

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

О. Ф. Ускова, Н. А. Каплиева,
О. Д. Горбенко

**НАЧАЛА СТРУКТУРНОГО ПРОГРАММИРОВАНИЯ
НА ЯЗЫКЕ C++**

Задачник-практикум

Воронеж
Издательский дом ВГУ
2019

УДК 681.3.06(075)
ББК 32.973
У75

Р е ц е н з е н т –

д-р физ.-мат. наук, профессор кафедры математического и прикладного анализа факультета ПММ Воронежского государственного университета
А. И. Шашкин

Ускова О. Ф.

У75 Начала структурного программирования на языке С++ : задачник-практикум / О. Ф. Ускова, Н. А. Каплиева, О. Д. Горбенко. – Воронеж : Издательский дом ВГУ, 2019. – 261 с.

Задачник-практикум содержит материал практической направленности по языку программирования С++, необходимый для усвоения курса «Информатика и программирование». В него включены некоторые справочные сведения, вопросы для самопроверки, упражнения и задачи с решениями, тренировочные задания, задания для самостоятельной работы на компьютере.

Рекомендован студентам первого курса очной формы обучения факультета ПММ ВГУ. Для направлений: 01.03.02 – Прикладная математика и информатика, 02.03.02 – Фундаментальная информатика и информационные технологии, 01.03.03 – Механика и математическое моделирование, 38.03.05 – Бизнес-информатика.

УДК 681.3.06(075)
ББК 32.973

© Ускова О. Ф., Каплиева Н. А.,
Горбенко О. Д., 2019
© Воронежский государственный
университет, 2019
© Оформление.
Издательский дом ВГУ, 2019

Глава 1. ЛИНЕЙНЫЕ АЛГОРИТМЫ

1.1. Задачи с решениями

1. Вычислить среднюю скорость движения велосипедиста от пункта А до пункта С через пункт В. Расстояние между пунктами А и В равно S_1 км, расстояние между пунктами В и С равно S_2 км. Время движения велосипедиста между пунктами А и В равно t_1 час, время движения между пунктами В и С равно t_2 час.

2. В банк положена сумма денег в размере x рублей на срочный депозит на квартал из расчета 24 % годовых. Вычислить, какая сумма будет на счету вкладчика через три месяца.

3. Треугольник на плоскости задан координатами своих вершин $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Найти его периметр.

4. Найти сумму цифр задаваемого целого трехзначного числа x .

1.2. Варианты заданий для самостоятельной работы

1. Даны длины катетов прямоугольного треугольника. Вычислить его периметр.

2. По введенному радиусу R вычислить длину окружности, площадь круга и объем шара, площадь равностороннего треугольника, вписанного в этот круг.

3. Даны основания и высота равнобедренной трапеции. Найти ее периметр.

4. Заданы координаты вершин треугольника на плоскости. Найти его площадь.

5. Найти сумму членов арифметической прогрессии, если известны ее первый член, разность и число членов прогрессии.

6. Лодка плывет из одного пункта в другой и обратно. Скорость лодки v км/час, расстояние между пунктами S км, скорость течения реки u км/час известны. Вычислить общее время движения лодки.

7. Вычислить путь, пройденный лодкой из пункта А в пункт В и обратно, если известно, что в стоячей воде скорость ее движения V км/час, скорость течения реки U км/час, время движения по течению t_1 часов, против течения t_2 часов.

8. Найти координаты вершин параболы $y = ax^2 + bx + c$. Коэффициенты a , b , c вводятся пользователем.

9. Функция $y = \sin(x)$ на отрезке $[0, \pi/2]$ аппроксимируется (приближенно заменяется) выражением $x - x^3/6 + x^5/120$. Для заданного значе-

ния x вычислить y по этой формуле и сравнить с точным значением, полученным с помощью встроенной функции $\sin(x)$.

10. Правительство гарантирует, что инфляция в новом году составит p % в месяц. Какой рост цен за год можно ожидать?

