева, снизу, сверху).

Робот-упаковщик выдает сообщение об отказе в том случае, если нет места для размещения нужной детали или поступила несуществующая деталь.

Задание. Разложить по ячейкам  $k$  деталей разных видов.

### Задание 6. Исполнитель – Диспетчер

На одноколейной магистрали имеется тупик (см. рисунок), попасть в который можно через две стрелки, расположенные до и после него. Длина состава – 20 вагонов, в тупике умещаются 11 вагонов, включая паровоз.

Обозначим участок до тупика – А; после тупика – В; тупик – Т; паровоз – S; вагоны –  $W_1, W_2, \dots, W_{20}$ .

Диспетчер может использовать следующие команды:

- СТРЕЛКИ В X (X – это Т или АВ) переключение обеих стрелок на тупик Т или на магистраль АВ;
- РАСЦЕПИТЬ X И Y (X и Y – это S или  $W_i$ ) – расцепить, если это возможно, X и Y;
- ПЕРЕМЕСТИТЬСЯ В X (X – это Т, А или В) – переместить S вместе с прицепленным вагоном в X;
- СЦЕПИТЬ X И Y.

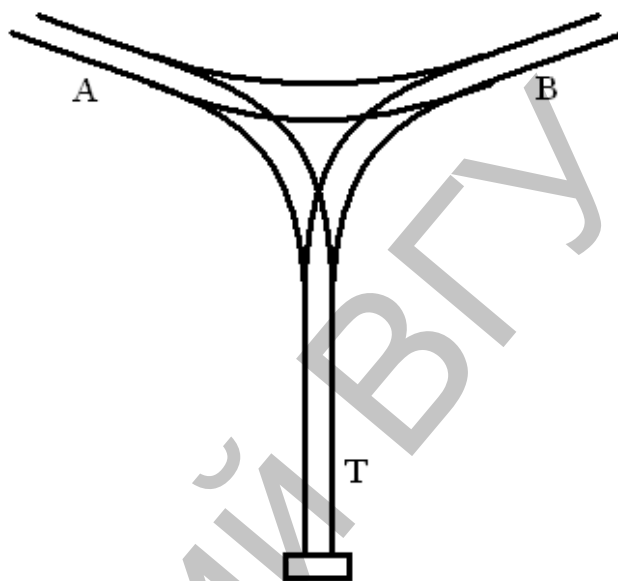
Разрешается использовать процедуры (подпрограммы), составленные из указанных команд. Напишите программу действий диспетчера, обеспечивающих перестановку двух последних вагонов поезда в голову состава. Постарайтесь использовать наименьшее число команд.

### Задание 7. Исполнитель – Плюсик

Плюсик работает со стеком, в котором может храниться от одного до трех чисел, и счетным устройством для выполнения арифметических операций. Счетное устройство работает так. Если, например, от 7 отнимается 5, то число 7 вводится в первый вход устройства, а число 5 – во второй вход устройства. Устройство сообщает, какую операцию выполнять, оно работает и выдает результат на выходе Р.

Система команд исполнителя:

- ОЧИСТЬ – стек очищается от всех чисел;
- ЗАПОМНИ (N) – число N запоминается в стеке;
- СЛОЖИ – первое число из стека поступает на вход 2 счетного устройства, второе число – на вход 1, выполняется операция сложения; результат с выхода Р запоминается в стеке;
- ВЫЧТИ – первое число из стека поступает на вход 2 счетного устройства, второе число – на вход 1, выполняется операция вычитания; результат с выхода Р запоминается в стеке;



- УМНОЖЬ – первое число из стека поступает на вход 2 счетного устройства, второе число – на вход 1, выполняется операция умножения; результат с выхода Р запоминается в стеке;
- ДЕЛИ – первое число из стека поступает на вход 2 счетного устройства, второе число – на вход 1, выполняется операция деления; результат с выхода Р запоминается в стеке.

Когда исполнитель сигнализирует НЕ МОГУ ВЫПОЛНИТЬ:

- ЗАПОМНИ (N) – если  $N > 999$  или в стеке нет места;
- СЛОЖИ – в стеке нет двух чисел или сумма больше 999;
- ВЫЧТИ – в стеке нет двух чисел или число в верхушке стека больше числа, расположенного под ним;
- УМНОЖЬ – в стеке нет двух чисел или результат больше 999;
- ДЕЛИ – в стеке нет двух чисел, или деление на нуль, или нацело не делится.

При написании программы разрешается использовать только команды Плюсика. Помогите Плюсику решить следующие задачи:

а) Крестьянин пошел на базар. Он купил 9 упаковок овса по 5 кг в каждой для любимой лошади, а для дочки купил 7 пачек ванильных сухариков по 3 кг. Килограмм овса стоит 15 копеек, а килограмм сухариков – 13 копеек. Сколько денег заплатил крестьянин за покупку?

б) Проверить арифметическое равенство:  $\frac{13 \cdot 7 + 8 \cdot 5 + 4}{17 \cdot 7 - 8 \cdot 11 - 16} = 9$ .

### Задание 8. Исполнитель – Вычислитель

Исполнитель работает только с целыми числами. Система его команд такова:

- ЗАПРОСИТЬ (a,b) – ввести числитель и знаменатель дроби a/b;
- СООБЩИТЬ (a,b) – вывести числитель и знаменатель дроби a/b;
- ВЫЧИСЛИТЬ ДРОБЬ ВИДА 1 – из дроби a/b получить дробь (a-b)/b;
- ВЫЧИСЛИТЬ ДРОБЬ ВИДА 2 – из дроби a/b получить дробь (a+b)/b;
- ВЫЧИСЛИТЬ ДРОБЬ ВИДА 3 – из дроби a/b получить дробь b/a.

Т.е. из любой дроби вида a/b исполнитель может получить любую из указанных дробей.

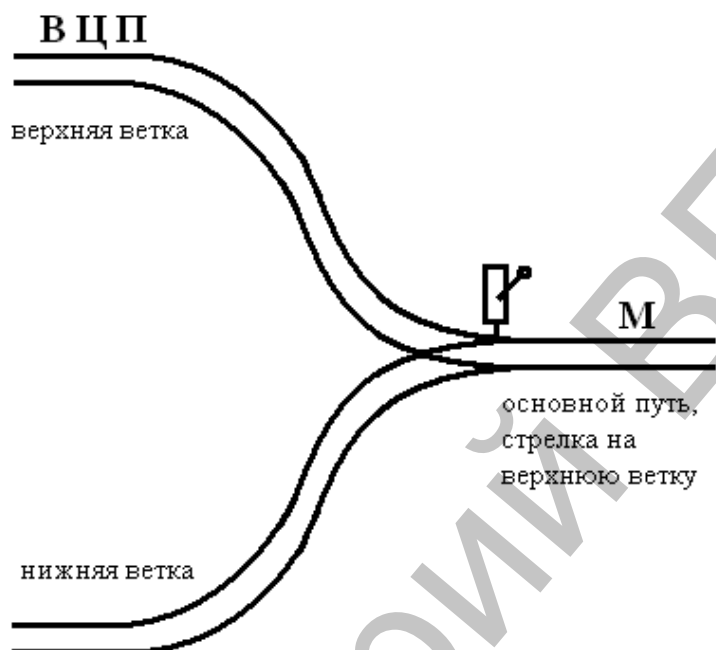
При написании программы разрешается использовать команды Вычислителя, управляющие конструкции языка программирования (ветвления, циклы и процедуры, образованные этими конструкциями и командами исполнителя) и логические отношения и функции.

а) Как из дроби 67/91 получить дробь 1/2?

б) Получить из дроби 1/4 дробь 11/119 за наименьшее число шагов.

### Задание 9. Исполнитель – Машинист

Среда Машиниста включает три железнодорожных пути со стрелкой и железнодорожный состав, состоящий из тепловоза (М) и трех вагонов: крытого (В), цистерны (Ц), платформы (П).



Система команд Машиниста:

- **ВПЕРЕД** – если тепловоз стоит в начале основного пути, то по этой команде он продвигается до стрелки; если же он стоит у стрелки, то команда заставляет тепловоз идти в тупик по той из веток, путь на которую открыт стрелкой;
- **НАЗАД** – тепловоз движется задним ходом и откатывается в начало основного пути;
- **СТРЕЛКА** – переводит стрелку на развилке из двух путей; если первый путь был закрыт (и, значит, открыт второй), то команда открывает первый путь и закрывает второй;
- **ПРИЦЕПИ** – к тепловозу (или к целому составу, который везет тепловоз) прицепляется вагон (только один);
- **ОТЦЕПИ** – отцепляет последний вагон состава.
- **НЕ МОГУ** – это сообщение Машинист дает в следующих случаях:

Команда	Когда возникает ситуация «НЕ МОГУ»
1 ВПЕРЕД НАЗАД	Тепловоз или весь состав упирается в тупик
2 СТРЕЛКА	Тепловоз не у стрелки
3 ПРИЦЕПИ	Тепловоз не у вагонов или прицеплены не все вагоны
4 ОТЦЕПИ	Состав не в тупике или отцеплены все вагоны

Задание 1. Переставить вагоны на нижний путь в обратном порядке.

Задание 2. Переставить платформу и вагон на нижний путь.



### **Задание 10. Исполнитель – Муравей**

Исполнитель «Муравей» живет на клетчатой доске  $8 \times 8$  клеточек. Он умеет передвигаться влево, вправо, вверх и вниз на одну клетку при нажатии на соответствующие клавиши-стрелки. Если на пути ему попадаются «кубики», то он может их передвинуть, как бы толкая перед собой. Выйти за пределы доски или столкнуться с ней кубики Муравей не может.

Управляя Муравьем, необходимо сложить из кубиков заданную фигуру, получить из одной фигуры другую, исправить ошибку в слове или арифметическом примере (на верхней грани кубиков нарисованы буквы и цифры), сложить слово или несколько слов.

Программа должна предусматривать как непосредственный, так и пакетный режим работы с этим исполнителем. В первом случае управление Муравьем производится с помощью стрелок, последовательность команд должна быть отражена в отдельном окне на экране. При пакетном режиме готовая программа для исполнителя загружается из файла (предусмотреть возможность редактирования программы для Муравья).

### **Задание 11. Исполнитель – Таракан**

Таракан может ползать по клетчатому полю и передвигать по нему кубики с разными буквами на них. За один ход он сдвигается на одну клетку вверх, вниз, вправо, влево (но не по диагонали).

Натолкнувшись на один или несколько кубиков, он будет толкать их перед собой, пока не изменит направление движения. Считается, что поле обнесено по краям глубоким рвом. Таракан может спихнуть в ров кубики, но сам лезть туда боится. Для этого Таракан умеет определять положение рва, если он находится в соседней клетке: справа (слева, сверху, снизу) ров.

Задание. Собрать из кубиков разбросанное слово.

### **Задание 12. Исполнитель – Кузнечик**

Средой данного исполнителя служит координатная ось, вдоль которой он может перемещаться следующим образом:

- ПРЫЖОК ВПЕРЕД – Кузнечик прыгает на 3 единицы вперед;
- ПРЫЖОК НАЗАД – Кузнечик прыгает на 2 единицы назад;
- СООБЩИТЬ ЧИСЛО – Кузнечик указывает координату своего положения на оси.

Начальное положение исполнителя – точка с координатой 0, «лицом» в сторону положительных чисел.

При написании программы разрешается использовать команды Кузнечика и процедуры, образованные этими командами.

а) Как Кузнечику попасть из точки 67 в точку -2?

б) Вывести значения четных чисел в диапазоне от 1 до 30.

### **Задание 13. Исполнитель – Вездеход**

Вездеход представляет собой круг диаметром девять единиц. Он движется в прямоугольном поле, в котором может находиться некоторое количество

стен произвольной длины, расположенных под произвольными углами. Вездеход может поворачиваться на произвольный угол и двигаться в любом заданном направлении. На вездеходе имеется локатор, который может измерять расстояние до ближайшей стены в направлении движения этого исполнителя.

Вездеход может отвечать на запросы о своем местоположении.

Система команд Вездехода следующая:

- ВПЕРЕД ( $n$ ) – сдвинуться вперед на  $n$  единиц;
- НАЛЕВО ( $a$ ) – повернуться налево на  $a$  градусов;
- НАПРАВО ( $a$ ) – повернуться направо на  $a$  градусов;
- РАССТОЯНИЕ – команда возвращает расстояние от центра Вездехода до ближайшей стены в направлении движения Вездехода;
- КООРДИНАТА\_Х – команда возвращает координату Х центра Вездехода;
- КООРДИНАТА\_У – команда возвращает координату У центра Вездехода;
- УГОЛ – команда возвращает направление Вездехода.

При написании программы разрешается использовать команды Вездехода, все управляющие конструкции языка программирования, арифметические операции и команду присваивания.

Вездеход стоит перед стеной конечной длины, имеющей несколько проходов. Ему требуется, не сходя с места, определить, имеется ли среди них такой проход, через который он сможет пройти. Если да, то стать в положение, симметричное исходному относительно стены, пройдя через найденный проход. Если нет, то обойти стену и стать в положение, симметричное исходному относительно стены.

#### **Задание 14. Исполнитель – Весовщик**

Программой для Весовщика является последовательность занумерованных команд из следующего набора:

- положить монету с номером  $N$  на правую чашку весов
- положить монету с номером  $N$  на левую чашку весов
- снять монету с номером  $N$  с правой чашки
- снять монету с номером  $N$  с левой чашки
- проверка условий: правая тяжелее? левая тяжелее?

Задача 1. Составить алгоритм нахождения 1 фальшивой монеты среди  $k$  настоящих монет того же достоинства с помощью чашечных весов, если известно, что фальшивая монета тяжелее настоящей.

Задача 2. Имеется 1000 монет, из которых одна фальшивая (легче других). Придумайте способ нахождения фальшивой монеты за 7 взвешиваний на чашечных весах без гирь.

#### **Задание 15. Исполнитель – Корректор**

Корректор – робот, подобно машине Тьюринга работающий с длинной лентой, разбитой на ячейки. Кроме ленты, у него есть ящик, в который он может копировать символ с ленты или записывать из него копию символа

на ленту. Цель работы исполнителя – корректировка информации на ленте. Корректор может выполнять следующие действия:

- ПИСАТЬ( $a$ ) – записать символ  $a$  в обозреваемую ячейку ленты;
- ЯЩИК+ – символ с ленты копируется в ящик;
- ЯЩИК– – символ из ящика копируется на ленту;
- ВЛЕВО – шаг исполнителя по ленте влево;
- ВПРАВО – шаг исполнителя по ленте вправо;
- ПЛЮС – замена символа на ленте следующим по порядку в заданном алфавите символом;
- МИНУС – замена символа на ленте предыдущим в заданном алфавите символом;
- ГОЛОС – подача звукового сигнала.

Корректор располагает условной командой, которая позволяет сравнивать взаимное положение ( $>$ ,  $<$ ,  $=$ ) в алфавите двух символов – символа в текущей ячейке ленты с символом в ящике.

Алфавит исполнителя.  $\wedge$ , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, а, б, в, г, д, е, ж, з, и, к, л, м, н, о, п, р, с, т, у, ф, х, ц, ч, ш, щ, ь, ы, ь, э, ю, я, \_ (пробел). Специальный символ пустого места обозначается  $\wedge$ , пробел – чистой ячейкой.

Задание 1. К десятичному числу на ленте прибавить 10.

Задание 2. Заменить в тексте определенной длины один символ на другой.

### **Задание 16. Исполнитель – Переместитель**

Исполнитель может перемещаться вдоль бесконечной ленты, разбитой на ячейки (клетки). В ячейках хранится символьная информация.

Система команд исполнителя следующая:

- ВПРАВО – переместиться на одну клетку вправо;
- ВЛЕВО – переместиться на одну клетку влево;
- ПРОЧИТАТЬ – прочитать символ из ячейки, над которой находится;
- ПЕРЕПИСАТЬ – переписать символ из ячейки, над которой находится исполнитель, в пустую.

При написании программы разрешается использовать команды Переместителя, все управляющие конструкции языка программирования, арифметические операции и команду присваивания.

Пусть длина ленты –  $2 \times n + 1$  ячеек. Из них одна пустая, а в остальных расположено по  $n$  символов « $a$ » и « $b$ ». Переместителю необходимо расположить все символы « $a$ » в начало ленты, а все « $b$ » – в конец, причем конечное положение пустой ячейки значения не имеет.

### **Задание 17. Исполнитель – Водолей**

Имеются несколько сосудов различной емкости. Исполнитель Водолей может выполнить следующие команды:

- НАПОЛНИ СОСУД ( $l$ );
- ВЫЛЕЙ ИЗ СОСУДА ( $l$ );
- ПЕРЕЛЕЙ ИЗ ( $l$ ) В ( $r$ ), – где  $l, r$  – один из сосудов.

При написании программы разрешается использовать команды Водолея, все управляющие конструкции языка программирования, арифметические операции и команду присваивания.

а) Имеются два сосуда емкостью 5 и 7 л. Для приготовления раствора необходимо отмерить 4 л воды. Как это сделать?

б) В бидоне не менее 10 л молока. Как отлить из него 6 л молока с помощью пустых девятилитрового ведра и пятилитрового бидона?

в) Ученики колдуна украли у него бутылку с бальзамом и три пустых флакона для дележки добычи. В укромном месте они прочитали надписи, сделанные на бутылке и флаконах, и узнали, что бутылка вмещает 30 унций, а флаконы – 14, 12 и 6 унций соответственно. Как им разделить добычу поровну?

### **Задание 18. Исполнитель – Кантователь кубика**

Под кантованием понимается перекачивание кубика через соприкасающееся со столом ребро без скольжения. Другие способы перемещения кубика запрещены. Нумерация граней кубика такова, что если его положить на грань с цифрой 5, то на боковых гранях будут цифры 1, 6, 4, 3 при обходе по часовой стрелке, а на верхней – цифра 2.

Кантователь кубика может выполнять следующие команды:

- перекантовать кубик на нас;
- перекантовать кубик от нас;
- перекантовать кубик влево;
- перекантовать кубик вправо.

На столе лежит игральный кубик гранью  $M(0)$  к нам, гранью  $L(0)$  вверх. Написать программу, определяющую последовательность кантований кубика, после выполнения которых кубик окажется на прежнем месте, но гранью  $M(k)$  к нам, а гранью  $L(k)$  вверх. (Решение можно получить, вращая кубик вокруг диагоналей, т.е. когда некоторая точка остается неподвижной, причем достаточно вращений вокруг двух диагоналей, которые определяются начальным и конечным состояниями. Если за два вращения не удалось получить искомым результат, то задача неразрешима).

### **Задание 19. Ханойские башни**

Имеются три стержня, на одном из которых собрано  $N$  колец в порядке уменьшения их диаметров (самый большой диаметр у нижнего кольца). Требуется переложить кольца на третий стержень, пользуясь вторым как вспомогательным.

Исполнитель, перекладывающий кольца имеет следующую систему команд:

- ПЕРЕЛОЖИТЬ КОЛЬЦО С  $(x)$  НА  $(y)$  – где  $x, y$  – номера стержней.
- Исполнитель сигнализирует НЕ МОГУ ВЫПОЛНИТЬ при попытке класть кольцо большего диаметра на кольцо меньшего диаметра.

При написании программы разрешается использовать команду исполнителя, все управляющие конструкции языка программирования, арифметические операции и команду присваивания.

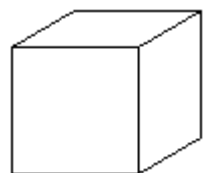
- а. Переложить 4 кольца.  
б. Переложить  $N$  колец.

### Задание 20. Исполнитель – Черепаха

Посередине листа белой бумаги установлен исполнитель Черепаха, система команд которого следующая:

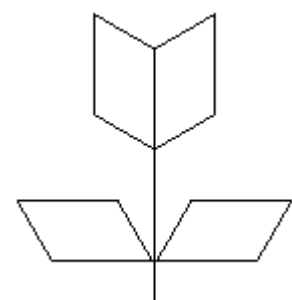
- ВПЕРЕД ( $n$ ), НАЗАД ( $n$ ) – команды перемещения на  $n$  шагов в указанном направлении;
- ВПРАВО ( $\alpha$ ), ВЛЕВО ( $\alpha$ ) – команды поворота на угол  $\alpha$  шагов в указанном направлении;
- РИСУЙ, НЕ РИСУЙ – команды, после исполнения которых происходит настройка исполнителя на то, каким будет перемещение – с прорисовкой линии или нет.

При написании программы разрешается использовать команды Черепахи, все управляющие конструкции языка программирования, арифметические операции и команду присваивания.

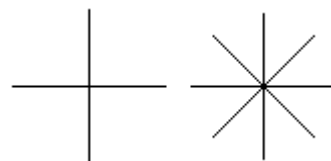


а. Составить программу рисования куба, например, такого, как показано на рисунке слева.

б. Составить программу рисования «цветка» с произвольным количеством «лепестков», имеющего круглую форму. Например, на рисунке справа приведены «цветы» с четырьмя и восьмью лепестками.



в. Составить программу рисования стилизованного цветка, который приведен на рисунке слева.



### Программирование игровых задач

Составить программу, реализующую игру, описанную в условии, соблюдая следующие требования. Программа должна по желанию игрока выводить правила игры, изображать игровое поле, позволять вводить очередной ход, отображать сделанный ход на игровом поле, проверять допустимость хода, сообщать результат игры.

#### Задание 1. Определители

Игровое поле имеет вид квадрата  $3 \times 3$ . Двое по очереди ставят в свободную клетку квадрата одно из чисел от 1 до 9 (за одну игру первый игрок делает пять ходов, второй – четыре). Число, однажды поставленное, ставить второй раз нельзя.

Если определитель полученной матрицы положителен, то выиграл первый игрок, в противном случае – второй.

#### Задание 2. Квадратобязнь

Игровое поле – квадрат  $7 \times 7$  (можно предложить игрокам выбрать размеры поля, начиная с квадрата  $3 \times 3$ ). Игроки по очереди ставят в свободные

клетки поля красные (первый) и белые (второй) фишки. Проигрывает тот, после хода которого четыре фишки его цвета оказались вершинами некоторого квадрата (стороны которого не обязательно параллельны сторонам игрового поля).

### Задание 3. Выбор шара

В урне находится  $n$  шаров. На каждом шаре написано число (все числа различные). Число  $n$  велико и известно заранее, а числа, написанные на шарах, – любые (не обязательно от 1 до  $n$  и целые) и заранее неизвестны.

Игра состоит в следующем. Играющий выбирает наудачу шар из урны, прочитывает на нем число и решает – берет он этот шар или нет.

Если не берет, то выбранный шар отбрасывается (не возвращается в урну), и игра продолжается. Играющий выбирает наудачу новый шар, читает на нем число, и снова решает, брать ему этот шар или нет, и т.д.

Если играющий берет очередной шар, то игра прекращается и игрок считается выигравшим в том и только в том случае, если на выбранном им шаре окажется самое большое среди чисел, написанных на всех  $n$  шарах, находящихся в урне с самого начала.

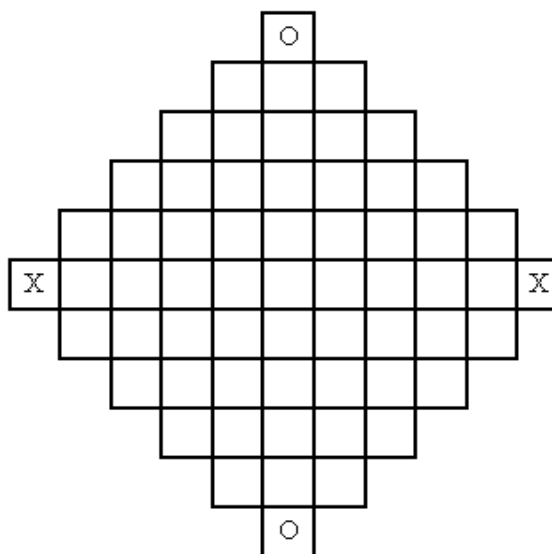
Требуется указать тактику играющего, при которой он будет иметь максимальные шансы на выигрыш. (Для начала попробуйте найти тактику, при которой вероятность выигрыша не стремится к нулю, когда число шаров  $n$  стремится к бесконечности).

### Задание 4. Фишки

Двое играют в такую игру: имеется  $n$  фишек; первый игрок берет любое количество фишек; единственное, что он не может, это взять их все. Затем он передает ход партнеру, и дальше они ходят по очереди, причем каждый игрок забирает одну или более фишек, но не больше чем удвоенное количество фишек, взятых предыдущим игроком. Выигрывает тот, кому достанется последняя фишка.

### Задание 5. Большие крестики-нолики

Исходный вид игрового поля показан на рисунке. Игроков двое. Ходят игроки поочередно. Каждый игрок может за один ход нарисовать вертикальный или горизонтальный отрезок единичной длины. Если на очередном шаге у игрока возникла ситуация, когда он должен нарисовать четвертую сторону квадрата, то он рисует четвертый отрезок и заполняет получившийся квадрат своим значком (0 или X) и получает право дополнительного хода. В результате после



игры все поле заполняется значками X или 0. Побеждает тот, чьих значков больше.

### Задание 6. Ход конём

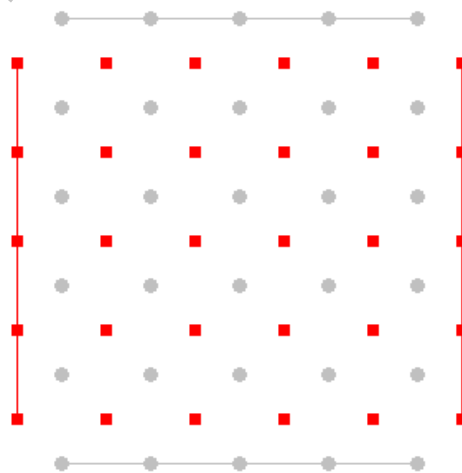
Квадратное поле из 64 клеток нужно закрасить, начиная с верхней левой клетки. Переходить от клетки к клетке можно только ходом коня. Изобразить индикатор, который будет показывать количество закрасенных клеток (количество очков).

### Задание 7. Чет и нечет

На прямоугольном игровом поле нарисованы пять кругов: один в середине прямоугольника и четыре в углах. Два из угловых кругов помечены «чет» и «нечет». В центральном круге помещаются восемь пронумерованных шашек столбиком по порядку номеров с цифрой 8 вниз и с цифрой 1 наверх. Задача состоит в том, чтобы за наименьшее число ходов переместить шашки с цифрами 1, 3, 5 и 7 из центрального столбика в кружок с пометкой «нечет», а шашки с цифрами 2, 4, 6 и 8 в кружок с пометкой «чет». Ходом считается всякое перемещение шашки с одного места на другое. За один ход можно перемещать с кружка на кружок только одну шашку, но при этом нельзя класть шашку, пронумерованную большим числом, на шашку, имеющую меньший номер. Нельзя на один, какой бы то ни было кружок (в том числе и обратно – на центральный), помещать одновременно шашки с четными и нечетными номерами.

### Задание 8. Ломаные

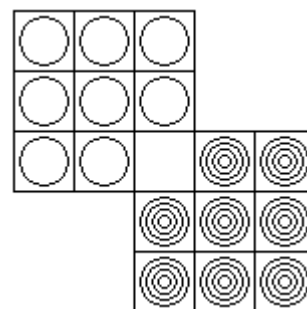
На рисунке изображено игровое поле. Участники по очереди проводят вертикальные и горизонтальные отрезки, соединяя ими точки своего цвета. Соединять точки по диагонали нельзя. Линии противников нигде не должны пересекаться. Выигрывает тот, кто первым построит ломаную, связывающую противоположные стороны доски своего цвета.



### Задание 9. Перемещение шашек

8 черных и 8 белых шашек расставлены так, как показано на рисунке. Требуется, не снимая шашек с поля, переместить все черные шашки на места белых, а белые – на места черных.

Шашки могут передвигаться вперед и назад, вправо и влево, но не наискось. В тех же направлениях разрешается перепрыгивать через одну шашку и на свободную клетку. Две шашки в одну клетку помещать нельзя. Очередность в перемещении белых и черных



шашек соблюдать не требуется; если нужно, то можно перемещать несколько раз подряд шашки одного цвета.

### Задание 10. «Магический» квадрат

В этой игре девять чисел от единицы до девятки определенным образом перемещаются по полям разграфленного квадрата  $3 \times 3$ . Игра напоминает сборку кубика Рубика. Будем считать, что при сдвиге какого-либо его столбца сверху вниз содержимое верхнего поля передвигается в среднее, содержимое среднего – в нижнее, содержимое нижнего – в верхнее. Строки сдвигаются аналогично.

3	8	9
7	5	6
4	1	2

Цель игры – превратить допустимыми перемещениями исходное расположение цифр на полях квадрата в такое, в котором они расположены на дополнительной клавиатуре компьютера или калькулятора.

### Задание 11. Тест на тренировку памяти и внимания

На экран выводится таблица, в некоторых клетках которой расположены кружки. После небольшой задержки кружки исчезают. Требуется указать клетки таблицы, в которых находились кружки.

Тест содержит несколько уровней. На первом уровне таблица может иметь размеры  $3 \times 4$  клетки, на втором  $4 \times 6$  и т.д. На каждом уровне количество кружков изменяется от 1 до максимального, зависящего от размера таблицы. Расположение кружков выбирается случайным образом. При правильном выполнении задания число кружков увеличивается на 1, при неправильном – остается тем же. Если же и второй раз задание будет выполнено неверно, число кружков следует уменьшить на единицу. За правильно выполненное задание начисляются очки. Их количество зависит от числа кружков.

Предусмотреть автоматический переход на следующий уровень при прохождении текущего, а также возможность пользователю выбрать стартовый уровень.

### Задание 12. Кроссворд

Разработать программу разгадывания кроссвордов. На экране рисуется сетка кроссворда с проставленными номерами. В меню выбирается номер по горизонтали или вертикали, при этом внизу появляется формулировка задания. При вводе слова-ответа буквы появляются в сетке кроссворда. Исправления делаются только в режиме замены. Когда будут заполнены все клетки (или раньше, если пользователь нажмет, например, клавишу F1), правильно отгаданные слова будут выделены зеленым, а отгаданные неправильно – красным цветом.

Выбирать отгадываемое слово удобнее всего непосредственно на сетке кроссворда с помощью курсора. После выбора слова внизу появляется текст одного или двух (по горизонтали и вертикали) вопросов. Слово-отгадка вписывается по буквам в сетку кроссворда.



Пользователь может набирать ответ как прописными, так и строчными буквами, и это должно учитываться при проверке правильности решения (т.е. правильность ответов не должна зависеть от регистра, на котором набраны буквы). В программе должна быть предусмотрена возможность выбора кроссворда из списка (имя и сетка кроссворда вместе с заданиями должна храниться в файле).

### **Перколяция (просачивание)**

Составить программу решения задачи перколяции. Графическая иллюстрация процесса обязательна.

#### **Задание 1.**

Квадрат размера  $a \times a$  разбит вертикальными и горизонтальными отрезками. На пересечении отрезков находятся узлы. Удаление узла – это удаление всех входящих в него отрезков длины 1.  $N$  узлов, выбранные случайным образом, удалены. Определить, может ли вода протечь с верхней стороны квадрата на нижнюю, т.е. существует ли связный путь из единичных отрезков, не лежащих на сторонах квадрата? Найти коэффициент перколяции  $P = \frac{N}{a^2}$ , при которых половина решеток дает протекание, а половина – его отсутствие.

#### **Задание 2.**

В задании 1 заменить квадрат на прямоугольник размера  $a \times b$ , горизонтальная сторона которого меняется в заданных пределах. Построить график зависимости коэффициента перколяции  $P = \frac{N}{ab}$  от длины этой стороны. Значения  $a, N$  вводить с клавиатуры.

#### **Задание 3.**

$N$  клеток, выбранных случайным образом на шахматной доске, покрашены в синий цвет, остальные – в красный. Определить, может ли ладья пройти снизу вверх по синим клеткам. Найти пороговое значение коэффициента перколяции ( $P = \frac{N}{64}$ ), т.е. определить, при каком числе синих клеток протекание ладьи происходит в 70% случаев.

#### **Задание 4.**

На красную шахматную доску случайным образом брошено  $N$  синих клеток. Определить, сможет ли король пройти слева направо по красным клеткам. Найти пороговое значение перколяции ( $P = \frac{N}{64}$ ), т.е. определить, при каком числе красных клеток вероятность протекания короля приблизительно равна  $1/2$ ?

**Задание 5.**

Задание 3 рассмотреть для прямоугольной доски размера  $a \times b$ , вертикальная сторона которой меняется в заданных пределах. Построить график зависимости коэффициента перколяции  $P = \frac{N}{ab}$  от длины этой стороны.

Значения  $a, N$  вводить с клавиатуры.

**Задание 6.**

В двумерной квадратной решетке  $a \times a$  каждый узел либо занят с вероятностью  $P$ , принадлежащей  $[0, 1]$ , либо свободен с вероятностью  $1-P$ . Найти коэффициент перколяции при протекании с левого края на правый.

**Задание 7.**

В задании 6 найти коэффициент перколяции при протекании сверху вниз в половине случаев.

**Задание 8.**

Задание 4 рассмотреть для прямоугольной доски размера  $a \times b$ , горизонтальная сторона которой меняется в заданных пределах. Построить график зависимости количества синих клеток, при которых коэффициент перколяции  $P = \frac{N}{ab}$  равен  $1/2$  от длины этой стороны. Значения  $a, N$  вводить с клавиатуры.

**Задание 9.**

В квадрате, рассеченном на клетки, каждое ребро с вероятностью  $P$  становится ребром с сопротивлением  $r$  и с вероятностью  $1-P$  ребром с сопротивлением  $R \gg r$ . Найти пороговое значение вероятности  $P$ , при котором протекание тока происходит в 10% случаев (по ребрам с сопротивлением  $r$ ).

**Задание 10.**

В задании 9 заменить квадрат на прямоугольник и определить пороговое значение вероятности  $P$ , при котором протекание тока происходит в половине случаев.

**Задание 11.**

Квадрат размера  $51 \times 51$  разбит вертикальными и горизонтальными отрезками на 2601 одинаковых квадратов  $1 \times 1$ . На пересечении отрезков находятся узлы. Удаление узла – это удаление всех входящих в него отрезков длины 1.  $N$  узлов, выбранные случайным образом, удалены.

1. Определить, может ли вода «протечь» с верхней стороны квадрата на нижнюю, т.е. существует ли связный путь из единичных отрезков, не лежащих на сторонах квадрата, один конец которого лежит на верхней стороне, другой – на нижней?
2. При каких  $N$  существование протекания более возможно, чем его отсутствие? Найти коэффициент перколяции  $P = N/2601$ , при которых эти со-

бытия равновероятны, т.е. половина решеток с  $N$  удаленными узлами дает протекание, а половина – его отсутствие.

3. Значение стороны квадрата 51 выбрано случайно. Поменяйте это значение и посмотрите, что случится с  $N$ .
4. Замените квадрат на прямоугольник, одна сторона которого фиксирована, а другая меняется в заданных пределах. Как зависит коэффициент перколяции от длины второй стороны?

### Задание 12.

Некоторые клетки на шахматной доске покрашены в красный, остальные – в синий цвета. Доказать, что либо ладья может перейти с первой горизонтали на восьмую по синим клеткам, либо король с вертикали  $A$  на вертикаль  $H$  – по красным.

1. На красной шахматной доске окрашено в синий цвет несколько клеток. Найти путь ладьи снизу вверх по синим клеткам или путь короля слева направо по красным.
2. Найти пороговое значение перколяции: при каком числе клеток вероятность прохождения ладьи равна  $1/2$ .
3. Исследовать доски с другими размерами (в том числе прямоугольные).

### Исследовательские задачи

#### Задание 1. Решение квадратного уравнения

Рассмотрим квадратное уравнение  $ax^2 + bx + c = 0$ . Перепишем его в виде  $x = \frac{-(bx + c)}{ax}$ . Положим теперь  $x_{n+1} = \frac{-(bx_n + c)}{ax_n}$ .

Выбрав произвольное значение  $x_0$  (отличное от нуля), получаем последовательность  $x_0, x_1, \dots$

1. Проверьте, что эта последовательность сходится к корню уравнения.
2. К какому из двух корней сходится эта последовательность? Как это зависит от выбора значений  $x_0$ ?
3. Что происходит с последовательностью, если у исходного уравнения нет вещественных корней?
4. Как ведет себя последовательность, если мы начнем ее с комплексного числа?
5. Сравните этот метод приближенного решения квадратного уравнения (метод итераций) с другими, известными Вам.

#### Задание 2. Эллиптический бильярд

Точечный шарик равномерно и прямолинейно движется внутри эллипса

$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ , отражаясь от него по обычному закону отражения. Это означает, что равны угол падения и угол отражения, измеряемые относительно касательной к эллипсу в точке падения. Для удобства наблюдения след

шарика должен оставаться на экране. Наблюдая движение шарика, исследовать:

- 1) как будет меняться траектория шарика, пущенного из фокуса эллипса;
- 2) как выглядит траектория, пересекающая отрезок между двумя фокусами;
- 3) как выглядит траектория, не пересекающая этот отрезок;
- 4) по теореме Биркгофа в этом бильярде существуют замкнутые траектории из любого числа звеньев. Попробуйте отыскать траектории из двух, трех, четырех, пяти звеньев.

Программа должна позволять задавать начальное положение и направление движения шарика, а также полуоси эллипса (или одну из этих полуосей).

### **Задание 3. Случайные фрактальные скопления**

В начальный момент на экране изображены окружность и точка в центре этой окружности. Точка является зародышем – первым элементом интересующего нас множества. Шаг процедуры состоит в следующем. Случайно выбранная точка на окружности начинает случайное «блуждание»: она с одинаковой вероятностью  $1/4$  делает шаги влево, вверх, вниз или вправо. Если одна из четырех соседних с ней точек оказывается принадлежащей к нашему множеству, то блуждающая точка «прилипает» (добавляется к множеству), а с окружности отправляется в путь следующая точка. Если же точка в своих блужданиях вышла за пределы окружности, она считается «потерянной» и с окружности запускается следующая точка.

Через некоторое время получится множество, похожее на кучу железных опилок на магните.

1. Попробуйте применить различную раскраску точек в зависимости от того, на каком шаге произошло «прилипание».
2. Радиус окружности 50 (точек) – хорошая база для начала исследования (попробуйте другие его значения).
3. Поменяйте правило случайного блуждания:
  - разрешите переходы по диагонали;
  - отправляйте точку с треугольника, с квадрата;
  - замените квадратную решетку шестиугольной.
4. Самый интересный вопрос – это вычисление размерности такого фрактала. Такое множество более плотно, чем отрезок (размерность которого равна 1), но разреженнее круга (размерность которого равна 2). Проверьте это утверждение.

### **Задание 4. «Светофор»**

На окружности расположены  $n$  точек, каждая окрашена в один из трех цветов: красный, желтый или зеленый. Шаг процедуры состоит в одновременном изменении цветов всех точек. На каждом шаге цвет каждой точки меняется по следующему правилу:

- если две точки, соседние с данной, имеют одинаковый цвет, то точка перекрашивается в этот цвет;
- если две точки, соседние с данной, имеют разные цвета, то точка перекрашивается в третий цвет.

Ясно, что если в начальный момент все точки окрашены в один цвет, то перекрашивания не происходит.

Существуют ли другие конфигурации, в которых ничего не меняется? При каких  $n$  это возможно?

Исследуйте поведение различных конфигураций при различных  $n$ . При каких  $n$  существуют циклы длины 2 (т.е. конфигурация повторяет саму себя через каждые два шага)? Циклы длины 3, 4, 5?

Перечислите все длины циклов при заданном  $n$ .

Попробуйте обобщить задачу на:

- большее количество цветов;
- на другое расположение точек, например, точки на торе.

Описанное преобразование хорошо выражается в терминах линейной алгебры над полем из трех элементов.

### Задание 5. «100»

Дан автобусный билет с номером, состоящим из  $n$  цифр. Расставить между цифрами знаки арифметических операций («+», «-», «\*», «/») и скобки таким образом, чтобы значение полученного выражения было равно 100. Можно образовывать многозначные числа из стоящих рядом цифр.

Выражение должно быть корректным с точки зрения арифметики. Допустимы лишние скобки, не нарушающие корректности выражения.

Условимся, что

- знаки операций означают:  
«+» – сложение;  
«-» – вычитание (унарный минус не допускается!);  
«\*» – умножение;  
«/» – деление нацело;
- используется стандартный приоритет операций;
- число  $n$  в номере билета не больше 6;
- при возникновении затруднений в решении задачи допустимо частное решение, использующее не все знаки арифметических действий и/или скобки;
- если для указанного номера решение найти не удастся, то программа выдает сообщение об этом на экран.

Задания

1. Исследуйте номера билетов из шести цифр, если номера задаются произвольно. Каково соотношение количества билетов, для которых удалось найти решение, к количеству билетов, для которых решение не найдено.

2. Пусть значение выражения, полученного согласно описанным правилам, должно равняться не 100, а 500. Как изменится соотношение между количествами «удачных» и «неудачных» билетов?
3. Как повлияет на решение задачи добавление еще одной операции – возведение в степень? (Введем для этой операции обозначение « $\wedge$ »).

### Задание 6. Фибоначчиева система счисления

Эта знаменитая последовательность впервые была предложена в 1202 г. Леонардо Фибоначчи. В его «Книге о счете» есть такое упражнение: «Сколько пар кроликов получится от одной пары за год?». При этом предполагаются следующие допущения: каждая пара дает приплод – пару кроликов – каждый месяц; каждая новая пара становится плодоносящей в возрасте одного месяца; кролики никогда не умирают.

Всякая последовательность Фибоначчи состоит из натуральных чисел и задается указанием двух первых ее членов ( $F_1$  и  $F_2$ ), а каждый член, начиная с третьего, равен сумме двух, непосредственно ему предшествующих ( $F_k = F_{k-1} + F_{k-2}$ ). Задание фибоначчиевой системы счисления определяется указанием:

- цифрового множества  $\{0, 1\}$ ;
- базиса системы, в качестве которого берутся числа, образующие основную последовательность Фибоначчи ..., 34, 21, 13, 8, 5, 3, 2, 1.

Фибоначчиева система счисления позиционна, поэтому каждое число может быть записано в цифровой и многочленной форме:

$$101001 = 1 \times 13 + 0 \times 8 + 1 \times 5 + 0 \times 3 + 0 \times 2 + 1 \times 1 = 2410.$$

Всякое целое десятичное число всегда можно представить в виде равного ему числа, записанного в фибоначчиевой системе счисления:

$$3710 = 1 \times 34 + 1 \times 3 = 10000100.$$

1. Напишите программу замены десятичных чисел на равные им числа фибоначчиевой системы счисления. Приведите примеры ее работы.
2. Напишите программу дешифратор, преобразующую числа фибоначчиевой системы счисления в равные им десятичные числа. При вводе данных учтите тот факт, что в фибоначчиевой системе счисления две цифры 1 не должны записываться рядом (кстати, почему?)
3. Реализуйте сложение двух чисел в фибоначчиевой системе счисления.
4. Рассмотрите последовательность «слов Фибоначчи». Пусть  $S_1 = \langle a \rangle$ ,  $S_2 = \langle b \rangle$  и  $S_{n+2} = S_{n+1}S_n$ ,  $n > 0$ , т.е. слово  $S_{n+2}$  получается, если слово  $S_n$  приписать справа от слова  $S_{n+1}$ . Так, например,  $S_3 = \langle ba \rangle$ ,  $S_4 = \langle bab \rangle$ ,  $S_5 = \langle babba \rangle$  и т.д. Ясно, что слово  $S_n$  содержит  $F_n$  букв.

Исследуйте свойства слова  $S_n$  (где встречаются двойные буквы, можно ли предсказать  $k$ -ю букву, какова плотность буквы «а» и т.д.).