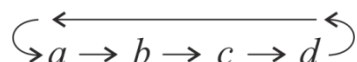
11. Коммерсант, имея стартовый капитал в k рублей, занялся торговлей, которая ежемесячно увеличивает его капитал на p %. Через сколько лет он накопит сумму S рублей, достаточную для покупки автомобиля стоимостью m рублей?

12. Заданы уравнения двух пересекающихся прямых на плоскости: $y = k_1x + b_1$; $y = k_2x + b_2$. Найти угол между ними в градусах и минутах.

13. Поезд метро едет с постоянной скоростью V км/час в течение времени t час, после чего начинает торможение, доводя поезд в течение времени t час до полной остановки. Вычислить путь S , пройденный поездом.

14. Вывести цифры заданного целого четырехзначного числа в обратном порядке.

15. Обменять значения вещественных переменных a, b, c, d в порядке, указанном стрелками:



16. Обменяйте значения переменных x, y, z, t так, чтобы они образовали порядок, обратный исходному.

17. За первый год производительность труда на заводе выросла на p_1 %, за второй и третий соответственно на p_2 % и p_3 %. Найти среднегодовой прирост продукции в процентах.

18. Даны два трехмерных вектора, заданных своими координатами: $A(x_a, y_a, z_a)$, $B(x_b, y_b, z_b)$. Найти угол в градусах между ними.

19. Известны координаты вершин треугольника. Вычислить длины его высот.

20. Найти корни квадратного уравнения $ax^2 + bx + c = 0$, если известно, что коэффициенты его отличны от нуля, а дискриминант положителен.

21. У квадрата ABCD на плоскости заданы координаты двух его противоположных вершин A и C. Найти координаты вершин B и D. Расположение квадрата произвольно, его стороны могут не быть параллельными осям координат.

22. Треугольник задан на плоскости координатами вершин $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Найти сумму длин его медиан.

23. В равнобедренном прямоугольном треугольнике известна длина высоты, опущенной на гипотенузу. Найти периметр этого треугольника.

24. Найти частное от деления произведений четных и нечетных цифр целого четырехзначного числа. Если в числе не содержится ни одной нечетной цифры, вывести соответствующее сообщение.

25. Турист разделил свой маршрут на три части разной протяженности. На первом участке пути он шел t_1 часов со скоростью V_1 км/час, на втором участке – t_2 часов со скоростью V_2 км/час. Последний отрезок пути он прошел за t_3 минут со скоростью V_3 км/час. Пройдя половину всего пути, он сделал привал. Сколько км турист прошел после привала?

26. Ввести три вещественных числа x, y, z – зарплату трех сотрудников. Найти их среднюю зарплату и отклонение каждой зарплаты от средней.

1.3. Решения

1. Средняя скорость V определяется по формуле $(S_1 + S_2) / (t_1 + t_2)$, значения переменных S_1, S_2, t_1, t_2 вводятся с клавиатуры.

```
#include <iostream.h>
using namespace std;
int main()
{
    double S1, S2, t1, t2, V;
    cout<<"Enter S1="; cin>>S1;
    cout<<"Enter S2="; cin>>S2;
    cout<<"Enter t1="; cin>>t1;
    cout<<"Enter t2="; cin>>t2;
    V=(S1+S2)/(t1+t2);
    cout << "Average speed = "<< V<< "\n";
    return 0;
    system ("pause");
}
```

3.

```
#include <iostream.h>
using namespace std;
int main()
{ double x1, y1, x2, y2, x3, y3, a, b, c;
  cout<<"Enter first point coordinates: x1= "; cin>>x1;
  cout<<"y1=";
  cin>>y1;
  cout<<"\nEnter second point coordinates: x2= "; cin>>x2;
  cout<<"y2=";
  cin>>y2;
```

```

    cout<<"\nEnter third point coordinates: x3= "; cin>>x3;
    cout<<"y3=";
    cin>>y3;
    a = sqrt(pow(x1-x2,2)+pow(y1-y2,2));
    b = sqrt(pow(x1-x3,2)+pow(y1-y3,2));
    c = sqrt(pow(x2-x3,2)+pow(y2-y3,2));
    cout << "Perimeter = " << a+b+c << endl;
    return 0;
    system ("pause");
}