### Задание 7. Система счисления $\phi$ -чисел

Число  $\phi$  имеет очень интересную историю, Евклид назвал его «отношением крайнего и среднего». Отношение  $A$  к  $B$  равно отношению  $A+B$  к  $A$ , ес-

ли отношение А к В равно  $\varphi$ . В эпоху Возрождения это число называли «божественной пропорцией», а в прошлом столетии оно было названо «отношением золотого сечения», потому что в искусстве отношение  $\varphi$  считается эстетически самым благоприятным. Пропорция золотого сечения

$\varphi = \frac{1+\sqrt{5}}{2} \approx 1,6180339\dots$ . Известно несколько приближений к этому числу:

$$\varphi \approx \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{2}}}}}}; \quad \varphi \approx 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}}}$$

1. Сравните предложенные приближенные формулы и оцените их точность.
2. Рассмотрите действительные числа, записанные по основанию  $\varphi$  с помощью цифр 0 и 1 (например,  $100.1 = \varphi^2 + \varphi^{-1}$ ). Покажите, что существует бесконечное число способов представления в таком виде числа 1 (например,  $1=0.11=0.011111\dots$ ); но если потребовать, чтобы не встречались подряд две единицы и чтобы не появлялось бесконечной последовательности вида «01010101...», то каждое число имеет единственное представление.
3. Напишите программу замены действительных десятичных чисел на равные им числа системы счисления  $\varphi$ -чисел. Приведите примеры ее работы.
4. Напишите программу дешифратора, преобразующую числа системы счисления  $\varphi$ -чисел в равные им десятичные числа.
5. Реализуйте сложение и вычитание двух чисел в системе счисления  $\varphi$ -чисел.

### Задание 8. Дешифровка заданного сообщения

Для шифрования применен следующий код. Исходный текст без изменения порядка следования символов разбит на группы по три буквы каждая. Далее все первые буквы каждой из групп заменены на другие, смещенные на одно и то же число позиций в алфавите (по направлению к началу или концу) от положения шифруемой буквы. Таким образом перекодированы вторые и третьи буквы в группах с использованием других значений смещения (положительных или отрицательных). При этом величина сдвига для всех букв, расположенных на одних и тех же местах в группах, сохраняется одинаковой. Для сокращения времени расчетов величина сдвига ограничена значениями от  $-5$  до  $+5$ . Используемый алфавит состоит только из заглавных русских букв (кроме Ё) и как бы замкнут в кольцо, так что

если код шифруемой буквы оказывается, например, на 2 меньше кода буквы «А», то она заменяется на букву «Ю». Пробел считается буквой «Э». Считается, что в тексте сообщения она не встречается. Для расшифровки сообщения служит ключ – кодовое слово, которое заведомо есть в исходном тексте, но неизвестно где расположено.

Составить программу для автоматической шифровки и дешифровки кода. Выбор соответствующего действия производится через меню.

Исходные данные: зашифрованный текст сообщения и кодовое слово длиной до 20 букв.

### Задание 9. Римские цифры

Римскую запись чисел можно назвать десятиной непозиционной системой счисления. Десятичной – потому что в основе записи лежит тот же принцип счета десятками, что и в привычной нам десятичной системе. Это, в частности, означает что римскую запись любого числа можно разбить на группы символов, каждая из которых кодирует одну цифру десятичной записи этого числа. В отличие от нашей десятичной системы римская является непозиционной, потому что для каждого десятичного разряда используются свои обозначения, а не единый набор цифр, как в позиционной системе. В каждом разряде римская система имеет специальные символы для одной и для пяти единиц.

Вот как кодируются в римской записи обычные десятичные цифры и какие символы соответствуют им в различных разрядах:

1 – о  
2 – оо  
3 – ооо  
4 – оп  
5 – п  
6 – по  
7 – поо  
8 – пооо  
9 – од

	о	п
единицы	I	V
десятки	X	L
сотни	C	D
тысячи	M	

Здесь «о» (от Один) – символ, соответствующий единице соответствующего разряда, «п» (Пять) – пятерка, «д» (Десять) – единица следующего разряда. Цифра 0 не кодируется. Если в каком-то десятичном разряде стоит ноль, этот разряд просто пропускается при записи числа. Заметим, что благодаря непозиционной записи, то есть разным обозначениям в разных разрядах, такой пропуск не мешает однозначному прочтению числа.

В разряде тысяч пятерки нет. Это означает, что по стандартным правилам нельзя записать число, в котором в разряде тысяч будет цифра 4 или больше. Введем ограничение: будем рассматривать числа от 1 до 3999.

1. Написать программу, которая по введенному десятичному числу будет выдавать соответствующее в римской записи и наоборот.



2. Страницы книги пронумерованы римскими цифрами. Определить, сколько литер каждого вида потребуется для нумерации всех страниц книги от 1 до N. Число N вводится в десятичном виде,  $0 < N < 4000$ .

### Задание 10. Графический редактор «Калейдоскоп»

Разработать программу, имитирующую детский калейдоскоп. Основой картинки является равносторонний треугольник, в котором может перемещаться графический курсор.

Фигуры, нарисованные в треугольнике, повторяются на экране, как если бы они отражались в сторонах треугольника, как в зеркале. Предусмотреть несколько инструментов графического редактора: рисование линий, графических примитивов, закраску областей цветом.

### Задание 11. Клеточные автоматы и машинная графика

Жизнь – это многоклеточное сообщество, населяющее пустыни. Пустыня представляет собой квадратную решетку, каждая ячейка которой вмещает одну клетку Жизни. Мерой течения времени служит смена поколений Жизни, приносящая в колонию клеток смерть и рождение.

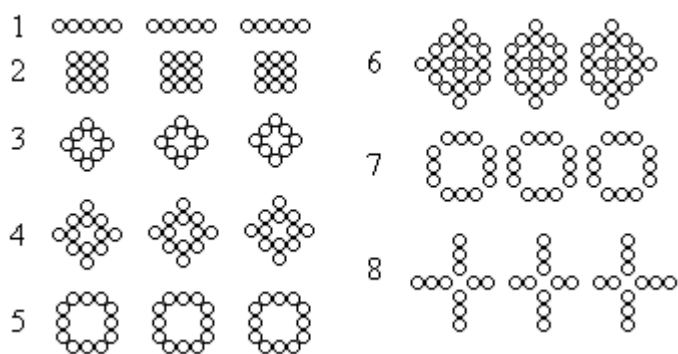
Чтобы проследить за историей развития колонии, разместим в пустыне клетки Жизни в их начальном положении. Смена поколений будет происходить по следующим правилам.

1. Соседями клетки считаются все клетки, находящиеся в восьми ячейках, расположенных рядом с данной по горизонтали, вертикали или диагонали.
2. Если у некоторой клетки меньше двух соседей, она погибает от одиночества. Если клетка имеет больше трех соседей, она погибает от тесноты.
3. Если рядом с пустой ячейкой окажется ровно три соседние клетки Жизни, то в этой ячейке рождается новая клетка.
4. Гибель и рождение происходят в момент смены поколений. Таким образом, гибнущая клетка может способствовать рождению новой, но рождающаяся клетка не может воскресить гибнущую, и гибель одной клетки, уменьшив локальную плотность населения, не может предотвратить гибель другой.

Напишите программу, моделирующую колонию Жизни. Исходными данными служит начальное расположение клеток, а в качестве результата нужно получить вид сверху всех поколений колонии.

История одной колонии Жизни. Номер поколения выписан слева от каждой картинке.

Хотя этого и не видно из примеров, некоторые колонии разрастаются невероятным образом при весьма скромных начальных размерах. Есть другие колонии, которые мед-



ленно перемещаются по пустыне, переходя на все новые и новые территории. Растущая или движущаяся колония может выйти из поля зрения, если его положение и границы зафиксированы, поэтому, вероятно, понадобится еще и метод вывода, перемещающий нашу точку зрения вслед за изменениями колонии.

Развитие темы. Колония может все время расти, непрерывно меняя свое расположение, форму или число клеток. Однако чаще колония становится в конце концов стационарной, начиная циклически повторять один и тот же конечный набор состояний. Длина цикла называется периодом колонии. (По этому определению период мертвой и пустой колонии равен единице.) Измените вашу программу так, чтобы она выявляла стационарные колонии и сообщала о них.

История колонии Жизнь зачаровывает, если ее просматривать как фильм, но она будет еще увлекательней, если предстанет в цвете. Каждой клетке при рождении может быть приписан некоторый цвет, определяемый, возможно, ее поколением или генами, переданными ей родителями. Циклические, но при этом движущиеся колонии (а таких немало!) великолепны в своем сверкающем многоцветном наряде.

Любая колония имеет преемника, но не у каждой есть предшественник. Такие изолированные колонии называются садами Эдема. Сад Эдема можно увидеть, только если поместить его на плоскость в качестве начальной конфигурации. Подумайте, как использовать вашу программу для нахождения сада Эдема.

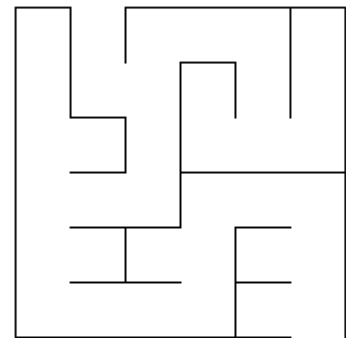
### **Задание 12. Автоматическое построение «лабиринтов»**

Тезей должен был найти выход из Критского лабиринта или погибнуть от руки Минотавра. Но что поразительно: найти вход в лабиринт – задача не менее трудная.

Здесь не представляется возможным описать все мыслимые лабиринты, да это и не требуется. Мы займемся простыми лабиринтами, построенными на прямоугольнике  $m \times n$ , где  $m, n$  – положительные целые числа.

Внутри и на границах прямоугольника поставлены стенки по ребрам покрывающей его единичной квадратной сетки. Чтобы построить из прямоугольника лабиринт, выьем одну единичную стенку на одной из сторон прямоугольника (получится вход в лабиринт); выьем одну единичную стенку на противоположной стороне (получится выход) и еще удалим какое-то число строго внутренних стенок.

Говорят, что лабиринт имеет решение, если между входом и выходом внутри лабиринта есть путь в виде ломаной, не имеющей общих точек со стенками. Решение единственно, если любые два таких пути проходят через одни и те же внутренние ячейки сетки. На рисунке приведен пример лабиринта  $6 \times 6$ .



Напишите программу, которая по исходным данным  $m$  и  $n$  строит прямоугольный лабиринт  $m \times n$  (проверьте, допустимы ли заданные  $m$  и  $n$ ). Предусмотрите, чтобы программа при каждом обращении к ней порождала разные лабиринты. Лабиринт должен иметь единственное решение, и, чтобы получившийся лабиринт был интересным, все ячейки должны быть соединены с основным путем, дающим решение.

Теоретически нельзя удовлетворить требованию, чтобы любые два лабиринта (даже при одинаковых  $m$  и  $n$ ) были различны, поскольку существует лишь конечное число лабиринтов любого наперед заданного размера, а программу можно вызвать большее число раз. Однако число лабиринтов какого-нибудь размера очень велико, и поэтому вероятность повторения лабиринта можно сделать очень маленькой.

Практически это достигается, если программа будет производить «случайный» выбор различных вариантов, опираясь на какое-либо доступное ей, но неуправляемое значение (обычно берут дату и время вызова программы). Варианты, между которыми выбирает программа, это, например, положение входа и выхода и положение хотя бы нескольких внутренних разрушаемых стенок.

Один из возможных подходов к решению таков.

Выбираем вход; затем, начав от него, добавляем по одной ячейке к главному пути-решению, пока он не достигнет выходной стороны. После этого удаляем некоторые внутренние стенки так, чтобы все клетки оказались соединенными с главным путем.

Чтобы главный путь не получился прямым коридором, следует при его построении предусмотреть случайные повороты.

### **Литература**

1. Ахо А.В., Хопкрофт Дж.Э., Ульман Дж.Д. Структуры данных и алгоритмы. – М.: Издательский дом «Вильямс», 2003. – 384 с.
2. Бондарев В.М., Рублинецкий В.И., Качко Е.Г. Основы программирования. – Харьков: Фолио; Ростов н/Д: Феникс, 1997. – 368 с.
3. Брудно А.Л., Каплан Л.И. Московские олимпиады по программированию. – М.: Наука, 1990. – 208 с.
4. Макконнелл Дж. Основы современных алгоритмов. – М.: Техносфера, 2004. – 268 с.
5. Немнюгин С.А. Turbo Pascal. – СПб: Изд-во «Питер», 2000. – 496 с.

### 3. ПРАКТИКУМ ПО РЕШЕНИЮ ЗАДАЧ В MS EXCEL

#### Составление отчетов, консолидация данных и построение сводных таблиц

##### 1. Вычисление промежуточных итогов

MS Excel позволяет автоматически составлять итоговые отчеты. Для этого используется команда меню *Данные/Итоги*. Рассмотрим пример.

В таблице 1 перечислены периферийные устройства к компьютерам и офисное оборудование, которые закупила некоторая организация.

Таблица 1

Товар	Тип	Модель	Цена	Кол-во	Всего
Принтер	лазерный	LBP-3200	155	3	465
Принтер	лазерный	Samsung ML-1615	163	2	326
Принтер	лазерный	XEROX Phaser 3122	178	1	178
Принтер	матричный	EPSON LX-300+	175	5	875
Принтер	матричный	EPSON FX-2190	578	1	578
Принтер	струйный	Canon PIXMA iP1500	74	6	444
Принтер	струйный	HP DeskJet 3940	62	10	620
Принтер	струйный	Lexmark Z617	68	2	136
Сканер	планшетный	ScanJet 5590C	499	1	499
Сканер	планшетный	Mustek BearPaw 1200	51	4	204
Сканер	планшетный	Mustek ScanExpress	187	2	374
Телефон	проводной	Panasonic KX-T7565	73	2	146
Телефон	проводной	Siemens 815S	29	10	290
Телефон	радиотелефон	Panasonic KX-TC2105	39	12	468
Телефон	радиотелефон	Panasonic KX-TCD 815	170	6	1020
Телефон	радиотелефон	Siemens Gigaset C345	98	5	490
Факсы	термопленка	PANASKX-FC243RU	351	2	702
Факсы	лазерный	PANASKX-FL513RU	365	2	730
Факсы	лазерный	PANASKX-FL543RU	414	1	414
Факсы	термобумага	PANASKX-FT904RU	241	1	241

**Задание 1.** Вычислить промежуточные итоги (количество и общую стоимость) по каждому виду товара.

**Выполнение задания.** Для автоматического вычисления итогов таблица должна быть отсортирована по виду товара. Далее ее нужно выделить и выбрать команду меню *Данные/Итоги*. Появившееся диалоговое окно *Промежуточные итоги* требуется заполнить следующим образом:

- в поле «При каждом изменении» нужно указать, по какому столбцу вычисляются итоги (столбец Товар);
- в качестве операции выбрать сумму;

- установить флажки напротив строк «Количество» и «Всего» в поле добавления итогов;
- установить флажки напротив строк «Заменить текущие итоги» и «Итоги под данными»;
- Щелкнуть по кнопке ОК.

В результате будут получены итоговые значения количества и суммы для каждого вида товара, приобретенного организацией:

1	2	3	1	2	3	4	5	6
			Товар	Тип	Модель	Цена	Кол-во	Всего
•	2		Принтер	лазерный	LBP-3200	155	3	465
•	3		Принтер	лазерный	Samsung ML-1615	163	2	326
•	4		Принтер	лазерный	XEROX Phaser 3122	178	1	178
•	5		Принтер	матричный	EPSON LX-300+	175	5	875
•	6		Принтер	матричный	EPSON FX-2190	578	1	578
•	7		Принтер	струйный	Canon PIXMA iP1500	74	6	444
•	8		Принтер	струйный	HP DeskJet 3940	62	10	620
•	9		Принтер	струйный	Lexmark Z617	68	2	136
-	10		<b>Принтер Итог</b>				30	3622
•	11		Сканер	планшетный	ScanJet 5590C	499	1	499
•	12		Сканер	планшетный	Mustek BearPaw 1200	51	4	204
•	13		Сканер	планшетный	Mustek ScanExpress	187	2	374
-	14		<b>Сканер Итог</b>				7	1077
•	15		Телефон	проводной	Panasonic KX-T7565	73	2	146
•	16		Телефон	проводной	Siemens 815S	29	10	290
•	17		Телефон	радиотелефон	Panasonic KX-TC2105	39	12	468
•	18		Телефон	радиотелефон	Panasonic KX-TCD 815	170	6	1020
•	19		Телефон	радиотелефон	Siemens Gigaset C345	98	5	490
-	20		<b>Телефон Итог</b>				35	2414
•	21		Факсы	термопленка	PANASKX-FC243RU	351	2	702
•	22		Факсы	лазерный	PANASKX-FL513RU	365	2	730
•	23		Факсы	лазерный	PANASKX-FL543RU	414	1	414
•	24		Факсы	термобумага	PANASKX-FT904RU	241	1	241
-	25		<b>Факсы Итог</b>				6	2087
-	26		<b>Общий итог</b>				78	9200

Для отмены действий и удаления промежуточных итогов нужно выбрать команду меню *Данные/Итоги*, кнопка «Убрать все».

MS Excel автоматически структурирует лист рабочей книги таким образом, что при подведении промежуточных итогов взаимосвязанные данные объединяются в группы. Управление группами осуществляется кнопками уровней 

1	2	3
---	---	---

, которые находятся в левом верхнем углу экрана под адресным полем. Для сокрытия или отображения групп данных также используются кнопки показа деталей структуры «+» и «-».

В рассмотренном примере на рабочем листе имеется три структурных уровня:

- первый состоит из одной позиции и определяет общее количество товара всех видов, а также суммарное значение затрат;
- второй уровень разбивает товары на группы по наименованию и выдает суммы по виду товара;
- третий уровень отображает все элементы списка.

Например, если требуется показать детальную информацию по сканерам и общие итоги по всем остальным видам товара, результирующая таблица будет выглядеть следующим образом:

1	2	3	4	5	6		
1	Товар	Тип	Модель	Цена	Кол-во	Всего	
+	10	<b>Принтер Итог</b>			30	3622	
-	11	Сканер	планшетный	ScanJet 5590C	499	1	499
-	12	Сканер	планшетный	Mustek BearPaw 1200	51	4	204
-	13	Сканер	планшетный	Mustek ScanExpress	187	2	374
-	14	<b>Сканер Итог</b>			7	1077	
+	20	<b>Телефон Итог</b>			35	2414	
+	25	<b>Факсы Итог</b>			6	2087	
-	26	<b>Общий итог</b>			78	9200	

**Задание 2.** Вычислить промежуточные итоги согласно приведенным в таблице 2 вариантам индивидуальных заданий. Используя кнопки управления группами и кнопки показа деталей структуры отобразить на экране полученный итоговый отчет.

Таблица 2

№	Задание для индивидуального выполнения
1	Отобразить в таблице стоимость закупки по каждому из приведенных типов принтера: лазерному, матричному, струйному.
2	Отобразить в таблице среднее значение сумм, потраченных на покупку телефонов и факсов по их типам.
3	Отобразить в таблице количество и стоимость закупки сканеров по их типам.
4	Отобразить в таблице количество приобретенных принтеров: лазерных, матричных, струйных.
5	Отобразить в таблице количество и стоимость закупки всех телефонов и факсов.
6	Отобразить в таблице стоимость всех принтеров и сканеров, а также общую сумму, потраченную на их закупку.
7	Отобразить в таблице среднюю цену принтеров по их типам (лазерные, матричные, струйные).
8	Отобразить в таблице минимальные цены для телефонов и факсов по их типам.
9	Отобразить в таблице максимальную цену по каждому типу принтера.
10	Отобразить в таблице количество и стоимость приобретенных сканеров и факсов.

## 2. Построение консолидированных отчетов

Консолидация данных – это процедура получения обобщенных данных из нескольких областей. Источники данных могут находиться на том же листе, что и итоговая таблица, на других листах той же книги или в других книгах. Данные в этих областях должны быть единообразно организованы, заголовки столбцов должны быть одинаковыми для всех областей.

Для выполнения следующего задания необходимо создать новую книгу и скопировать данные по принтерам на Лист 1, по сканерам на Лист 2, по телефонам и факсам на Лист 3. На все листы нужно скопировать заголовки из исходной таблицы.

**Задание 3.** Построить консолидированный отчет по закупкам периферийного оборудования к компьютерам и офисной техники.

Для получения таблицы необходимо выполнить следующие действия.

- Добавить к рабочей книге еще один лист и оформить на нем заголовок отчета, например:

	A	B	C	D	E	F	G	
1			<b>Отчет о закупке техники</b>					
2								
3		<b>Товар</b>	<b>Тип</b>	<b>Модель</b>	<b>Цена</b>	<b>Кол-во</b>	<b>Всего</b>	
4								

- Выделить ячейку B3, определяющую верхний левый угол консолидированной таблицы и выполнить команду меню **Данные/Консолидация**.
- В диалоговом окне **Консолидация** выбрать функцию **Сумма** (другие способы подведения итогов содержатся в раскрывающемся списке окна **Функция**).
- Установить курсор в поле **Ссылка**.
- Последовательно выделить все блоки ячеек на различных листах рабочей книги, содержащие исходные данные для обобщенной таблицы, нажимая после задания каждого диапазона кнопку «Добавить» в окне **Консолидация**. Все исходные таблицы будут занесены в список диапазонов.
- Установить флажки напротив строк «подписи верхней строки» и «значения левого столбца».
- Щелкнуть по кнопке ОК.

Итоговые данные на всех листах будут сведены в единую таблицу.

Примечание: для автоматического обновления данных области консолидации при изменении исходных областей нужно установить флажок «Создавать связи с исходными данными».

Итоговый отчет будет выглядеть следующим образом:

	A	B	C	D	E	F	G
1			<b>Отчет о закупке техники</b>				
2							
3		<b>Товар</b>	<b>Тип</b>	<b>Модель</b>	<b>Цена</b>	<b>Кол-во</b>	<b>Всего</b>
4		Принтер			1453	30	3622
5		Сканер			737	7	1077
6		Телефон			409	35	2414
7		Факсы			1371	6	2087

При необходимости можно скрыть столбцы с пустыми ячейками командой меню **Формат, Столбец, Скрыть**, предварительно выделив их.