```

Замечание. Операция возведения в степень реализована в системах программирования на языке C++ в виде библиотечной функции `pow` (сокращение от *power* (англ.) – степень). Ее вызов приводит к затратам времени, превышающим время на выполнение обычной операции умножения. Поэтому для возведения какого-либо числового значения в целую степень разумнее пользоваться операцией `*` (умножение):

```

a = (x1-x2)*(x1-x2)+(y1-y2)*(y1-y2);
b = (x1-x3)*(x1-x3)+(y1-y3)*(y1-y3);
c = (x2-x3)*(x2-x3)+(y2-y3)*(y2-y3);

```

4. $S = x\%10 + x/10\%10 + x/100;$

1.4. Указания к решению

6. Время, в течение которого лодка плывет по течению: $t_1 = S / (V + U)$; время, в течение которого лодка плывет против течения: $t_2 = S / (V - U)$. Общее время: $t_1 + t_2$.

10. За каждый месяц цены возрастут в $1 + p/100$ раз, за год рост составит $(1 + p/100)^{12}$ раз или в процентах: $S = [(1 + p/100)^{12} - 1] \cdot 100$.

12. Используйте формулу $\operatorname{tg} \varphi = (k_2 - k_1) / (1 + k_1 \cdot k_2)$.

13. $S = S_1 + S_2$, где S_1 – путь, пройденный поездом при равномерном движении, S_2 – путь, пройденный поездом при равнозамедленном движении.

$$S_1 = V \cdot t, \quad S_2 = \frac{at^2}{2},$$

где $a = V/t$ – ускорение при равнозамедленном движении. Сравните полученный результат с результатом, вычисленным по формуле $S = 3Vt/2$.

17. Общий прирост за три года определится по формуле

$$p = [(1 + p_1/100)(1 + p_2/100)(1 + p_3/100) - 1] \cdot 100.$$

Тот же результат можно получить при среднегодовом приросте r :

$$p = [(1 + r / 100)^3 - 1] \cdot 100.$$

Остается только найти величину r .

19. Используйте формулу

$$\cos \varphi = \frac{(A, B)}{(|A| \cdot |B|)} = \frac{x_a x_b + y_a y_b + z_a z_b}{\sqrt{(x_a^2 + y_a^2 + z_a^2)} \sqrt{(x_b^2 + y_b^2 + z_b^2)}}.$$

Глава 2. АЛГОРИТМЫ С ВЕТВЛЕНИЯМИ

2.1. Задачи с решениями

1. Определить по введенным пользователем координатам попадания выстрела в мишень количество набранных очков (рис. 2.1).

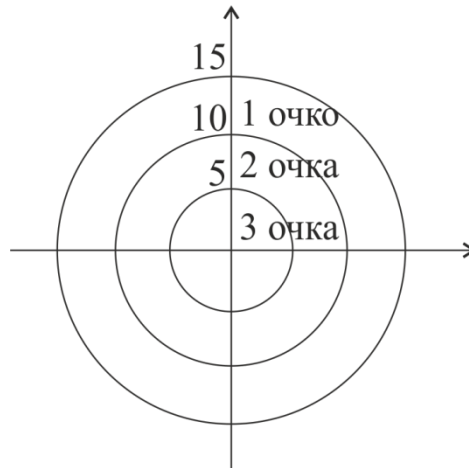


Рис. 2.1

2. Найти максимальное из двух заданных вещественных чисел a и b , вводимых пользователем.

3. Найти максимальное из трех заданных вещественных чисел a , b и c , вводимых пользователем.