Следует иметь в виду, что если не установлен флажок «Создавать связи с исходными данными», в обобщенной таблице автоматический пересчет данных не производится, то есть изменение данных в одной из исходных таблиц не приведет к изменению значений в итоговой таблице. Для того, чтобы обновить данные в итоговой таблице, необходимо запустить консолидацию еще раз, выделив предварительно итоговую таблицу.

### **Сортировка, фильтрация, реализация запросов к базе данных с использованием функций категории работы с базой данных**

Имеется база данных (БД) по ассортименту и наличию в продаже литературы в некотором книжном магазине (файл Книги.xls), созданная в среде табличного процессора MS Excel (см. Приложение 1). Наименования полей базы данных поясняются в таблице 3.

Таблица 3

#### **Наименования полей базы данных по литературе**

<b>Столбец</b>	<b>Наименование поля</b>	<b>Пояснения</b>
A	Тема	Тема издания
B	Автор	Автор книги
C	Название	Название издания
D	Город	Город, в котором находится издательство
E	Издательство	Название издательства
F	Год	Год издания
G	Страниц	Количество страниц в книге
H	Тираж	Тираж издания
I	Цена	Цена одного экземпляра
J	Количество	Количество экземпляров, имеющих в наличии в торговом зале и на складе магазина

#### **1. Сортировка**

Для упорядочения ячеек по значениям (без учета формата) предусмотрен определенный порядок сортировки.

При сортировке по возрастанию используется следующий порядок:



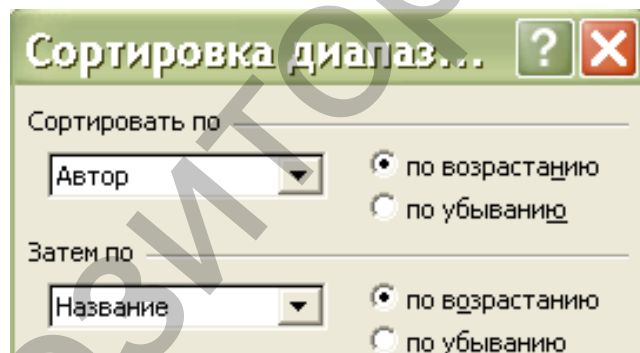
- числа сортируются от наименьшего отрицательного до наибольшего положительного числа;
- текст, в том числе содержащий числа, сортируется в следующем порядке: 0 1 2 3 4 5 6 7 8 9 ' - ! " # \$ % ( ) \* , . / : ; ? @ [ \ ] ^ \_ { | } ~ + < + > А В С D E F G H I J K L M N O P Q R S T U V W X Y Z А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я;
- логическое значение ЛОЖЬ предшествует значению ИСТИНА;
- пустые ячейки всегда помещаются в конец списка.

При сортировке по убыванию все перечисленные порядки заменяются на обратные за исключением пустых ячеек, которые всегда помещаются в конец списка.

**Задание 1.** Провести двухуровневую сортировку базы данных согласно критериям, изложенным в таблице 4 (по умолчанию рассматривается сортировка в порядке возрастания критерия).


**Пример выполнения задания.** Провести двухуровневую сортировку БД, используя критерии: первичный – по автору, вторичный – по названию (оба критерия – по возрастанию). Сценарий сортировки:

- 1) выделение диапазона ячеек, занимаемого исходной базой данных;
- 2) вызов команды инструментального меню *Данные/Сортировка...*;
- 3) пример заполнения диалогового окна *Сортировка диапазона:*



- 4) визуальный контроль результатов сортировки; ниже приведен начальный фрагмент БД после сортировки:

	А	В	С	Д
1	<b>Тема</b>	<b>Автор</b>	<b>Название</b>	<b>Город</b>
2	образование	Бартон В.И.	Логика: Учебное пособие	Минск
3	образование	Берков В.Ф.	История логики	Минск
4	культурология	Берков В.Ф.	Культура диалога	Минск
5	образование	Берков В.Ф.	Логика. Учебник для вузов	Минск
6	история	Бобраков А.В.	Вокруг Европы по волнам истории	Москва

- 5) с целью подготовки к выполнению следующего задания необходимо выполнить отмену результатов сортировки, например, щелчком мышью по кнопке  на стандартной панели инструментов или с помощью сочетания клавиш Ctrl+z.

## Варианты индивидуальных заданий

№	Критерии сортировки	
	Первичный	Вторичный
1	По городу (по убыванию)	По названию издательства
2	По названию книги	По убыванию тиража издания
3	По теме (по убыванию)	По названию книги
4	По автору (по убыванию)	По цене издания
5	По названию книги	По убыванию количества страниц
6	По названию издательства	По количеству экземпляров
7	По теме (по убыванию)	По тиражу издания
8	По году издания	По убыванию цены издания
9	По количеству экземпляров	По городу (по убыванию)
10	По автору (по убыванию)	По названию издательства


## 2. Использование автофильтра

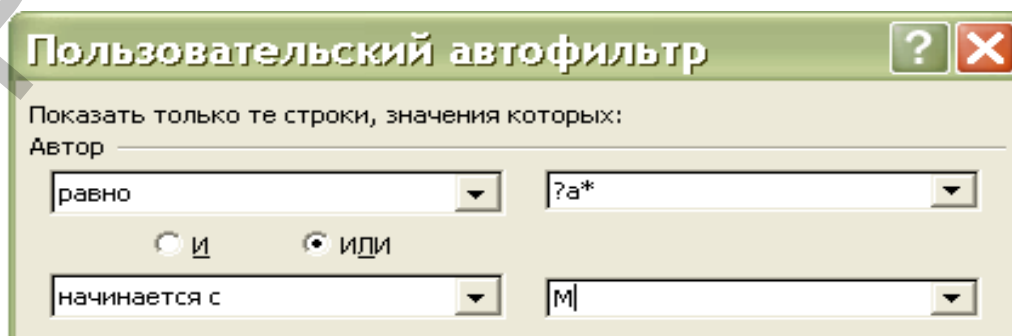
**Задание 2.** Используя операцию автофильтра, провести выборку записей из БД согласно приведенным в таблице 5 критериям фильтрации. Подробно по шагам описать необходимые действия пользователя.

**Пример выполнения задания.** Используя операцию автофильтра, провести выборку записей из БД согласно критерию – фамилии авторов, начинающиеся с буквы «М» или имеющие вторую букву фамилии – «а». Сценарий запроса к БД:

- 1) выделение диапазона ячеек, занимаемого исходной базой данных;
- 2) вызов команды меню *Данные/Фильтр/Автофильтр*; на рисунке приведен начальный фрагмент БД после включения автофильтра с преобразованием всех наименований полей в раскрывающиеся списки:

	А	В	С
1	Тема	Автор	Название
2	образование	Бартон В.И.	Логика: Учебное пособие
3	образование	Берков В.Ф.	История логики

- 3) выбор щелчком по кнопке  в раскрывающемся списке поля *Автор* позиции (*Условие...*);
- 4) пример заполнения диалогового окна *Пользовательский автофильтр*:



Пользовательский автофильтр

Показать только те строки, значения которых:

Автор

равно ?a\*

И  ИЛИ

начинается с М

Примечание: знак «?» – обозначает любой единичный знак, «\*» – последовательность любых символов. В данном примере запись «?а\*» означает, что будут показаны строки, в которых первая буква – любая, вторая буква – «а», далее – произвольный набор символов. Равносильная запись – «начинается с ?а».

- 5) визуальный контроль результатов фильтрации;
- 6) с целью подготовки к выполнению следующего задания – отмена результатов фильтрации посредством выбора в раскрывающемся списке поля *Автор* позиции (*Все*).

Таблица 5

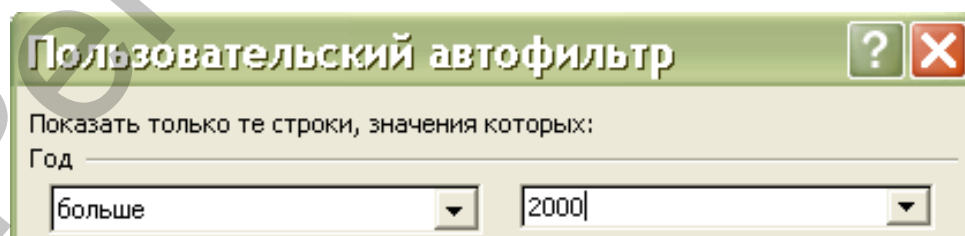
### Варианты индивидуальных заданий

№	Критерии фильтрации
1	Фамилия автора начинается с «По» или буквы «Ш»
2	Количество экземпляров – менее 10 или более 30
3	Издательство – «МГУ», «ЕГУ», «БГУ» или «Университетское»
4	Название содержит слова «Учебник», «Учебное» или «Основы»
5	Фамилия автора начинается с «Л», третья буква – «к»
6	Тираж – более 500, но менее 2500 экземпляров
7	Год издания – между 2000-м и 2003-м
8	Тема издания – психология, педагогика или образование
9	Цена издания более 7000, но менее 10000 рублей
10	Фамилия автора начинается с «Ко» или буквы «В»

**Задание 3.** Используя многошаговую операцию автофильтра, провести выборку записей из БД согласно приведенным в таблице 6 критериям фильтрации. Подробно по шагам описать необходимые действия пользователя.

**Пример выполнения задания.** Используя многошаговую операцию автофильтра, провести выборку записей из БД согласно критериям – книги по истории, изданные после 2000 года. Сценарий запроса к БД:

- 1) выбор в раскрывающемся списке поля *Тема* позиции *история*;
- 2) выбор в раскрывающемся списке поля *Год* позиции (*Условие...*)
- 3) пример заполнения диалогового окна *Пользовательский автофильтр*:



- 4) визуальный контроль результатов фильтрации;
- 5) с целью подготовки к выполнению следующего задания – отмена результатов фильтрации посредством выбора в инструментальном меню пунктов *Данные/Фильтр/Отобразить все* и выключение автофильтра повторным выбором пунктов меню *Данные/Фильтр/Автофильтр*.

### Варианты индивидуальных заданий

№	Критерии фильтрации
1	Книги по лингвистике, изданные в Минске
2	Книги, изданные в издательстве «Новое знание» в 2002 году
3	Книги 2003 года издания, изданные в Москве
4	Книги, изданные в Минске, количество – не более 10 экземпляров
5	Книги по психологии, тираж которых больше 1000 экземпляров
6	Книги ценой более 10000, изданные в Могилеве
7	Книги по философии, количество страниц – более 200
8	Книги автора Беркова В.Ф., изданные в 2002 году
9	Книги по педагогике издательства ТетраСистемс
10	Книги, изданные в Москве тиражом не менее 2000 экземпляров

**Задание 4.** Используя операцию расширенного фильтра, выполнить одношаговую фильтрацию согласно критериям задания 3. Соответствующий блок критериев расположить над таблицей БД. Подробно по шагам описать необходимые действия пользователя.

**Пример выполнения задания.** Используя операцию расширенного фильтра, выполнить одношаговую фильтрацию согласно критериям – книги по истории, изданные после 2000 года.

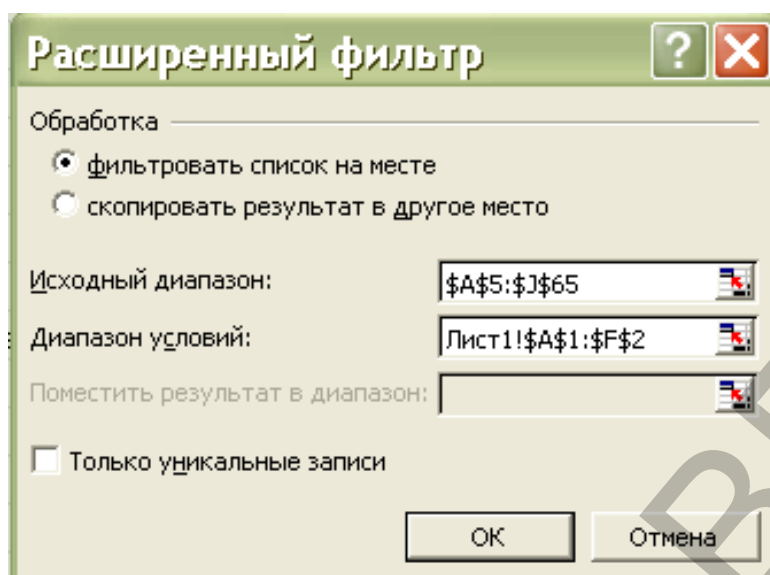
Чтобы отфильтровать список с помощью расширенного фильтра, перед исходной таблицей необходимо вставить несколько строк и в первую строку скопировать заголовок таблицы. Далее в нижележащие строки вводится условие отбора. Между диапазоном условий и фильтруемым списком следует оставить по крайней мере одну пустую строку.

Следует иметь в виду, что комбинированный критерий фильтрации формируется из частных критериев в отдельных ячейках блока по правилу: объединение критериев в строке осуществляется логической операцией **И**, в столбце – логической операцией **ИЛИ**. Сценарий запроса к БД:

- 1) резервирование диапазона ячеек для размещения блока критериев, для этого требуется вставить четыре строки над таблицей исходной БД;
- 2) копирование строки с наименованиями полей БД в первую строку блока критериев;
- 3) внесение во вторую строку блока критериев условий выборки записей, как это изображено на следующем рисунке:

	А	В	С	Д	Е	Ф
1	Тема	Автор	Название	Город	Издательство	Год
2	история					>2000

- 4) выделение диапазона ячеек исходной БД;
- 5) вызов команды меню *Данные/Фильтр/Расширенный фильтр...*
- 6) пример заполнения диалогового окна *Расширенный фильтр*:



- 7) визуальный контроль результатов фильтрации;
- 8) с целью подготовки к выполнению следующего задания – отмена результатов фильтрации посредством выбора в инструментальном меню пунктов *Данные/Фильтр/Отобразить все*.

### 3. Использование функций категории *Работа с базой данных*

**Задание 5.** Реализовать запрос к БД, используя функции категории *Работа с базой данных*. Подробно по шагам описать необходимые действия пользователя. Варианты запросов приведены в таблице 7.

Задание 5 предполагает использование функций категории *Работа с базой данных*, которые применяют механизм расширенной фильтрации с последующей обработкой ее результатов:

<p><b>БДСУМ</b></p>	<p>– суммирование значений в указанном столбце;            Синтаксис: <b>БДСУММ (БД; поле; критерий)</b>  <b>БД</b> – интервал ячеек, составляющих список или базу данных;  <b>поле</b> – столбец, используемый функцией. Аргумент «поле» может быть задан как текст с названием столбца, либо как число, задающее положение столбца в списке: 1 – для первого поля, 2 – для второго и т.д.;  <b>критерий</b> – интервал ячеек, содержащих задаваемые условия. Любой интервал ячеек, содержащий по крайней мере одно название столбца и по крайней мере одну ячейку под названием столбца с условием, может быть использован в качестве аргумента <b>критерий</b></p>
<p><b>БСЧЁТ</b></p>	<p>– подсчет числа значений в указанном столбце, который должен содержать не текстовые значения; для этой функции и всех последующих аргументы аналогичны функции БСЧЁТ;</p>

<b>ДМАКС</b>	– нахождение максимального значения в указанном столбце;
<b>ДМИН</b>	– нахождение минимального значения в указанном столбце;
<b>ДСРЗНАЧ</b>	– вычисление среднеарифметического значения в указанном столбце.

Все вышеперечисленные функции имеют три аргумента:

- диапазон ячеек, занимаемых исходной БД;
- ячейка с именем поля-столбца, по которому после фильтрации производится суммирование, подсчет числа значений, поиск максимума и прочее;
- диапазон ячеек блока критериев фильтрации.

Следует иметь в виду, что при использовании функции БСЧЁТ в качестве имени поля, по которому производится подсчет числа записей, прошедших фильтрацию, следует указать поле не текстового типа, например, арифметического (см.ниже пример выполнения задания).

**Пример выполнения задания.** Реализовать запрос к БД, используя функции категории *Работа с базой данных*. Привести расчетную формулу для подсчета книг по истории и философии, изданных в Минске. Сценарий запроса к БД:

1) заполнение блока критериев новыми условиями выборки записей, как это показано на рисунке:

	А	В	С	Д
1	<b>Тема</b>	<b>Автор</b>	<b>Название</b>	<b>Город</b>
2	история			Минск
3	философия			Минск

2) ввод в ячейку **А68** (под таблицей исходной БД) с помощью *Мастера функций* соответствующей расчетной формулы, для этого необходимо в категории функций *Работа с базой данных* выбрать функцию БСЧЁТ. Пример заполнения диалогового окна:

**Аргументы функции** ? X

БСЧЁТ

База\_данных:  = {"Тема";"Автор";"Название";"Город"}

Поле:  = "Год"

Критерий:  = A1:D3

= 10

Подсчитывает количество числовых ячеек в выборке из заданной базы данных по заданному критерию.

**Критерий** диапазон, содержащий условие базы данных. Диапазон включает заголовки столбцов и одну ячейку с условием.

- 3) завершение диалога с *Мастером функций*, в результате чего в ячейку **A68** должна быть введена формула: **=БСЧЁТ(A5:J65;F5;A1:D3)**, где **F5** – ячейка имени поля с арифметическими значениями, используемыми для подсчета числа записей, удовлетворяющих условиям фильтрации;
- 4) просмотр будет результат запроса к БД в ячейке **A68** (равен 10).

Таблица 7

### Варианты индивидуальных заданий

№	Запрос к базе данных
1	Количество книг по философии, изданных в Москве и Могилеве
2	Количество экземпляров книг по образованию и искусству, имеющихся на складе книжного магазина
3	Среднее количество страниц в книгах, опубликованных в издательстве ТетраСистемс в 2002 году
4	Максимальную стоимость книг с тиражом более 1000 экземпляров
5	Количество книг, изданных в издательстве «Новое знание» в 2002 и 2004 году, имеющихся на складе книжного магазина
6	Количество книг, изданных в издательстве «Амалфея» 2000 году с количеством страниц более 400
7	Среднюю цену книг по лингвистике, опубликованных в 2001 году
8	Общее количество книг, изданных в издательстве «ЕГУ» тиражом 1000 экземпляров, имеющихся на складе магазина
9	Суммарный тираж книг по философии, опубликованных в Минске и Могилеве
10	Минимальный тираж книг, изданных в издательстве МГУ города Могилева

## **Построение диаграмм типа «Поверхность» и анимированных графиков**

### **1. Построение диаграмм типа «Поверхность»**

При создании двумерных графиков обычно требуется задать несколько рядов данных, расположенных в строках или столбцах. Один из рядов задает значения абсцисс, остальные – соответствующие значения ординат точек. При создании объемных диаграмм данные задаются в прямоугольном блоке ячеек, где левый столбец задает значения по оси X, верхняя строка – значения по оси Y, данные внутри блока – соответствующие значения по оси Z.

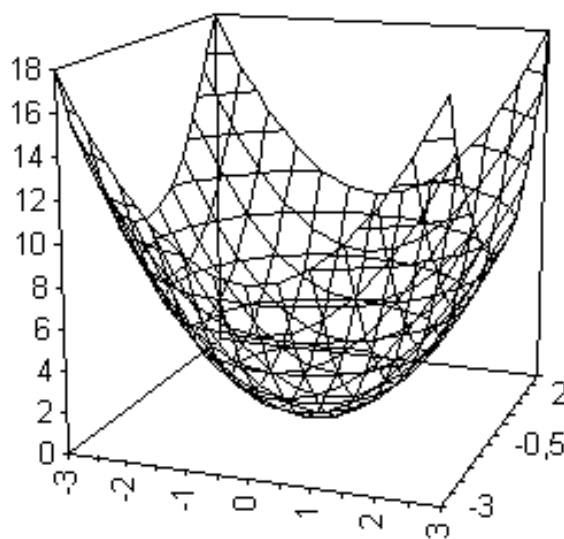
**Задание 1.** Построить график поверхности, заданной уравнением. Варианты заданий приведены в таблице 8.

**Пример выполнения задания.** Построить график поверхности, заданной уравнением  $z(x, y) = x^2 + y^2$  на прямоугольной области определения  $x \in [-3, 3]$ ;  $y \in [-3, 3]$ .

Для решения данной задачи необходимо подготовить блок ячеек, задающих значения  $x$ ,  $y$  и  $z$ . В диапазон ячеек A2:A14 нужно ввести значения абсцисс от -3 до 3 с шагом 0,5. Аналогичным образом в диапазон ячеек B1:N1 помещаются значения ординат. В ячейку B2 вводится формула  $=A2^2+B1^2$  и копируется на диапазон B3:B14. Далее в расположенных в ячейках B2:B14 формулах необходимо поменять относительные адреса ячеек, на которые ссылается формула, на абсолютные, а абсолютные адреса – на относительные. Это необходимо для того, чтобы при горизонтальном распространении формул ссылки в формулах на столбец A2:A14 не менялись (поменять относительные адреса ячеек на абсолютные и наоборот можно следующим образом: выделить ячейку, в которой необходимо поменять адреса, щелкнуть мышью в строке формул по адресу, подлежащему изменению, и, нажимая несколько раз клавишу F4, изменить адрес на тот, который нужен). После выполненной замены блок ячеек B2:B14 копируется на диапазон C2:N14. Таким образом, в диапазоне ячеек B2:N14 будет получен блок исходных данных.

Для построения графика поверхности следует выделить все данные и в *Мастере диаграмм* выбрать тип диаграммы *Поверхность*. Вид графика поверхности представлен на рисунке.

Изменение внешнего вида объемной диаграммы выполняется сле-





дующим образом: в меню *Диаграмма* следует выбрать команду *Объемный вид* и установить требуемые параметры. Если флажок *Изометрия* снят, можно задать параметры «Перспектива» и «Высота».

Для изменения угла поворота и возвышения объемной диаграммы нужно с помощью мыши щелкнуть пересечение любых двух осей, чтобы выделить углы диаграммы, и перетащить угол.

Таблица 8

<b>Варианты индивидуальных заданий</b>	
<b>№</b>	<b>Уравнение функции двух переменных</b>
1	$z(x, y) = e^{\cos(y)} \cdot \sin(x), x \in [-5, 5]; y \in [-5, 5]$
2	$z(x, y) = \cos(x) + y^3, x \in [-5, 5]; y \in [-5, 5]$
3	$z(x, y) = \sin(x) - \frac{x+2y}{15}, x \in [-4, 4]; y \in [-4, 4]$
4	$z(x, y) = \frac{x^2}{2} - \frac{y^2}{5}, x \in [-5, 5]; y \in [-5, 5]$
5	$z(x, y) = \sin\left(\sqrt{\frac{1}{3} x \cdot y }\right), x \in [-5, 5]; y \in [-5, 5]$
6	$z(x, y) = e^{\cos(x)} + \frac{1}{10}y, x \in [-5, 5]; y \in [-5, 5]$
7	$z(x, y) = \cos(x^2) + \frac{1}{2}y, x \in [-2, 2]; y \in [-2, 2]$
8	$z(x, y) = e^{\cos(x)} \cdot e^{\sin(y)}, x \in [-2, 5]; y \in [-2, 5]$
9	$z(x, y) = \sin(x) \cdot e^{\sin(y)}, x \in [-4, 4]; y \in [-4, 4]$
10	$z(x, y) = \frac{x + y^2}{\cos(y)}, x \in [-3, 3]; y \in [-3, 3]$

## 2. Построение анимированных графиков

Иногда в иллюстративных целях нужно показать не просто математическую зависимость между переменными в виде графика, а движущееся изображение. Например, при изучении физики можно показать движение маятника или движение планет по орбитам средствами электронной таблицы. Обычный график можно «оживить» следующим образом. В выделенные ячейки по очереди записываются пары значений переменных  $X$  и  $Y$ . Если построить точечный график по этим двум ячейкам (то есть на графике будет отображена только одна точка), то при смене значений в выделенных

ячейках точка на графике будет перемещаться и, таким образом, можно получить «анимированную картинку» движения точки.

Примечание: на диаграмме нужно изменить формат осей, убрав в опции «шкала» режим автоподбора минимума и максимума, и установить для них фиксированные значения.

**Задание 2.** Изобразить движение точки по заданной траектории. Варианты заданий приведены в таблице 9.

**Пример выполнения задания.** Изобразить движение точки по окружности произвольного радиуса.

Известно, что изображение окружности можно получить с помощью параметрического уравнения  $x = R \cdot \cos f$ ,  $y = R \cdot \sin f$ , где  $x$  и  $y$  – координаты точки,  $R$  – радиус окружности,  $f$  – угол. Задавая значение угла от 0 до  $2\pi$  с шагом  $\pi/10$ , можно вычислить при определенном значении радиуса значения  $x$  и  $y$ .

Для решения задачи сначала надо заполнить ряды данных для формирования окружности, используя область ячеек A1:C20. В ячейки A1:A20 заносятся значения угла  $f$  от 0 до 5,966 с шагом 0,314. В ячейки B1 и C1 заносятся формулы для вычисления значений  $x$  и  $y$  – координат точки. Далее блок ячеек B1:C1 копируется на диапазон B2:C20. Фрагмент расчетной таблицы имеет вид:

	A	B	C
1	0	=COS(A1)	=SIN(A1)
2	=A1+0,314	=COS(A2)	=SIN(A2)
3	=A2+0,314	=COS(A3)	=SIN(A3)
4	=A3+0,314	=COS(A4)	=SIN(A4)

Затем по диапазону ячеек B1:C20 строится точечный график для окружности.

Для задания координат движущейся точки используются ячейки B22 и C22. Изменение значений  $x$  и  $y$  осуществляется с помощью VBA. Задача программы на VBA – по очереди брать пары значений из диапазона B1:C20 и записывать их в ячейки B22 и C22. Если на рабочем листе отсутствует панель элементов управления, ее можно включить командой меню **Вид, Панели инструментов, Элементы управления**. Используя панель элементов управления, нужно создать элемент управления «кнопка» и, дважды щелкнув по полученной кнопке, перейти в окно VBA. Свойству **Caption** присваивается значение «Начать движение» – это будет надпись на кнопке. Процедура должна выглядеть следующим образом:

```
Private Sub CommandButton1_Click()
    For i = 1 To 20 Step 1
        Cells(22, 2) = Cells(i, 2)
        Cells(22, 3) = Cells(i, 3)
    DoEvents
    Next i
End Sub
```

Таким образом, в процедуре формируется цикл для переменной  $i$  и идет последовательная пересылка вычисленных пар значений координат точки

в ячейки B22 и C22. Команда DoEvents дает возможность ЭТ произвести перерасчет листа.

Далее необходимо добавить к построенному графику окружности точку, координаты которой заданы в ячейках B22 и C22. Изменив размеры и форму маркера, можно сделать движущуюся точку более заметной. Анимация запускается щелчком по созданной кнопке «Начать движение».

Таблица 9

### Варианты индивидуальных заданий

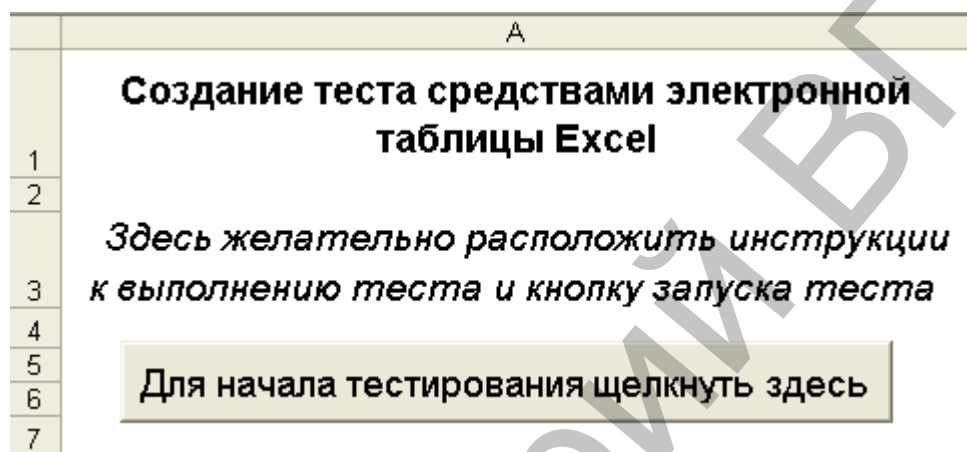
№	Уравнение траектории
1	Циклоида: $x = a(t - \sin t)$ ; $y = a(1 - \cos t)$ , $-\infty < t < +\infty$ , где $a$ – радиус катящегося круга, $t$ – угол, на который поворачивается круг при своем качении по прямой
2	Кардиоида: $x = a \cos t(1 + \cos t)$ , $y = a \sin t(1 + \cos t)$ ; $0 \leq t \leq 2\pi$
3	Трактриса: $x = a(\ln \operatorname{tg} \frac{t}{2} + \cos t)$ , $y = a \sin t$
4	Декартов лист: $x = \frac{3at}{t^3 + 1}$ , $y = \frac{3at^2}{t^3 + 1}$
5	Улитка Паскаля: $x = a \cos^2 t + l \cos t$ , $y = a \cos t \sin t$ , $0 \leq t \leq 2\pi$
6	Эпициклоида: $x = (a + b) \cos \varphi - a \cos((a + b)\varphi / a)$ $y = (a + b) \sin \varphi - a \sin((a + b)\varphi / a)$ ; где угол $-\infty \leq \varphi \leq +\infty$ , а отношение $m = \frac{b}{a}$ определяет вид кривой
7	Эвольвента окружности: $x = r(\cos t + t \sin t)$ , $y = r(\sin t - t \cos t)$ $-\infty < t < +\infty$
8	Лемниската Бернулли: $x = a\sqrt{2} \frac{t + t^3}{1 + t^4}$ ; $y = a\sqrt{2} \frac{t - t^3}{1 + t^4}$ , $-\infty < t < +\infty$
9	Гипоциклоида: $x = (b - a) \cos \varphi + a \cos((b - a)\varphi / a)$ $y = (b - a) \sin \varphi - a \sin((b - a)\varphi / a)$ ; где угол $-\infty \leq \varphi \leq +\infty$ , а отношение $m = \frac{b}{a}$ определяет вид кривой
10	Строфоида: $x = \frac{2at^2}{1 + t^2}$ , $y = \frac{at(t^2 - 1)}{1 + t^2}$

## Создание тестирующей программы

Входящие в состав электронной таблицы арифметические, логические функции и функции обработки текста позволяют достаточно просто создавать тестирующие программы.

**Задание 1.** Создать тестирующую программу.

Для выполнения задания нужно поместить на лист рабочей книги Excel инструкции к выполнению теста и кнопку запуска *CommandButton1*, например:



Последовательность действий при подготовке теста описана далее.

1. Перейти в редактор VBA (команда *Сервис, Макрос, Редактор Visual Basic*). Создать новую форму (команда *Вставка, UserForm*).

The screenshot shows the Visual Basic UserForm1 editor with the following content:

- Form Title:** UserForm1
- Fields:**
  - Фамилия:
  - Имя:
  - Класс:
- Example Test:** *Пример теста, построенного средствами электронной таблицы Excel*
- 1. Альтернативный ответ:** Является ли атом мельчайшей неделимой частицей материи?
  - Является
  - Не является
- 2. Множественный выбор:** Какие из перечисленных законов и правил относятся к области физики?
  - Закон Паули
  - Закон падающего бутерброда
  - Правила дорожного движения
  - Закон сохранения импульса
- 3. Множественный выбор:** Самолет, выполняя фигуры высшего пилотажа, описывает "мертвую петлю". Каково направление вектора скорости в точке А траектории?
  - 1
  - 2
  - 3
  - 4
- 4. Соответствие:** В каждое поле ответа поставьте цифру, соответствующую правильному ответу.
  - Напряжение измеряется
  - Сила тока измеряется
  - Сопротивление измеряется
  - 1. Амперметром
  - 2. Линейкой и циркулем
  - 3. Омметром
  - 4. Вольтметром
- 5. Совпадение ответа с эталоном:**
  - Сколько простых чисел находится в диапазоне от 0 до 9?
  - Как зовут человека, с чьим именем связано учения об эволюции в биологических системах?
- 6. Дополнение:** Вам дано начало предложения. Закончите его.
  - Единицей измерения информации является.
- Button:** Показать результаты тестирования

Разместить на форме метки, текстовые поля для ввода фамилии, имени и класса, фреймовую структуру.

Для задания метки или заголовка фрейма использовать свойство *Caption*. Добавить к созданной форме метки, задающие вопросы теста; рисунок; текстовые поля для ввода свободных ответов; элементы управления для организации выбора из нескольких предложенных вариантов ответов. Добавить кнопку *CommandButton2* для вывода результатов тестирования в ячейки электронной таблицы. Окончательный вид формы:

2. Далее необходимо подготовить ячейки ЭТ, в которые будут вводиться данные из формы. Ответы ученика (столбец F) будут сравниваться с правильными ответами (столбец G). Логические формулы для сравнения ответа ученика с правильными ответами вводятся в столбец H. Если ответ верен, то есть совпадает с эталоном, то в ячейке выставляется 1, иначе – 0. По количеству правильных ответов можно формировать оценку за тест. Когда тест будет полностью отлажен, целесообразно эти ячейки скрыть. Заполнить таблицу следующим образом:

	C	D	E	F	G	H
	<b>Данные ученика</b>		<b>Номер и тип теста</b>	<b>Ответ ученика</b>	<b>Правильный ответ</b>	<b>Проверка ответа</b>
1	Фамилия		1. Альтернативных ответов			
2	Имя		является/не является		1	=ЕСЛИ(F3=G3;1;0)
3	Класс		2. Множественного выбора			
4			Закон Паули		1	=ЕСЛИ(F5=G5;1;0)
5			Закон бутерброда		0	=ЕСЛИ(F6=G6;1;0)
6			Дорожного движения		0	=ЕСЛИ(F7=G7;1;0)
7			Сохранения импульса		1	=ЕСЛИ(F8=G8;1;0)
8			3. Множественного выбора			
9			1/2/3/4		1	=ЕСЛИ(F10=G10;1;0)
10			4. Соответствия			
11			напряжение		4	=ЕСЛИ(F12=G12;1;0)
12			сила тока		1	=ЕСЛИ(F13=G13;1;0)
13			сопротивление		3	=ЕСЛИ(F14=G14;1;0)
14			5. Свободного изложения			
15			Простые числа		4	=ЕСЛИ(F16=G16;1;0)
16			Эволюция		Дарвин	=ЕСЛИ(F17=G17;1;0)
17			6. Дополнения			
18					бит	=ЕСЛИ(F19=G19;1;0)
19						

Для формирования оценки нужно ввести:

- в ячейку C21 текст: «Количество правильных ответов на вопросы теста»;
- в ячейку C22 текст: «Процент правильных ответов к общему количеству ответов»;
- в ячейку C23 текст: «Оценка за тест»;
- в ячейку H21 формулу подсчета правильных ответов: =СУММ(H3:H19);
- в ячейку H22 формулу для получения процента правильных ответов: =H21/12;

- в ячейку H23 формулу:  
=ЕСЛИ(H22<30%;2;ЕСЛИ(H22<50%;3;ЕСЛИ(H22<75%;4;5)))

3. После формирования структуры таблицы необходимо написать процедуру для кнопки вызова формы *CommandButton1*. В этой процедуре надо очистить ячейки таблицы, предназначенные для ввода ответов ученика и показать на экране форму с тестом:

```
Private Sub CommandButton1_Click()
    Range("d2:d4").Clear
    Range("f3:f19").Clear
    UserForm1.Show
End Sub
```

4. Далее приведены процедуры обработки событий, которые определяют введенные в форму данные и заносят их в ячейки рабочего листа. Например:

- для текстового поля, в которое вводится фамилия учащегося:

```
Private Sub TextBox1_Change()
    Cells(2, 4) = TextBox1.Text
End Sub
```

- для переключателей *OptionButton* (альтернативный ответ):

```
Private Sub OptionButton1_Click()
    If OptionButton1.Value = True Then
        Cells(3, 6) = 0
    End If
End Sub
Private Sub OptionButton2_Click()
    If OptionButton2.Value = True Then
        Cells(3, 6) = 1
    End If
End Sub
```

- для переключателя *CheckBox* (выбор из нескольких вариантов ответов):

```
Private Sub CheckBox1_Click()
    If CheckBox1.Value = True Then
        Cells(5, 6) = 1
    Else: Cells(5, 6) = 0
    End If
End Sub
```

Аналогично в блок ответов ученика вводятся данные из остальных вопросов теста.

5. Далее необходимо написать процедуру для кнопки вывода результатов тестирования *CommandButton2*. В этой процедуре надо скрыть форму с тестом, а результаты тестирования занести в ячейки, доступные для просмотра:

```
Private Sub CommandButton2_Click()
    UserForm1.Hide
    Cells(7, 1) = "Результаты тестирования:"
```

```

Cells(8, 1) = "Уважаемый " + Cells(2, 4) + " " +
Cells(3, 4)
Cells(9, 1) = "Из 12 заданий теста выполнено правильно"
Cells(10, 1) = Cells(21, 8)
Cells(11, 1) = "Оценка за тест"
Cells(12, 1) = Cells(23, 8)
End Sub

```

В результате выполнения этой процедуры после выполнения теста ученик увидит на экране следующую информацию:

7	<b>Результаты тестирования:</b>
8	<b>Уважаемый Иванов Вася</b>
9	<b>Из 12 заданий теста выполнено правильно</b>
10	<b>9</b>
11	<b>Оценка за тест</b>
12	<b>4</b>

**Задание 2.** Самостоятельно создать тестирующую программу по одному из разделов информатики.

### **Решение систем линейных алгебраических уравнений**

Методы решения линейных алгебраических систем принято разделять на прямые и итерационные.

*Прямые методы* используют конечные соотношения (формулы) для вычисления неизвестных, и, следовательно, позволяют получить решение системы за конечное число арифметических операций. Из-за того, что решение выражается в виде точных формул через коэффициенты системы, эти методы называют *точными*. Однако точное решение может быть получено лишь при выполнении вычислений с бесконечным числом разрядов (при вычислениях с использованием ЭВМ вычисления производятся с ограничением числа знаков).

Эти методы сравнительно просты и наиболее универсальны, т.е. пригодны для решения довольно широкого класса систем линейных алгебраических уравнений. К прямым методам относятся: метод решения линейной системы с использованием обратной матрицы (матричный метод), метод Крамера, метод подстановки, методы последовательного исключения неизвестных (метод Гаусса и его модификации: метод главного элемента, метод квадратного корня, метод отражений и др.), метод ортогонализации.

Прямые методы имеют свои недостатки:

- как правило, они требуют хранения в памяти ЭВМ сразу всей матрицы;
- прямые методы не учитывают структуру матрицы – при большом количестве нулевых элементов в разреженных матрицах над этими элементами все равно выполняются арифметические действия;
- в процессе решения происходит накопление погрешностей, поскольку вычислений на любом этапе используют результаты предыдущих операций.

По этим причинам при вычислениях с использованием ЭВМ прямыми методами пользуются при обработке сравнительно небольших систем с плотно заполненной матрицей и не близким к нулю определителем.

*Итерационные методы* являются приближенными. Решение системы в данном случае представляется как предел последовательных приближений, вычисляемых по единому алгоритму. При использовании этих методов задается некоторое приближенное решение – *начальное приближение*, после чего проводится цикл вычислений, называемый *итерацией*, в результате которого находят новое приближение. Итерации проводятся до тех пор, пока решение не будет получено с требуемой точностью.

По сравнению с простыми методами алгоритмы решений систем линейных алгебраических уравнений итерационными методами обычно более сложные, а объем вычислений трудно определить заранее. Однако использование итерационных методов в сочетании с вычислениями за компьютером иногда становится более предпочтительным, так как:

- при итерационных вычислениях не требуется хранить в памяти машины всю матрицу системы – бывает достаточным хранить лишь несколько векторов с  $n$  компонентами, иногда элементы матрицы можно вовсе не хранить, а вычислять их по мере необходимости;
- поскольку точность вычислений в каждой итерации определяется лишь результатами предыдущей итерации и практически не зависит от ранее выполненных вычислений, при использовании итерационных методов погрешности окончательных результатов не накапливаются.

Итерационные методы используются при решении систем с довольно большим количеством уравнений. Поскольку сходимость итераций может быть очень медленной, одной из задач численных методов является отыскание эффективных путей ускорения сходимости итерационных процессов.

К итерационным методам относятся: метод простой итерации, метод Зейделя, метод релаксаций, градиентные методы и их модификации.

**Матричный метод.** Пусть дана система  $n$  уравнений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases} \quad (1)$$

Рассмотрим матрицу, составленную из коэффициентов при неизвестных:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & a_{nn} \end{pmatrix}$$



Свободные члены и неизвестные можно записать в виде векторов:

$$B = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix}, X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

Используя правило умножения матриц, систему уравнений (1) можно записать так:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & a_{nn} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix} \text{ или } AX = B$$

Это равенство называется *простейшим матричным уравнением*. Такое уравнение решается следующим образом. Пусть матрица  $A$  – невырожденная<sup>1</sup>, тогда существует обратная<sup>2</sup> матрица  $A^{-1}$ . Умножив на нее обе части матричного уравнения, имеем:

$$A^{-1}(AX) = A^{-1}B$$

Используя сочетательный закон умножения, перепишем это равенство в виде:  $(A^{-1}A)X = A^{-1}B$ . Поскольку  $A^{-1} \cdot A = E$  и  $E \cdot X = X$ , находим

$$X = A^{-1}B.$$

Таким образом, чтобы решить матричное уравнение, нужно:

1. Найти обратную матрицу  $A^{-1}$ .
2. Вычислить произведение обратной матрицы  $A^{-1}$  на вектор-столбец свободных членов  $B$ , т.е.  $A^{-1} \cdot B$ .
3. Пользуясь определением равных матриц, записать ответ.

Среда MS Excel располагает следующими математическими функциями:

- 1) **МОБР**(массив) – возвращает обратную матрицу для матрицы, хранящейся в массиве; функция МОБР производит вычисления с точностью до 16 значащих цифр, что может привести к небольшим численным ошибкам округления.

<sup>1</sup> **Невырожденной** (неособенной) называется квадратная матрица  $A$ , определитель которой не равен нулю ( $\det A \neq 0$ ).

<sup>2</sup> **Обратной** по отношению к данной называется матрица  $A^{-1}$ , которая, будучи умноженной как справа, так и слева на указанную матрицу, дает единичную матрицу, т.е.  $A \cdot A^{-1} = A^{-1} \cdot A = E$ . Единичная матрица представляет собой квадратный массив, у которого диагональные элементы равны 1, а все остальные элементы равны 0.

- 2) **МУМНОЖ**(массив\_1; массив\_2) – возвращает произведение матриц (матрицы хранятся в массивах). Результатом является массив с таким же числом строк, как массив\_1 и с таким же числом столбцов, как массив\_2. К аргументам этой функции предъявляются следующие требования:
- Количество столбцов аргумента массив\_1 должно быть таким же, как количество строк аргумента массив\_2, и оба массива должны содержать только числа.
  - Массив\_1 и массив\_2 могут быть заданы как интервалы, массивы констант или ссылки.
  - Если хотя бы одна ячейка в аргументах пуста или содержит текст, или если число столбцов в аргументе массив\_1 отличается от числа строк в аргументе массив\_2, то функция МУМНОЖ возвращает значение ошибки #ЗНАЧ!.
- 3) **МОПРЕД**(массив) – возвращает определитель матрицы (матрица хранится в массиве). Под массивом понимается числовой массив с равным количеством строк и столбцов.
- Массив может быть задан как интервал ячеек, например, A1:C3 или как массив констант, например {1;2;3;4;5;6;7;8;9} или как имя, имеющее интервал или массив.
  - Если какая-либо ячейка в массиве пуста или содержит текст, то функция МОПРЕД возвращает значение ошибки #ЗНАЧ!
  - МОПРЕД также возвращает значение ошибки #ЗНАЧ!, если массив имеет неравное количество строк и столбцов.