4. Вычислить значение функции y в задаваемой пользователем точке x :

$$y = \begin{cases} a - \cos(2x - \pi / 12) + 1, & \text{если } x < \pi / 4; \\ 2(a - \cos(2x - \pi / 12))^2, & \text{если } \pi / 4 \leq x < \pi / 2; \\ \frac{1}{a - \cos(2x - \pi / 12)} + 3, & \text{если } \pi / 2 \leq x < \pi; \\ 3(a - \cos(2x - \pi / 12) + 7), & \text{если } x \geq \pi. \end{cases}$$

5. По приведенному ниже фрагменту программы сформулировать условие задачи:

```
float a, b, c, m;  
cin >> a >> b >> c;  
if (a < b)  
{  
    if (a < c) m = a;  
    else m = c;  
}  
else  
{
```



```

    if (b<c) m = b;
    else m = c;
}
cout << m;

```

6. Известно, что одна из трех монет имеет достоинство 5 копеек, другая 10 копеек, третья 50 копеек. Определить, можно ли набрать заданную пользователем сумму, используя только две монеты из трех.

7. Ввести три вещественных числа a , b , c . Определить, сколько из них принадлежит заданному интервалу (x, y) .

2.2. Варианты заданий для самостоятельной работы

2.2.1. Задание 1

1. С клавиатуры вводится целое число из диапазона от 0 до 10. Проверить, является ли оно нулем или числом Фибоначчи. (Числами Фибоначчи называются натуральные числа, вычисляемые по рекуррентной формуле $F_{i+1} = F_i + F_{i-1}$, $F_0 = F_1 = 1$.)

2. В каждый подарочный набор входят 1 карандаш, 2 линейки, 3 ручки, 4 тетради. Имеется a карандашей, b тетрадей, c ручек и d линеек. Сколько подарочных наборов можно получить из этих предметов?

3. С клавиатуры вводятся две даты. Вывести сообщение «Правильно», если первая введенная дата предшествует второй, «Неправильно», если вторая дата предшествует первой, и «Равны», если введены одинаковые даты.

4. Требуется ввести натуральное число $1 \leq n \leq 120$, задающее возраст человека в годах, и вывести его с соответствующей подписью «лет» или «год» или «года». (Например, «21 год» или «45 лет» или «33 года».)

5. Дано четырехзначное целое число. Выяснить, является ли оно перевертышем.

6. Требуется ввести целое число $1 \leq k \leq 7$, являющееся номером дня недели, и вывести название соответствующего дня недели.

7. Требуется ввести два числа и один из знаков арифметической операции $+$, $-$, $*$, $/$, $\%$. Вывести результат соответствующего действия.

8. Даны две точки на плоскости $A(x_1, y_1)$ и $B(x_2, y_2)$. Определить, какая из них находится ближе к началу координат.

9. Дано целое трехзначное число. Определить его наибольшую цифру.

10. Дано целое двухзначное число. Определить, равен ли квадрат этого числа учетверенной сумме кубов его цифр.

11. Дано четырехзначное целое число. Определить, равны ли суммы его крайних и средних цифр.

12. Составить программу для определения весовой категории спортсмена в зависимости от его веса

$$\text{категория} = \begin{cases} \text{легкая,} & \text{если вес} < 62 \text{ кг;} \\ \text{средняя,} & \text{если } 62 \leq \text{вес} \leq 75 \text{ кг;} \\ \text{тяжелая,} & \text{если } 75 < \text{вес} \leq 88 \text{ кг;} \\ \text{сверхтяжелая,} & \text{если вес} \geq 88 \text{ кг.} \end{cases}$$

13. Составить программу вычисления корней квадратного уравнения $ax^2 + bx + c = 0$.

14. Даны три числа x, y, z . Если они могут быть длинами сторон треугольника, определить, является ли он прямоугольным.

15. Даны два целых трехразрядных числа a, b . Если какая-либо из цифр числа a равна какой-либо цифре числа b , заменить меньшее из введенных чисел a, b на сумму цифр большего числа.

16. Даны три целых трехразрядных числа a, b, c . Если среди их младших цифр нет ни одного нуля, переменной a присвоить значение переменной b , переменной b – значение переменной c , переменной c – первоначальное значение переменной a .

17. В зависимости от задаваемого номера месяца (от 1 до 12) вывести название соответствующего времени года.

18. Требуется каждое из трех заданных положительных чисел увеличить на сумму цифр минимального.