МОПРЕД производит вычисления с точностью примерно 16 значащих цифр, что может в некоторых случаях приводить к небольшим численным ошибкам. Например, определитель сингулярной матрицы отличается от нуля на  $1E-16$ .

**Пример 1.** Решить систему линейных алгебраических уравнений

$$\begin{cases} 12x_1 - 8x_2 - 4x_3 + 6x_4 = -6 \\ -5x_1 - 4x_2 + 16x_3 - 8x_4 = 7 \\ 15x_1 - 5x_2 - 7x_3 + 20x_4 = 56 \\ 2x_1 + 6x_2 + 2x_3 - x_4 = -3 \end{cases}$$

**Решение.** В ячейки B1:E4 введем матрицу, составленную из коэффициентов при неизвестных, в ячейки H1:H4 введем вектор-столбец свободных членов. Для вычисления определителя матрицы  $A$  в ячейку B6 введем следующую формулу: =МОПРЕД(B1:E6).

Определитель матрицы оказался равным 18220, матрица  $A$  является квадратной, следовательно, матрица является невырожденной и для отыскания решений системы применим матричный метод.

Для построения обратной матрицы выделяем диапазон ячеек B8:B11, затем вводим в ячейку B8 следующую формулу: =МОБР(B1:E6), закончив ввод в строке формул нажатием комбинации клавиш **Ctrl + Shift + Enter**, поскольку введенная формула представляет собой *формулу массива*<sup>3</sup>.

В ячейки A19:E22 введем формулы для проверки правильности найденного решения. Лист с формулами в Excel представлен на рисунке 1. На рисунке 2 приведены результаты этого решения.

	A	B	C	D	E	F	G	H
1	A =	12	-8	-4	6	B =	-6	
2		-5	-4	16	-8		7	
3		15	-5	-7	20		56	
4		2	6	2	-1		-3	
5								
6	detA =	=МОПРЕД(B1:E4)						
7								
8	A <sub>обр</sub> =	=МОБР(B1:E4)		=МОБР(B1:E4)	=МОБР(B1:E4)	=МОБР(B1:E4)		
9		=МОБР(B1:E4)		=МОБР(B1:E4)	=МОБР(B1:E4)	=МОБР(B1:E4)		
10		=МОБР(B1:E4)		=МОБР(B1:E4)	=МОБР(B1:E4)	=МОБР(B1:E4)		
11		=МОБР(B1:E4)		=МОБР(B1:E4)	=МОБР(B1:E4)	=МОБР(B1:E4)		
12								
13	X =	=МУМНОЖ(B8:E11;H1:H4)						
14		=МУМНОЖ(B8:E11;H1:H4)						
15		=МУМНОЖ(B8:E11;H1:H4)						
16		=МУМНОЖ(B8:E11;H1:H4)						
17								
18	<b>Проверка</b>							
19	=B1*\$B\$13	=C1*\$B\$14	=D1*\$B\$15	=E1*\$B\$16	=СУММ(A19:D19)			
20	=B2*\$B\$13	=C2*\$B\$14	=D2*\$B\$15	=E2*\$B\$16	=СУММ(A20:D20)			
21	=B3*\$B\$13	=C3*\$B\$14	=D3*\$B\$15	=E3*\$B\$16	=СУММ(A21:D21)			
22	=B4*\$B\$13	=C4*\$B\$14	=D4*\$B\$15	=E4*\$B\$16	=СУММ(A22:D22)			

Рис. 1. Лист с формулами для решения СЛАУ методом нахождения неизвестных через обратную матрицу.

<sup>3</sup> **Формула массива** выполняет несколько вычислений и возвращает либо одно, либо несколько значений. Формулы массива обычно обрабатывают два и более набора аргументов, называемых аргументами массива, при этом все аргументы массива должны иметь одинаковое количество строк и столбцов. Чтобы вернуть несколько значений, формулу необходимо ввести в несколько ячеек. Чтобы ввести формулу массива, следует нажать клавиши Ctrl + Shift + Enter. MS Excel заключает формулы массива в фигурные скобки { }.

	A	B	C	D	E	F	G	H
1	A =	12	-8	-4	6		B =	-6
2		-5	-4	16	-8			7
3		15	-5	-7	20			56
4		2	6	2	-1			-3
5								
6	detA =	18220						
7								
8	A <sub>обр</sub> =	0,094	2E-04	-0,02	0,107			
9		-0,04	-0,02	0,009	0,111			
10		-0,02	0,067	0,036	0,042			
11		-0,09	0,018	0,082	-0,04			
12								
13	X =	-2,16						
14		0,277						
15		2,517						
16		5,372						
17								
18	Проверка							
19	-25,95	-2,22	-10,1	32,23	-6			
20	10,81	-1,11	40,27	-43	7			
21	-32,43	-1,39	-17,6	107,4	56			
22	-4,324	1,662	5,034	-5,37	-3			

Рис. 2. Решение системы линейных алгебраических уравнений.

Решение можно существенно сократить, если не выводить на экран обратную матрицу, а включить её вычисление в формулу для произведения матриц. Этот вариант проиллюстрирован на рисунках 3 и 4.

	A	B	C	D	E	F	G	H
1	A =	12		-8	-4	6	B =	-6
2		-5		-4	16	-8		7
3		15		-5	-7	20		56
4		2		6	2	-1		-3
5								
6	detA =	=МОПРЕД(B1:E4)						
7								
8	X =	=МУМНОЖ(МОБР(B1:E4);H1:H4)						
9		=МУМНОЖ(МОБР(B1:E4);H1:H4)						
10		=МУМНОЖ(МОБР(B1:E4);H1:H4)						
11		=МУМНОЖ(МОБР(B1:E4);H1:H4)						
12								
13	Проверка							
14	=B1*\$B\$8	=C1*\$B\$9	=D1*\$B\$10	=E1*\$B\$11	=СУММ(A14:D14)			
15	=B2*\$B\$8	=C2*\$B\$9	=D2*\$B\$10	=E2*\$B\$11	=СУММ(A15:D15)			
16	=B3*\$B\$8	=C3*\$B\$9	=D3*\$B\$10	=E3*\$B\$11	=СУММ(A16:D16)			
17	=B4*\$B\$8	=C4*\$B\$9	=D4*\$B\$10	=E4*\$B\$11	=СУММ(A17:D17)			

Рис. 3. Лист с формулами для решения системы линейных алгебраических уравнений матричным методом.

	A	B	C	D	E	F	G	H
1	A =	12	-8	-4	6		B =	-6
2		-5	-4	16	-8			7
3		15	-5	-7	20			56
4		2	6	2	-1			-3
5								
6	detA =	18220						
7								
8	X =	-2,16						
9		0,277						
10		2,517						
11		5,372						
12								
13	Проверка							
14	-25,95	-2,22	-10,1	32,23	-6			
15	10,81	-1,11	40,27	-43	7			
16	-32,43	-1,39	-17,6	107,4	56			
17	-4,324	1,662	5,034	-5,37	-3			

Рис. 4. Решение системы линейных алгебраических уравнений.

**Метод Крамера.** Удобным аппаратом исследования систем линейных алгебраических уравнений являются методы, связанные с вычислением определителей. Одним из них является метод Крамера. Для использования этого метода необходимым является выполнение следующих условий (которые следуют из теоремы Крамера<sup>4</sup>):

- число уравнений должно быть равно числу неизвестных;
- главный определитель системы должен быть отличен от нуля.

Отыскание единственного решения системы из  $n$  уравнений с  $n$  неизвестными методом Крамера сводится к вычислению  $n+1$  определителя:

1. Из коэффициентов при неизвестных составим матрицу  $A$  (ее называют *основной матрицей системы*), из свободных членов – вектор-столбец  $B$ , т.е.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & a_{nn} \end{pmatrix}, B = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix}.$$

*Расширенной матрицей системы* называют матрицу, полученную из матрицы  $A$  добавлением столбца свободных членов:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & a_{nn} & b_n \end{pmatrix}$$

<sup>4</sup> **Теорема Крамера:** система  $n$  уравнений с  $n$  неизвестными, определитель которой отличен от нуля, всегда имеет решение, и притом единственное.

2. Обозначим буквой  $\Delta$  определитель матрицы  $A$ , его называют главным определителем системы:

$$\Delta = \det A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & a_{nn} \end{vmatrix}.$$

Вычислим определитель и убедимся, что он отличен от нуля. Если основная матрица системы является квадратной и главный определитель системы отличен от нуля, то ее ранг равен рангу расширенной матрицы. По теореме Кронекера-Капелли (см., например, [1], с. 18) это означает, что система совместна и, кроме того, имеет единственное решение.

3. В главном определителе заменим поочередно столбцы коэффициентов при  $x_1, x_2, \dots, x_n$  на столбец свободных членов, получим  $n$  определителей (для  $n$  неизвестных):

$$\Delta_{x_1} = \begin{vmatrix} b_1 & a_{12} & \dots & a_{1n} \\ b_2 & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ b_n & a_{n2} & a_{n3} & a_{nn} \end{vmatrix}, \Delta_{x_2} = \begin{vmatrix} a_{11} & b_1 & \dots & a_{1n} \\ a_{21} & b_2 & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & b_n & a_{n3} & a_{nn} \end{vmatrix}, \dots,$$

$$\Delta_{x_n} = \begin{vmatrix} a_{11} & a_{12} & \dots & b_1 \\ a_{21} & a_{22} & \dots & b_2 \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & b_n \end{vmatrix}.$$

4. Значение каждого из  $n$  неизвестных отыскивается по формулам Крамера:

$$x_1 = \frac{\Delta_{x_1}}{\Delta}, x_2 = \frac{\Delta_{x_2}}{\Delta}, \dots, x_n = \frac{\Delta_{x_n}}{\Delta}.$$

В случае, когда главный определитель равен нулю, особого внимания заслуживают два случая:

- 1)  $\Delta = 0$  и каждый определитель  $\Delta_{x_i} = 0, i = 1, 2, \dots, n$ . Это имеет место только тогда, когда коэффициенты при неизвестных  $x_i$  пропорциональны, т.е. каждое уравнение системы получается из первого уравнения умножением обеих его частей на некоторую константу  $k$ . Очевидно, что при этом система имеет бесконечное множество решений.
- 2)  $\Delta = 0$  и хотя бы один определитель  $\Delta_{x_i} \neq 0, i = 1, 2, \dots, n$ . Это имеет место только тогда, когда коэффициенты при всех неизвестных  $x_i$  пропорциональны. При этом получается система из противоречивых уравнений, которая не имеет решений.

Реализация метода Крамера в среде Excel упрощается в связи с возможностью использования функции МОПРЕД.

**Пример 2.** Решить систему линейных алгебраических уравнений из примера 1 методом Крамера.

$$\begin{cases} 12x_1 - 8x_2 - 4x_3 + 6x_4 = -6 \\ -5x_1 - 4x_2 + 16x_3 - 8x_4 = 7 \\ 15x_1 - 5x_2 - 7x_3 + 20x_4 = 56 \\ 2x_1 + 6x_2 + 2x_3 - x_4 = -3 \end{cases}$$

**Решение.** В ячейки В1:Е4 введем матрицу, составленную из коэффициентов при неизвестных, в ячейки Н1:Н4 – вектор-столбец свободных членов. В ячейке F4 вычислим главный определитель и убедимся, что он отличен от нуля.

Поскольку количество неизвестных в системе равно количеству уравнений в ней, а главный определитель не равен нулю, метод Крамера применим для решения этой системы линейных алгебраических уравнений.

В ячейку С6 введем ссылку на ячейку С2, выделим ее и скопируем, используя маркер заполнения<sup>5</sup>, на диапазон С6:Е9. В ячейки В6:В9 введем ссылки на значения столбца свободных членов. Таким образом, сформирована матрица для вычисления определителя  $D_1 = \Delta_{x_1}$ . Остальные три матрицы формируются аналогичным образом.

В ячейках F9, F14, F19, F24 используя функцию МОПРЕД вычисляем определители  $D_1, D_2, D_3, D_4$  соответствующих матриц. В ячейки J1:J4 вводим формулы Крамера.

Проверку найденных решений можно выполнить умножением основной матрицы системы на вектор-столбец X, в результате получаем значения, совпадающие со значениями вектора-столбца свободных членов. Соответствующие формулы приведены в ячейках Н7:Н10. Все расчетные формулы приведены на рисунке 5.

Очевидно, что найденные методом Крамера решения совпадают с решениями, найденными матричным методом. На рисунке 6 приведены результаты вычислений.

<sup>5</sup> **Маркер заполнения** – черный квадрат в правом нижнем углу активной ячейки или выделения. Когда указатель находится на маркере заполнения, он изменяет вид со стрелки на перекрестье. Для заполнения ячеек данными, основанными на текущем выделении, необходимо перетащить маркер заполнения вниз или вправо.

	A	B	C	D	E	F	G	H	I	J
1	A =	12	-8	-4	6		B =	-6	X =	=F9/\$F\$4
2		-5	-4	16	-8			7		=F14/\$F\$4
3		15	-5	-7	20			56		=F19/\$F\$4
4		2	6	2	-1	=МОПРЕД(B1:E4)		-3		=F24/\$F\$4
5										
6	D <sub>1</sub> =	=H1	=C1	=D1	=E1					
7		=H2	=C2	=D2	=E2					
8		=H3	=C3	=D3	=E3					
9		=H4	=C4	=D4	=E4	=МОПРЕД(B6:E9)				
10										
11	D <sub>2</sub> =	=B1	=H1	=D1	=E1					
12		=B2	=H2	=D2	=E2					
13		=B3	=H3	=D3	=E3					
14		=B4	=H4	=D4	=E4	=МОПРЕД(B11:E14)				
15										
16	D <sub>3</sub> =	=B1	=C1	=H1	=E1					
17		=B2	=C2	=H2	=E2					
18		=B3	=C3	=H3	=E3					
19		=B4	=C4	=H4	=E4	=МОПРЕД(B16:E19)				
20										
21	D <sub>4</sub> =	=B1	=C1	=D1	=H1					
22		=B2	=C2	=D2	=H2					
23		=B3	=C3	=D3	=H3					
24		=B4	=C4	=D4	=H4	=МОПРЕД(B21:E24)				

Проверка

=МУМНОЖ(B1:E4;J1:J4)
=МУМНОЖ(B1:E4;J1:J4)
=МУМНОЖ(B1:E4;J1:J4)
=МУМНОЖ(B1:E4;J1:J4)

Рис. 5. Расчетные формулы для решения СЛАУ методом Крамера.

	A	B	C	D	E	F	G	H	I	J
1	A =	12	-8	-4	6		B =	-6	X =	-2,16
2		-5	-4	16	-8			7		0,277
3		15	-5	-7	20			56		2,517
4		2	6	2	-1	18220		-3		5,372
5										
6	D <sub>1</sub> =	-6	-8	-4	6					
7		7	-4	16	-8					
8		56	-5	-7	20					
9		-3	6	2	-1	-39396				
10										
11	D <sub>2</sub> =	12	-6	-4	6					
12		-5	7	16	-8					
13		15	56	-7	20					
14		2	-3	2	-1	5048				
15										
16	D <sub>3</sub> =	12	-8	-6	6					
17		-5	-4	7	-8					
18		15	-5	56	20					
19		2	6	-3	-1	45860				
20										
21	D <sub>4</sub> =	12	-8	-4	-6					
22		-5	-4	16	7					
23		15	-5	-7	56					
24		2	6	2	-3	97876				

Рис. 6. Решение системы линейных алгебраических уравнений.



**Метод Гаусса.** Данный метод включает в себя два этапа: прямой ход и обратный ход.

*Прямой ход метода Гаусса* основан на приведении матрицы к треугольному виду путем последовательного исключения неизвестных из уравнений системы. Такие преобразования не изменяют множество решений системы. Сначала с помощью первого уравнения исключается  $x_1$  из всех последующих уравнений системы. Затем с помощью второго уравнения исключается  $x_2$  из третьего и всех последующих уравнений. Этот процесс продолжается до тех пор, пока в левой части последнего ( $n$ -го уравнения) не останется лишь один член с неизвестным  $x_n$ .

$$\begin{cases} x_1 + \alpha_{12}x_2 + \dots + \alpha_{1n}x_n = \beta_1 \\ x_2 + \dots + \alpha_{2n}x_n = \beta_2 \\ \dots \\ x_n = \beta_n \end{cases} \quad (2)$$

Если в процессе преобразований возникает уравнение, в котором все коэффициенты, включая свободный член, нулевые, то такое уравнение можно исключить из системы без изменения множества решений, т.к. ему удовлетворяют любые действительные значения  $x_1, x_2, \dots, x_n$ . Если же процесс преобразований приведет к уравнению, в котором все коэффициенты при неизвестных равны нулю, а свободный член нулю не равен, то система несовместна, так как никакие действительные значения  $x_1, x_2, \dots, x_n$  такому уравнению не удовлетворяют.

Обратный ход метода Гаусса заключается в последовательном вычислении искоемых неизвестных: решая последнее уравнение, находим единственное неизвестное  $x_n$ . Подставляя найденное значение в предпоследнее уравнение, находим значение  $x_{n-1}$ . Последним будет найдено значение переменной  $x_1$  из первого уравнения.

**Пример 3.** Решить методом Гаусса систему линейных алгебраических уравнений:

$$\begin{cases} 4x_1 - 13x_2 + 6x_3 = -24 \\ -2x_1 + 4x_2 - 4x_3 = 18 \\ -8x_1 + 7x_2 - 20x_3 = 104 \end{cases}$$

**Решение.** Составим расширенную матрицу системы:

$$\begin{pmatrix} 4 & -13 & 6 & -24 \\ -2 & 4 & -4 & 18 \\ -8 & 7 & -20 & 104 \end{pmatrix}$$

Введем эту матрицу в ячейки A1:D3. Выполним прямой ход метода Гаусса. Разделим первое уравнение на 4, чтобы получить при  $x_1$  коэффициент,

равный 1. Для этого в ячейку A5 введем формулу  $=A1/\$A\$1$  и скопируем ее в ячейки B5:D5.

Чтобы исключить  $x_1$  из второго уравнения, в ячейку A6 введем формулу  $=A2/\$A\$2-A5$  и скопируем ее в ячейки B6:D6.

Чтобы исключить  $x_1$  из третьего уравнения, в ячейку A7 введем формулу  $=A3/\$A\$3-A5$  и скопируем ее в ячейки B7:D7. После таких преобразований получим следующую матрицу:

$$\begin{pmatrix} 1 & -3,25 & 1,5 & -6 \\ 0 & 1,25 & 0,5 & -3 \\ 0 & 2,375 & 1 & -7 \end{pmatrix}$$

Поскольку значения первой строки и первого столбца этой матрицы изменяться не будут, перенесем их значения в ячейки A9:D9 и A10:A11. Для этого в ячейку A9 введем формулу  $=A5$  и скопируем ее на диапазоны ячеек B9:D9 и A10:A11. Для преобразований оставшейся части матрицы в ячейку B10 введем формулу  $=B6/\$B\$6$  и скопируем ее в ячейки B11:D11. Таким образом, мы получили коэффициент 1 при  $x_2$  во втором уравнении. Для преобразования третьего уравнения в ячейку B11 введем формулу  $=B7/\$B\$7-B10$  и скопируем ее в ячейки C11:D11. После преобразований матрица примет следующие значения:

$$\begin{pmatrix} 1 & -3,25 & 1,5 & -6 \\ 0 & 1 & 0,4 & -2,4 \\ 0 & 0 & 0,02105 & -0,5474 \end{pmatrix}$$

На последнем этапе преобразований получим коэффициент 1 при  $x_3$  в третьем уравнении. Для этого оставим без изменения первую и вторую строки и первый и второй столбцы матрицы, а элементы третьей строки разделим на значение ячейки C11, чтобы получить единицу при  $x_3$ . Итак, в ячейку A13 введем формулу с относительной ссылкой на ячейку A9:  $=A9$  и скопируем ее в ячейки B9:D9, A14:D14, A15:B15. В ячейку C15 введем формулу  $=C11/\$C\$11$  и скопируем ее в ячейку D15. Получим матрицу, приведённую к треугольному виду:

$$\begin{pmatrix} 1 & -3,25 & 1,5 & -6 \\ 0 & 1 & 0,4 & -2,4 \\ 0 & 0 & 1 & -26 \end{pmatrix}$$

Обратный ход метода Гаусса заключается в нахождении значений неизвестных. Значение  $x_3$  найдем из последней строчки и вычислим в ячейке B19:  $=D15$ . Значение переменной  $x_2$  вычислим из второй строки в ячейке B18:  $=D14-C14*B19$ . И, наконец, значение переменной  $x_1$  вычислим из первой строки в ячейке B17:  $=D13-C13*B19-B13*B18$ . Решением данной

системы уравнений являются  $x_1 = 59$ ,  $x_2 = 8$  и  $x_3 = -26$ . В ячейках A22:A24 выполним проверку найденных решений.

Для исключения случайных ошибок предусматривается текущий контроль правильности вычислений. Для этого, после того, как в электронную таблицу внесена расширенная матрица системы, найдём контрольные суммы – суммы коэффициентов и свободных членов по строкам и внесём их в столбец E. Выполняя дальнейшие преобразования уравнений системы, над этими суммами будем производить те же операции, что и над свободными членами. После выполнения каждого преобразования вычислим суммы по столбцам (в нашем примере значения этих сумм будем размещать в строках 4, 8, 12 и 16). В столбце F будем размещать строчные суммы результатов (суммы сумм). Ясно, что при отсутствии случайных вычислительных ошибок пары чисел в ячейках E4 и F4, E8 и F8, E12 и F12, E16 и F16 должны практически совпадать. Значительные расхождения в контрольных суммах указывают на ошибки в вычислениях либо неустойчивость алгоритма вычислений по отношению к вычислительной погрешности.

Расчетные формулы приведены на рисунке 7, результаты вычислений – на рисунке 8.

	A	B	C	D	E	F
1	4	-13	6	-24	=СУММ(A1:D1)	
2	-2	4	-4	18	=СУММ(A2:D2)	
3	-8	7	-20	104	=СУММ(A3:D3)	
4	=СУММ(A1:A3)	=СУММ(B1:B3)	=СУММ(C1:C3)	=СУММ(D1:D3)	=СУММ(E1:E3)	=СУММ(A4:D4)
5	=A1/\$A\$1	=B1/\$A\$1	=C1/\$A\$1	=D1/\$A\$1	=E1/\$A\$1	
6	=A2/\$A\$2-A5	=B2/\$A\$2-B5	=C2/\$A\$2-C5	=D2/\$A\$2-D5	=E2/\$A\$2-E5	
7	=A3/\$A\$3-A5	=B3/\$A\$3-B5	=C3/\$A\$3-C5	=D3/\$A\$3-D5	=E3/\$A\$3-E5	
8	=СУММ(A5:A7)	=СУММ(B5:B7)	=СУММ(C5:C7)	=СУММ(D5:D7)	=СУММ(E5:E7)	=СУММ(A8:D8)
9	=A5	=B5	=C5	=D5	=СУММ(A9:D9)	
10	=A6	=B6/\$B\$6	=C6/\$B\$6	=D6/\$B\$6	=E6/\$B\$6	
11	=A7	=B7/\$B\$7-B10	=C7/\$B\$7-C10	=D7/\$B\$7-D10	=E7/\$B\$7-E10	
12	=СУММ(A9:A11)	=СУММ(B9:B11)	=СУММ(C9:C11)	=СУММ(D9:D11)	=СУММ(E9:E11)	=СУММ(A12:D12)
13	=A9	=B9	=C9	=D9	=СУММ(A13:D13)	
14	=A10	=B10	=C10	=D10	=СУММ(A14:D14)	
15	=A11	=B11	=C11/\$C\$11	=D11/\$C\$11	=E11/\$C\$11	
16	=СУММ(A13:A15)	=СУММ(B13:B15)	=СУММ(C13:C15)	=СУММ(D13:D15)	=СУММ(E13:E15)	=СУММ(A16:D16)
17	$x_1 =$	=D13-C13*B19-B1				
18	$x_2 =$	=D14-C14*B19				
19	$x_3 =$	=D15				
20						
21	Проверка					
22	=A1*\$B\$17+B1*\$B\$18+C1*\$B\$19					
23	=A2*\$B\$17+B2*\$B\$18+C2*\$B\$19					
24	=A3*\$B\$17+B3*\$B\$18+C3*\$B\$19					

Рис. 7. Расчетные формулы для решения системы уравнений методом Гаусса.

	A	B	C	D	E	F
1	4	-13	6	-24	-27	
2	-2	4	-4	18	16	
3	-8	7	-20	104	83	
4	-6	-2	-18	98	72	72
5	1	-3,25	1,5	-6	-6,75	
6	0	1,25	0,5	-3	-1,25	
7	0	2,375	1	-7	-3,625	
8	1	0,375	3	-16	-11,625	-11,625
9	1	-3,25	1,5	-6	-6,75	
10	0	1	0,4	-2,4	-1	
11	0	0	0,02105	-0,5474	-0,5263	
12	1	-2,25	1,92105	-8,9474	-8,2763	-8,2763
13	1	-3,25	1,5	-6	-6,75	
14	0	1	0,4	-2,4	-1	
15	0	0	1	-26	-25	
16	1	-2,25	2,9	-34,4	-32,75	-32,75
17	$x_1 =$	59				
18	$x_2 =$	8				
19	$x_3 =$	-26				
20						
21	Проверка					
22	-24					
23	18					
24	104					

Рис. 8. Результаты вычислений и проверка найденных решений.

**Решение системы линейных алгебраических уравнений с использованием функции ЛИНЕЙН.** Пакет MS Excel располагает функцией ЛИНЕЙН, которая рассчитывает статистику для ряда с применением метода наименьших квадратов, чтобы вычислить прямую линию, наилучшим образом аппроксимирующую имеющиеся данные. Функция возвращает массив, который описывает полученную прямую. Эта функция может быть использована для решения систем линейных алгебраических уравнений.

Уравнение для прямой линии имеет следующий вид  $y = mx + b$ . В случае нескольких интервалов значений  $x$  уравнение прямой линии можно записать следующим образом:  $y = m_1x_1 + m_2x_2 + \dots + b$ , где зависимое значение  $y$  является функцией независимого значения  $x$ . Значения  $m$  – это коэффициенты, соответствующие каждой независимой переменной  $x$ , а  $b$  – это постоянная. Заметим, что  $y$ ,  $x$  и  $m$  могут быть векторами. Функция ЛИНЕЙН возвращает массив  $\{m_n; m_{n-1}; \dots; m_1; b\}$ .

Рассмотрим использование этой функции на примере.

**Пример 4.** Решить систему линейных алгебраических уравнений

$$\begin{cases} x_1 - 4x_2 - 3x_3 = 7 \\ -2x_1 - 1x_2 - 3x_3 = 4 \\ 5x_1 + 2x_2 + 7x_3 = -9 \end{cases}$$

**Решение.** Вычислим значение главного определителя этой системы. Используя функцию МОПРЕД, получим:  $-4,88498E-15$ . Это очень близкое к нулю число, следовательно матричный метод и метод Крамера применять нельзя. Если использовать метод Гаусса, то после вычислений, аналогичных описанным в предыдущем примере, получим строчку, все значения в которой равны нулю (см. рис. 9).

	A	B	C	D	E	F
1	1	-4	-3	7	1	Определитель:
2	-2	-1	-3	4	-2	-4,88498E-15
3	5	2	7	-9	5	
4	4	-3	1	2	4	4
5						
6	1	-4	-3	7	1	
7	0	4,5	4,5	-9	0	
8	0	4,4	4,4	-8,8	0	
9	1	4,9	5,9	-10,8	1	1
10						
11	1	-4	-3	7	1	
12	0	1	1	-2	0	
13	0	0	0	0	0	
14	1	-3	-2	5	1	1

**Рис. 9.** Решение системы из примера 4 методом Гаусса.

Это означает, что третье уравнение можно исключить из системы, так как ему удовлетворяет любое множество действительных значений  $x_1, x_2, x_3$ . Попробуем решить эту систему, используя функцию ЛИНЕЙН. Сначала следует вспомнить, что результатом этой функции будут значения, записанные слева направо следующим образом:  $x_3, x_2, x_1$ . Поэтому введем в ячейки A16:C16 соответствующие обозначения. Выделим ячейки A17:C17 и в строку формул введем следующую запись:

=ЛИНЕЙН(D1:D3;A1:C3;ЛОЖЬ;ЛОЖЬ)

Обязательно ввод формулы необходимо закончить нажатием клавиш Ctrl + Shift + Enter, что позволит представить результат в виде таблицы. В нашем случае получим значения  $x_1 = -3, x_2 = -4, x_3 = 2$ .

Вернемся к вычислениям методом Гаусса. Поскольку третьему уравнению удовлетворяет любое множество действительных решений, выберем значение переменной  $x_3$  равной 2. Вычислив значения двух оставшихся переменных, получим:  $x_1 = -3, x_2 = -4$ . Это означает, что мы нашли одно из множества решений данной системы. Проверка подтверждает правильность найденного решения.

Расчетные формулы приведены на рисунке 10, результаты вычислений – на рисунке 11.

	A	B	C	D	E	F
1	1	-4	-3	7	=СУММ(A1:D1)	Определитель:
2	-2	-1	-3	4	=СУММ(A2:D2)	=МОПРЕД(A1:C3)
3	5	2	7	-9	=СУММ(A3:D3)	
4	=СУММ(A1:A3)	=СУММ(B1:B3)	=СУММ(C1:C3)	=СУММ(D1:D3)	=СУММ(E1:E3)	=СУММ(A4:D4)
5						
6	=A1/\$A\$1	=B1/\$A\$1	=C1/\$A\$1	=D1/\$A\$1	=E1/\$A\$1	
7	=A2/\$A\$2-A6	=B2/\$A\$2-B6	=C2/\$A\$2-C6	=D2/\$A\$2-D6	=E2/\$A\$2-E6	
8	=A3/\$A\$3-A6	=B3/\$A\$3-B6	=C3/\$A\$3-C6	=D3/\$A\$3-D6	=E3/\$A\$3-E6	
9	=СУММ(A6:A8)	=СУММ(B6:B8)	=СУММ(C6:C8)	=СУММ(D6:D8)	=СУММ(E6:E8)	=СУММ(A9:D9)
10						
11	=A6	=B6	=C6	=D6	=E6	
12	=A7	=B7/\$B\$7	=C7/\$B\$7	=D7/\$B\$7	=E7/\$B\$7	
13	=A8	=B8/\$B\$8-B12	=C8/\$B\$8-C12	=D8/\$B\$8-D12	=E8/\$B\$8-E12	
14	=СУММ(A11:A13)	=СУММ(B11:B13)	=СУММ(C11:C13)	=СУММ(D11:D13)	=СУММ(E11:E13)	=СУММ(A14:D14)
15						
16	$x_3$	$x_2$	$x_1$			
17	=ЛИНЕЙН(D1:D3;	=ЛИНЕЙН(D1:D3;	=ЛИНЕЙН(D1:D3;			
18						Проверка
19	$x_1$	=D11-C11*B21-B1				=МУМНОЖ(A1:C3;B19:B21)
20	$x_2$	=D12-C12*B21				=МУМНОЖ(A1:C3;B19:B21)
21	$x_3$	2				=МУМНОЖ(A1:C3;B19:B21)

Рис. 10. Расчетные формулы решения системы уравнений.

	A	B	C	D	E	F
1	1	-4	-3	7	1	Определитель:
2	-2	-1	-3	4	-2	-4,88498E-15
3	5	2	7	-9	5	
4	4	-3	1	2	4	4
5						
6	1	-4	-3	7	1	
7	0	4,5	4,5	-9	0	
8	0	4,4	4,4	-8,8	0	
9	1	4,9	5,9	-10,8	1	1
10						
11	1	-4	-3	7	1	
12	0	1	1	-2	0	
13	0	0	0	0	0	
14	1	-3	-2	5	1	1
15						
16	$x_3$	$x_2$	$x_1$			
17	2	-4	-3			
18						Проверка
19	$x_1$	-3				7
20	$x_2$	-4				4
21	$x_3$	2				-9

Рис. 11. Результаты вычислений.

Следует заметить, что при решении таких систем найденное решение не единственно. В этом легко убедиться, если вместо числа 2 в ячейку B21 подставить другое число, например, -5. Тогда изменятся значения ячеек B19 и B20, но результат проверки останется неизменным. Таким образом, метод Гаусса позволяет найти множество решений системы линейных ал-

гебраических уравнений (в том случае, когда одно из уравнений из системы можно исключить), тогда как использование функции ЛИНЕЙН позволяет найти одно решение системы.

### Задания для самостоятельной работы

**Задание 1.** Решите систему линейных алгебраических уравнений, обоснуйте выбор метода решения. Выполните проверку найденного решения.

$$1. \begin{cases} 4,33x_1 - 1,12x_2 - 1,08x_3 + 1,14x_4 = 3,52 \\ -1,12x_1 + 4,33x_2 + 0,24x_3 - 1,22x_4 = 1,57 \\ -1,08x_1 + 0,24x_2 + 7,21x_3 - 3,22x_4 = 0,54 \\ 1,14x_1 - 1,22x_2 - 3,22x_3 + 5,43x_4 = -1,09 \end{cases}$$

$$2. \begin{cases} 0,135x_1 + 0,188x_2 + 0,191x_3 + 0,178x_4 = 0,3516 \\ 0,188x_1 + 0,262x_2 + 0,265x_3 + 0,247x_4 = 0,4887 \\ 0,191x_1 + 0,265x_2 + 0,281x_3 + 0,266x_4 = 0,5105 \\ 0,178x_1 + 0,247x_2 + 0,266x_3 + 0,255x_4 = 0,4818 \end{cases}$$

$$3. \begin{cases} 2x_1 - 4x_2 - 3,25x_3 + x_4 = 4,84 \\ 3x_1 - 3x_2 - 4,3x_3 + 8x_4 = 8,89 \\ x_1 - 5x_2 + 3,3x_3 - 20x_4 = -14,01 \\ 2,5x_1 - 4x_2 + 2x_3 - 3x_4 = -20,29 \end{cases}$$

$$4. \begin{cases} 5,7x_1 - 7,8x_2 - 5,6x_3 - 8,3x_4 = 2,7 \\ 6,6x_1 + 13,1x_2 - 6,3x_3 + 4,3x_4 = -5,5 \\ 14,7x_1 - 2,8x_2 + 5,6x_3 - 12,1x_4 = 8,6 \\ 8,5x_1 + 12,7x_2 - 23,7x_3 + 5,7x_4 = 14,7 \end{cases}$$

$$5. \begin{cases} 7,3x_1 - 8,1x_2 + 12,7x_3 - 6,7x_4 = 8,8 \\ 11,5x_1 + 6,2x_2 - 8,3x_3 + 9,2x_4 = 21,5 \\ 8,2x_1 - 5,4x_2 + 4,3x_3 - 2,5x_4 = 6,2 \\ 2,4x_1 + 11,5x_2 - 3,3x_3 + 14,2x_4 = -6,2 \end{cases}$$

$$6. \begin{cases} 6,4x_1 + 7,2x_2 - 8,3x_3 + 42x_4 = 2,23 \\ 5,8x_1 - 8,3x_2 + 14,3x_3 - 6,2x_4 = 17,1 \\ 8,6x_1 + 7,7x_2 - 18,3x_3 + 8,8x_4 = -5,4 \\ 13,2x_1 - 5,2x_2 - 6,5x_3 + 12,2x_4 = 6,5 \end{cases}$$

$$7. \begin{cases} 8,2x_1 - 3,2x_2 + 14,2x_3 + 14,8x_4 = -8,4 \\ 5,6x_1 - 12x_2 + 15x_3 - 6,4x_4 = 4,5 \\ 5,7x_1 + 3,6x_2 - 12,4x_3 - 2,3x_4 = 3,3 \\ 6,8x_1 + 13,2x_2 - 6,3x_3 - 8,7x_4 = 14,3 \end{cases}$$

$$8. \begin{cases} 3,8x_1 + 14,2x_2 + 6,3x_3 - 15,5x_4 = 2,8 \\ 8,3x_1 - 6,6x_2 + 5,8x_3 + 12,2x_4 = -4,7 \\ 6,4x_1 - 8,5x_2 - 4,3x_3 + 8,8x_4 = 7,7 \\ 17,1x_1 - 8,3x_2 + 14,4x_3 - 7,2x_4 = 13,5 \end{cases}$$

$$9. \begin{cases} 1,7x_1 + 10x_2 - 1,3x_3 + 2,1x_4 = 3,1 \\ 3,1x_1 + 1,7x_2 - 2,1x_3 + 5,4x_4 = 2,1 \\ 3,3x_1 - 7,7x_2 + 4,4x_3 - 5,1x_4 = 1,9 \\ 10x_1 - 20,1x_2 + 20,4x_3 + 1,7x_4 = 1,8 \end{cases}$$

$$10. \begin{cases} 6,1x_1 + 6,2x_2 - 6,3x_3 + 6,4x_4 = 6,5 \\ 1,1x_1 - 1,5x_2 + 2,2x_3 - 3,8x_4 = 4,2 \\ 5,1x_1 - 5,0x_2 + 4,9x_3 - 4,8x_4 = 4,7 \\ 1,8x_1 + 1,9x_2 + 2,0x_3 - 2,1x_4 = 2,2 \end{cases}$$

**Задание 2.** Решите систему линейных алгебраических уравнений методами Гаусса, Крамера и матричным методом. Выполните проверку найденных решений.

$$1. \begin{cases} x_1 + 3x_2 + 5x_3 - 3x_4 + 2x_5 = 6 \\ x_1 + x_2 + 3x_3 + 5x_4 - 2x_5 = 1 \\ x_1 + 5x_2 + 4x_3 - 2x_4 + 3x_5 = -4 \\ x_1 + 3x_2 + 2x_3 + x_5 = -1 \\ 2x_1 + 12x_2 + 12x_3 - x_4 + 5x_5 = 3 \end{cases}$$

$$2. \begin{cases} 5x_1 - x_2 + 5x_3 - 3x_4 + 2x_5 = -29 \\ 5x_1 + 4x_3 + 5x_4 - 7x_5 = 40 \\ 3x_1 + x_2 + 4x_3 - 2x_4 = 65 \\ 15x_1 + 4x_2 + 2x_3 + x_5 = 23 \\ 2x_1 + 9x_2 + 12x_3 - x_4 + 5x_5 = 24 \end{cases}$$



$$3. \begin{cases} x_1 + 2x_2 - 2x_3 + 3x_4 - 3x_5 = 1 \\ 2x_1 + 2x_2 + 3x_3 - x_4 + 4x_5 = 1 \\ 4x_2 + x_3 - x_4 + 2x_5 = 1 \\ 2x_1 - 3x_2 + 4x_3 - 2x_4 - 3x_5 = 10 \\ x_1 + 2x_2 - 4x_3 + 6x_4 + 2x_5 = 20 \end{cases}$$

$$4. \begin{cases} 3x_1 + 8x_2 + 2x_3 + 5x_4 + 5x_5 = 12 \\ 5x_1 + 4x_2 - x_3 + 4x_4 = 3 \\ x_2 - 3x_3 + x_4 - 5x_5 = 0 \\ 2x_1 - 6x_3 - 5x_4 + 8x_5 = -3 \\ 4x_1 - 6x_3 + 2x_4 + 4x_5 = -15 \end{cases}$$

$$5. \begin{cases} -x_1 + x_2 - x_3 + 3x_4 - x_5 = 2 \\ x_2 + x_3 + x_4 + x_5 = 5 \\ x_1 + 2x_2 + x_3 + x_5 = -2 \\ x_1 + x_4 + x_5 = 5 \\ -4x_1 + x_2 + 2x_3 + 2x_4 = 1 \end{cases}$$

$$6. \begin{cases} 2x_1 + 3x_2 + x_3 - 2x_4 - 5x_5 = -5 \\ 2x_1 + 4x_2 + 3x_3 - 3x_4 - 5x_5 = 2 \\ x_1 + 3x_2 + 3x_3 - 2x_4 - 5x_5 = 5 \\ 4x_2 - x_3 - 3x_4 - 4x_5 = 11 \\ 5x_1 + 3x_2 + x_3 - 2x_4 - 5x_5 = 7 \end{cases}$$

$$7. \begin{cases} -4x_1 + x_2 + 2x_3 + x_4 = 2 \\ 2x_1 + 6x_2 - 2x_3 + x_4 + 4x_5 = 5 \\ 4x_1 + x_2 - 2x_3 - x_4 + 2x_5 = 4 \\ 2x_1 + 3x_2 - 3x_3 - x_4 + 3x_5 = 1 \\ 2x_2 + 3x_3 + 5x_4 = -1 \end{cases}$$

$$8. \begin{cases} -6x_1 + 2x_2 + 12x_3 + 5x_5 = 10 \\ -x_1 + 3x_2 + 5x_3 + 2x_4 + 4x_5 = -1 \\ -x_1 + x_2 - x_3 + 2x_4 + 3x_5 = 7 \\ x_2 - x_3 + 3x_4 + 4x_5 = -13 \\ 4x_2 - x_3 + x_5 = -13 \end{cases}$$

$$9. \begin{cases} 2x_1 - 2x_2 + x_3 - 4x_4 - 5x_5 = -22 \\ 3x_2 + 2x_4 - 4x_5 = 71 \\ 6x_1 + x_2 + x_4 - 5x_5 = 12 \\ -3x_1 + 2x_2 + 2x_3 - 2x_4 + 4x_5 = 2 \\ -x_1 + 3x_2 + 2x_3 + 2x_5 = 4 \end{cases}$$

$$10. \begin{cases} 2x_1 - 5x_2 + 2x_3 + x_4 + 3x_5 = -11 \\ -2x_1 - 6x_2 + 4x_3 - 2x_4 + 7x_5 = 20 \\ 2x_1 - 6x_2 + 3x_3 + x_4 + 3x_5 = 17 \\ x_1 - 2x_2 + 4x_3 + x_4 - 5x_5 = -4 \\ 3x_1 - x_2 + 12x_3 + 4x_4 - x_5 = 3 \end{cases}$$

**Задание 3.** Решите систему линейных алгебраических уравнений методом Гаусса и матричным методом. Выполните проверку найденных решений.

$$1. \begin{cases} x_1 + 2x_2 + x_3 + 9x_4 + 15x_5 + 2x_6 + 4x_7 + 4x_8 - 20x_9 - 24x_{10} = -33 \\ -4x_1 - 4x_3 + 2x_4 + 22x_5 + 3x_6 + 2x_7 + 16x_{10} = -2 \\ -3x_2 + x_3 - 11x_5 + 4x_6 + 8x_7 - 26x_8 - 11x_9 = 15 \\ -2x_1 + x_2 + 3x_3 + 2x_4 - 8x_5 + 3x_6 - 3x_7 - 18x_8 - 7x_9 - 9x_{10} = 55 \\ -4x_1 - 12x_2 + 4x_3 + x_4 - 20x_5 + 14x_6 - 3x_7 + 3x_8 + 15x_9 + 12x_{10} = 22 \\ -15x_1 + 10x_2 + 4x_3 + 3x_4 + 14x_5 - 5x_6 - 6x_7 + 5x_8 = -14 \\ 2x_1 + 14x_2 - 4x_3 - 5x_4 - 8x_5 - x_6 + 11x_7 - 11x_8 - 20x_9 - 18x_{10} = 17 \\ x_1 - 7x_2 + 9x_3 - 5x_4 - 2x_5 + 12x_7 + 2x_8 + 2x_9 + 2x_{10} = -28 \\ -2x_2 - 5x_4 - x_5 - x_6 + 19x_7 - 3x_8 - 6x_9 - 4x_{10} = 19 \\ -3x_1 - 3x_2 + 4x_4 - x_5 - 6x_6 - 7x_{10} = -1 \end{cases}$$

$$2. \begin{cases} -8x_1 - 3x_3 - 16x_4 + 15x_5 + 2x_6 - 24x_7 + 4x_8 - 20x_9 + 4x_{10} = 8 \\ -x_1 + 11x_2 - 3x_3 + 11x_4 + 22x_5 + 3x_6 + 16x_7 = 2 \\ 82x_2 - 3x_3 - 6x_4 - 11x_5 + 4x_6 - 26x_8 - 11x_9 + 8x_{10} = -6 \\ -x_2 + 7x_4 - 8x_5 + 3x_6 - 9x_7 - 18x_8 - 7x_9 - 3x_{10} = 4 \\ -6x_1 - 12x_2 + 5x_4 - 20x_5 + 14x_6 + 12x_7 + 3x_8 + 15x_9 - 3x_{10} = -5 \\ -14x_1 + 11x_2 + 4x_3 + 6x_4 + 14x_5 - 5x_6 + 5x_8 - 6x_{10} = 8 \\ -20x_1 + 14x_2 - 4x_3 + 7x_4 - 8x_5 - 18x_7 - 11x_8 + 20x_9 - 11x_{10} = -7 \\ -35x_1 - 7x_2 + 9x_3 + 4x_4 - 2x_5 + 2x_7 + 2x_8 + 2x_9 + 12x_{10} = -3 \\ -2x_2 + 4x_4 - x_5 - 4x_7 - 3x_8 - 6x_9 + 19x_{10} = -2 \\ -3x_2 + x_4 - x_5 + x_7 + x_8 + x_9 + x_{10} = -1 \end{cases}$$

$$\begin{array}{l}
 3. \left\{ \begin{array}{l}
 -17x_1 - 12x_3 + x_4 + 2x_5 + 2x_6 + 6x_7 - 7x_8 - 2x_9 = -23 \\
 x_1 - 10x_3 + 6x_4 + 5x_6 - 6x_7 + 5x_8 + x_9 + 27x_{10} = -18 \\
 -5x_1 + 3x_2 - 10x_3 + 5x_4 + 2x_5 - x_6 = 21 \\
 -6x_1 + 12x_2 - x_3 + 6x_4 + x_5 + 2x_6 + x_7 + x_8 + x_9 + x_{10} = -11 \\
 4x_1 - 70x_2 + 3x_3 + 10x_4 - x_8 - 5x_9 - 3x_{10} = -3 \\
 2x_1 - 33x_2 + 3x_3 - x_4 - x_6 + 2x_7 - x_8 - x_9 - x_{10} = 4 \\
 4x_1 - 25x_2 + 3x_3 - 2x_4 + x_5 + x_6 + x_8 + x_9 + x_{10} = 7 \\
 -x_1 - 16x_2 - 2x_3 - 3x_4 + x_5 + x_6 + x_8 + x_9 + 8x_{10} = 2 \\
 -103x_2 - 2x_3 + x_5 + x_7 + x_9 + 3x_{10} = 2 \\
 76x_2 - 2x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + 5x_{10} = 10
 \end{array} \right. \\
 \\
 4. \left\{ \begin{array}{l}
 7x_1 - x_3 + x_5 + 2x_7 + 3x_8 + 25x_9 - 10x_{10} = 33 \\
 6x_1 + x_2 - 5x_3 - 2x_4 - x_5 - 2x_6 - 2x_7 - 3x_8 - 11x_{10} = 19 \\
 14x_1 - x_3 + 3x_4 + 2x_5 + x_6 + 6x_7 + 3x_8 - 2x_9 - 23x_{10} = 18 \\
 30x_3 + 5x_4 + x_5 - 2x_6 + x_7 - 9x_8 + x_9 = -12 \\
 -x_1 + x_4 + x_5 + x_6 + x_7 - 2x_9 = -14 \\
 -x_1 - 4x_2 + x_3 + x_4 + x_6 - x_8 + 119x_{10} = 5 \\
 -32x_1 - 3x_2 + x_3 + x_4 - 9x_6 + x_8 - 6x_9 - 21x_{10} = 0 \\
 11x_1 - 2x_2 + 4x_3 + 5x_4 + 15x_5 + 11x_6 - 12x_7 + 17x_8 - x_9 + 2x_{10} = 0 \\
 -15x_1 - 24x_2 - x_3 - 2x_4 - 3x_5 - 3x_6 + 23x_8 - 2x_9 - x_{10} = -27 \\
 -91x_1 - 21x_2 - x_3 - x_4 + x_6 + 3x_8 + 17x_9 - 7x_{10} = -93
 \end{array} \right. \\
 \\
 5. \left\{ \begin{array}{l}
 x_2 + x_3 - 15x_5 + x_6 + 2x_7 + x_9 + x_{10} = 220 \\
 x_1 + x_3 + 20x_6 - 19x_7 + 3x_8 + 3x_9 - 3x_{10} = -51 \\
 -13x_1 + 14x_3 + 7x_4 + 4x_5 + x_6 - 2x_7 + x_8 + x_9 + 2x_{10} = -164 \\
 2x_1 + x_2 + x_3 - 4x_4 + 53x_5 + 3x_6 + x_8 + x_9 + 3x_{10} = 468 \\
 5x_1 + x_2 + x_3 - 16x_4 + 12x_5 + x_6 + x_8 + x_9 + x_{10} = -179 \\
 9x_1 - 107x_2 + x_3 - 19x_4 + x_5 + 7x_6 - x_7 + x_8 + 2x_{10} = -112 \\
 46x_1 - 68x_2 + 4x_3 - 82x_4 + 2x_5 + x_6 + 2x_7 + x_9 - 2x_{10} = -35 \\
 4x_2 + x_3 + 5x_4 - 2x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} = 128 \\
 -89x_1 - 2x_2 + x_3 + x_4 - 5x_5 - 2x_6 + 4x_7 + x_8 + x_9 = 95 \\
 15x_1 - 77x_2 - 23x_3 - x_4 + 2x_7 + 26x_8 + 15x_9 - 8x_{10} = 709
 \end{array} \right.
 \end{array}$$

$$6. \begin{cases} 5x_1 + 3x_3 + 13x_4 - 18x_5 - 8x_6 - 71x_7 + 109x_8 + 2x_{10} = -100 \\ 3x_2 + 3x_3 + 11x_4 + 3x_5 + 3x_6 + 3x_7 + 6x_8 + 3x_9 + 3x_{10} = -53 \\ 3x_2 - 43x_5 + 3x_6 + 3x_7 + 3x_8 + 3x_9 + 3x_{10} = 11 \\ 24x_1 - 3x_2 - 2x_4 - 37x_5 - 55x_7 + 6x_8 + 4x_9 + 3x_{10} = 17 \\ 3x_1 + 3x_2 + 3x_3 - 5x_5 + 3x_6 + 3x_8 + 3x_9 + 3x_{10} = 103 \\ -5x_1 + 4x_2 + 3x_3 + 3x_5 + 3x_7 - 3x_9 + 3x_{10} = -9 \\ 43x_1 - 8x_2 + 3x_3 + 3x_4 + 3x_5 + 73x_6 + 3x_7 + 3x_8 - 3x_9 + 2x_{10} = -64 \\ 3x_1 + 3x_2 + 3x_3 - 12x_4 + 3x_5 + 3x_6 + 3x_8 - 5x_9 + 3x_{10} = -35 \\ -5x_1 + 3x_2 + 3x_3 + x_4 - 3x_5 + 4x_6 + x_8 + 3x_9 + 3x_{10} = 18 \\ 3x_1 + 3x_2 + 3x_4 + 3x_6 + 3x_7 + 3x_{10} = 72 \end{cases}$$

$$7. \begin{cases} -2x_2 - 2x_3 + 4x_4 - 2x_5 + 3x_6 - 3x_8 + x_9 + 10x_{10} = 15 \\ -2x_1 - x_2 - 3x_3 + x_4 - 2x_5 + 2x_6 + 2x_7 + 2x_8 + x_9 = 18 \\ -6x_1 - 2x_2 - 3x_3 + 4x_4 - 2x_5 - 8x_7 + 2x_8 + 2x_9 - 5x_{10} = -12 \\ -2x_1 - 2x_3 + 5x_4 - 2x_5 - 7x_6 + 2x_7 - 7x_8 + 2x_9 + x_{10} = -12 \\ -2x_2 - 2x_3 + 4x_4 - 2x_5 + 2x_6 + x_7 + 2x_8 + 2x_9 + 2x_{10} = 10 \\ -2x_1 - 2x_3 - 4x_4 - 2x_5 - 10x_6 + 2x_7 + 10x_8 - 10x_9 + 2x_{10} = -19 \\ -2x_1 - 2x_2 - x_3 + x_4 + 16x_6 + 2x_7 + 11x_8 + 2x_9 - 2x_{10} = 11 \\ -2x_1 - 2x_2 - x_3 - 4x_4 - 2x_5 + 2x_6 + 5x_7 + 2x_9 + 4x_{10} = -30 \\ x_1 - 2x_2 - 2x_3 - 2x_5 + 2x_6 + 2x_7 + 2x_8 + 2x_9 + 2x_{10} = 18 \\ -2x_1 - 2x_2 - 4x_4 + 3x_5 + 5x_6 + 2x_7 + x_8 - 6x_{10} = -14 \end{cases}$$

$$8. \begin{cases} -5x_1 + 7x_2 + 11x_3 + 2x_4 - 15x_5 - 3x_6 + 20x_7 - 7x_8 - 27x_9 + 2x_{10} = 5 \\ -7x_2 - x_3 + 3x_4 + 20x_6 + 5x_7 + 4x_8 + 3x_9 + 2x_{10} = 14 \\ 2x_1 + 8x_2 - x_3 - 6x_4 + x_6 + 4x_7 + 5x_8 + 3x_9 + 2x_{10} = -8 \\ -3x_1 + 5x_2 - x_3 + 2x_4 - 3x_6 + 2x_7 - 6x_8 + 3x_9 + 2x_{10} = -27 \\ 81x_1 + 11x_3 + 3x_4 - 17x_5 + 20x_6 + 3x_7 - 7x_8 + 3x_9 + 2x_{10} = -31 \\ -4x_1 - 7x_2 + 14x_3 - 6x_4 + x_6 + 4x_8 - 27x_9 + 2x_{10} = -53 \\ 7x_2 + 11x_3 + 2x_4 - 16x_5 - 3x_6 + 2x_7 + 5x_8 + 53x_9 + x_{10} = 46 \\ 3x_1 - 8x_2 + 14x_3 + 3x_4 + 20x_6 - x_7 - 6x_8 - 11x_9 + 2x_{10} = 45 \\ 5x_1 + 6x_2 + 11x_3 - 6x_4 + 3x_5 + x_6 - 2x_7 + x_8 - 23x_9 + 2x_{10} = 11 \\ 3x_1 - 6x_2 + 14x_3 - x_4 - 20x_6 - 3x_7 + 26x_8 - 10x_9 + 18x_{10} = 7 \end{cases}$$

$$\begin{array}{l}
 9. \left\{ \begin{array}{l}
 x_1 + x_3 + 11x_4 - 15x_5 + x_6 + 2x_7 + x_9 + 7x_{10} = 14 \\
 x_2 + 3x_3 - 12x_4 + 20x_6 - 19x_7 + 3x_8 + 8x_{10} = 6 \\
 -13x_2 + x_3 + 3x_4 + 4x_5 + x_6 - 2x_7 + 2x_8 - 6x_{10} = 9 \\
 x_1 + 2x_2 + x_3 - 4x_4 + 53x_5 + 3x_6 + 15x_8 + x_9 - 6x_{10} = -9 \\
 x_1 + 5x_2 - 3x_3 - 15x_4 + 12x_5 + x_6 + 2x_8 + x_9 + 2x_{10} = 16 \\
 -107x_1 + 9x_2 + 16x_4 - 5x_5 - 2x_6 + 4x_7 + x_8 - 2x_9 - x_{10} = -7 \\
 -68x_1 + 46x_2 + 4x_3 - 82x_4 + 497x_5 + 26x_8 - 77x_9 = -2 \\
 4x_1 - 2x_3 + 4x_4 - 2x_5 + 2x_6 + x_7 + 5x_8 + 4x_9 + 5x_{10} = 11 \\
 -2x_1 - 89x_2 + x_3 + 2x_4 + x_5 + 7x_6 - x_7 + x_8 - 98x_9 + 3x_{10} = 18 \\
 -77x_1 + 15x_2 - x_4 + 2x_5 + x_6 + 2x_7 - 5x_9 + 4x_{10} = 4 \\
 \\
 10. \left\{ \begin{array}{l}
 -13x_1 + 14x_3 + 4x_5 + x_6 - 2x_7 - 5x_9 + 2x_{10} = -2 \\
 x_1 + x_3 - 7x_4 + 20x_6 - 19x_7 + 3x_9 - 3x_{10} = -3 \\
 46x_1 - 91x_2 + 4x_3 + 17x_4 + 2x_5 + x_6 + 2x_7 - 4x_8 + 8x_9 - 2x_{10} = -5 \\
 2x_1 + x_2 + x_3 + 53x_5 + 3x_6 - x_8 + 4x_9 + 3x_{10} = -2 \\
 5x_1 + x_2 + x_3 + 12x_5 + x_6 - 2x_8 + 6x_9 + x_{10} = 11 \\
 9x_1 - 63x_2 + x_3 + x_5 + 7x_6 - x_7 - 3x_8 - 5x_9 + 2x_{10} = 17 \\
 -89x_1 - 2x_2 + x_3 + x_4 - 5x_5 - 2x_6 + 4x_7 - 7x_8 + x_9 = 0 \\
 4x_2 + x_3 + 5x_4 - 2x_5 + x_6 + x_7 + 5x_8 - 7x_9 + x_{10} = -17 \\
 x_2 + x_3 + 7x_4 - 15x_5 + x_6 + 2x_7 + 2x_9 + x_{10} = 3 \\
 15x_1 - 85x_2 - 23x_3 - x_4 + 2x_7 + 14x_8 - 8x_{10} = -11
 \end{array} \right.
 \end{array}$$

### Литература

1. Гурский Е.И. и др. Руководство к решению задач по высшей математике. В 2-х ч. Ч. 1. – Мн.: Выш. шк., 1989. – 449 с.
2. Заварыкин В.М. и др. Численные методы: Учеб. пособие для студентов физ.-мат. спец. – М.: Просвещение, 1990. – 176 с.
3. Турчак Л.И. Основы численных методов: Учеб. пособие. – М.: Наука, 1987. – 320 с.

## ПРИЛОЖЕНИЕ

Тема	Автор	Название	Город	Издательство	Год	Страниц	Тираж	Цена	Количество
Образование	Бартон В.И.	Логика. Учебное пособие	Минск	Новое знание	2001	336	2000	6975	10
Образование	Берков В.Ф.	История логики	Минск	Новое знание	2001	176	3000	11250	10
Культурология	Берков В.Ф.	Культура диалога	Минск	Новое знание	2002	144	1000	6825	15
Образование	Берков В.Ф.	Логика. Учебник для вузов	Минск	ТетраСистемс	2002	416	2000	10500	20
История	Бобраков А.В.	Вокруг Европы по волнам истории	Москва	Огни	2003	342	350	9150	5
Лингвистика	Варпахович Л.В.	Лингвистика в таблицах и схемах	Минск	Новое знание	2003	128	5010	4875	15
История	Васильева Н.В.	Хозяева и гости дворянских имений Крыма	Москва	Глобус	2002	232	1000	7500	5
Философия	Вишневский М.И.	Введение в философию образования	Могилев	МГУ	2002	160	3000	10275	3
Психология	Выготский В.	Психология искусства	Минск	Современное слово	1998	480	2000	6000	10
Лингвистика	Гируцкий А.А.	Введение в языкознание	Минск	ТетраСистемс	2001	288	3100	6000	30
Педагогика	Гурьянова М.П.	Сельская школа и социальная педагогика	Минск	Амалфея	2000	448	1000	4125	10
Философия	Демидов А.	Феномены человеческого бытия	Минск	Экономпресс	1999	180	1000	3000	25
История	Еремина Т.С.	Мир русских икон: История, предания	Москва	ТЕРРА	2003	352	1000	15750	5

Тема	Автор	Название	Город	Издательство	Год	Страниц	Тираж	Цена	Количество
История	Еремина Т.С.	Мир русских монастырей: История, предания	Москва	ТЕРРА	2002	328	1000	20250	5
Искусство	Занусси К.	Между ярмаркой и салоном	Минск	ЕГУ	2003	223	1000	9000	15
История	Захаров В.А.	Мальтийский орден	Москва	Огни	2003	252	500	22500	5
Философия	Зеленкова И.Л.	Этика	Минск	ТетраСистемс	2001	480	4000	9000	30
История	Зелинский Ф.Ф.	Эллинская религия	Минск	Экономпресс	2003	330	3000	10500	3
Лингвистика	Иванов Е.Е.	Языковая природа афоризма. Очерки и извлечения	Могилев	МГУ	2001	440	300	7125	10
Психология	Кандыбович Л.А.	История психологии в Беларуси	Минск	Тесей	2002	208	1000	7650	30
История	Кеслер Я.А.	Русская цивилизация	Москва	ЭкоПресс	2002	432	3000	16500	10
Лингвистика	Клубков П.	Говорите, пожалуйста, правильно	СПб	Норинт	2002	192	6000	1875	10
Психология	Козубовский В.М.	Общая психология: методология, сознание, деятельность	Минск	Амалфея	2003	224	1000	9600	15
Психология	Коломинский Я.Л.	Психология взаимоотношений в малых группах	Минск	ТетраСистемс	2001	432	1100	7200	20
Образование	Костюкович П.И.	Религиоведение. Учебное пособие	Минск	Новое знание	2002	192	2000	5100	30
История	Кузнецов И.Н.	История Беларуси в документах и материалах	Минск	Амалфея	2000	672	4000	7125	15

Тема	Автор	Название	Город	Издательство	Год	Страниц	Тираж	Цена	Количество
История	Локотко А.И.	Историко-культурные регионы Беларуси	Минск	ЕГУ	2002	228	1000	12600	10
История	Лукиянов П.Г.	Новейшая история стран Центральной и Восточной Европы	Могилев	МГУ	2002	354	3000	13125	25
Философия	Малер А.М.	Стратегии сакрального смысла	Москва	Парад	2003	268	1000	10500	10
лингвистика	Мартынов В.В.	Основы семантического кодирования	Минск	ЕГУ	2001	140	1000	4875	10
Культурология	Мартынов В.Ф.	Мировая художественная культура	Минск	ТетраСистемс	2000	288	5000	5625	10
Философия	Мартынов М.И.	Философия: Вопросы и ответы	Минск	ТетраСистемс	2001	272	3000	6000	25
Лингвистика	Мечковская Н.Б.	Общее языкознание	Минск	Амалфея	2000	368	1000	5625	40
Искусство	Миронова Л.Н.	Цвет в изобразительном искусстве	Минск	Беларусь	2002	151	1000	11850	5
Образование	Мишаткина Т.В.	Этика	Минск	Новое знание	2002	508	1000	9000	10
Культурология	Морозов И.	Основы культурологии. Архетипы культуры	Минск	ТетраСистемс	2001	608	3100	10125	30
История	Мурьянов М.Ф.	Гимнография Киевской Руси	Москва	Наука	2003	453	740	20250	20
Лингвистика	Назаров Р.В.	Несекретные секреты: беседы о родной речи	СПб	Лицей	1998	240	5000	3750	20



Тема	Автор	Название	Город	Издательство	Год	Страниц	Тираж	Цена	Количество
История	Петросян Ю.А.	Османская империя: могущество и гибель	Москва	Эксмо	2003	416	3000	9750	5
Педагогика	Пионова Р.С.	Педагогика высшей школы	Минск	Университетское	2002	256	500	9750	30
Образование	Плотницкий М.И.	Экономика	Минск	Новое знание	2002	431	1000	9000	15
Психология	Поликарпов В.	Психология первой любви	Минск	Экономпресс	2002	208	1000	6750	5
бизнес	Потемкин А.П.	Виртуальная экономика	Москва	Порог	2004	448	5000	9750	30
Педагогика	Прокопьев И.И.	Педагогика	Минск	ТетраСистемс	2002	544	3000	12750	40
Педагогика	Пуйман С.А.	Педагогика	Минск	ТетраСистемс	2001	256	4000	5850	40
Лингвистика	Ремнева М.Л.	Пути развития русского литературного языка XI–XVII вв.	Москва	МГУ	2003	334	3000	16500	10
Культурология	Сарабянов Д.В.	Россия и Запад. Историко-художественные связи	Москва	Искусство	2003	296	1000	152250	15
Философия	Семенов Н.С.	Философские традиции Востока	Минск	ЕГУ	2004	303	1000	12750	5
Психология	Сизанов А.Н.	Познай себя	Минск	Польмя	2001	592	3000	7125	30
Психология	Сперлинг А.П.	Психология	Минск	Попурри	2002	432	3000	8850	40
Лингвистика	Супрун А.Е.	Исследования по лингвистике текста	Минск	БГУ	2001	308	100	6750	10
Философия	Фурс В.Н.	Контуры современной критической теории	Минск	ЕГУ	2002	164	1000	10500	10

Репозиторий ВГУ

Тема	Автор	Название	Город	Издательство	Год	Страниц	Тираж	Цена	Количество
Педагогика	Харламов И. Ф.	Педагогика. Компактный учебный курс	Минск	Университетское	2001	272	500	4050	10
Лингвистика	Херманн Фр.В.	Фундаментальная онтология языка	Минск	ЕГУ	2001	168	48	3600	10
Философия	Чесноков Н.Г.	Этика и истина.	Могилев	МГУ	2002	176	1000	3975	15
Образование	Чеховских М.И.	Основы психологии: Учебное пособие	Минск	Новое знание	2002	224	1000	5850	10
История	Чикалов Р.А.	Западная Европа и Соединённые Штаты Америки	Минск	Тесей	2000	504	3000	8250	10
Бизнес	Шрамко О.	Философия богатства	СПб	Галарт	2004	208	1000	6750	10
Философия	Шуман А.	Философская логика: истоки и эволюция	Минск	Экономпресс	2001	368	1000	9150	20
Лингвистика	Шумовский Т.А.	Странствия слов	Москва	Протей	2004	336	3000	18750	10

Репозиторий ВГУ