19. Даны три вещественных числа x, y, z . Вычислить значение переменной U по формуле

$$U = \frac{\min((x + y + z) / 3, x \cdot y \cdot z) + 1}{1 - \min^2(x \cdot y \cdot z, (x + y + z) / 3)}.$$

20. По заданному значению x вычислить значение функции y , заданной графически (рис. 2.2).

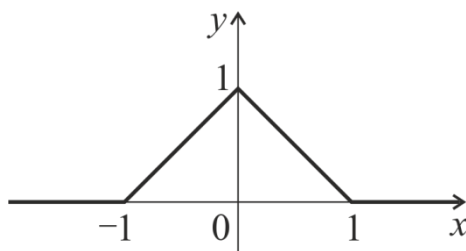


Рис. 2.2

21. По нажатой клавише выдать сообщение о назначении этой клавиши.

22. Даны координаты вершин четырехугольника на плоскости $A(x_1, y_1), B(x_2, y_2), C(x_3, y_3), D(x_4, y_4)$. Если этот четырехугольник является прямоугольником, стороны которого параллельны осям координат, вычислить его площадь, в противном случае вычислить его периметр.

23. Из одного поселка в другой со скоростью V_1 км/час выехал велосипедист. Через t_1 часов вслед за ним выехал мотоциклист со скоростью

V_2 м/мин. Определить, догонит ли мотоциклист велосипедиста через t часов после своего выезда.

24. Вводятся три вещественных числа. Требуется вывести их в том же порядке, заменив минимальное на полусумму двух других чисел.

25. Даны три целых положительных трехзначных числа. Требуется каждое число, отличное от максимального, увеличить на сумму цифр минимального.

26. Даны три целых положительных трехзначных числа. Вывести то число, сумма цифр которого максимальна.

27. Составить программу для проверки, пройдет ли кирпич с ребрами a , b , c в прямоугольное отверстие со сторонами x и y . Кирпич располагается так, что его ребра параллельны или перпендикулярны каждой из сторон отверстия.

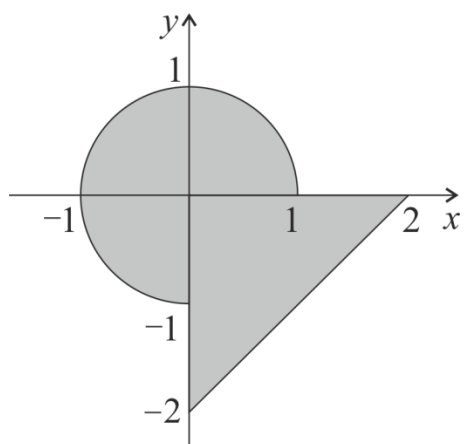
28. Найти полярные координаты r и φ точки на плоскости по ее декартовым координатам.

29. Факультету выделен стипендиальный фонд в размере F рублей. Известны результаты последней экзаменационной сессии: n_1 – число отличников ($n_1 > 0$), n_2 – число хорошистов, n_3 – число студентов, имеющих хотя бы одну оценку «удовлетворительно» и не имеющих ни одной неудовлетворительной оценки, n_4 – число задолженников. Каков размер стипендии у студентов каждой категории и каков остаток стипендиального фонда, идущего на материальную помощь, если стипендия хорошистов вычисляется по формуле $S_2 = S_3 + 0,5 \cdot S_3$, стипендия отличников вычисляется по формуле $S_1 = S_2 + 0,5 \cdot S_2$, где S_3 – стипендия троечников, размер которой вводится.

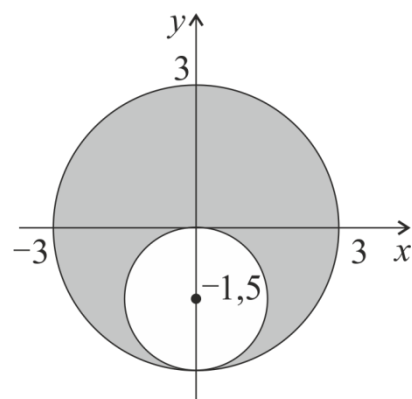
2.2.2. Задание 2

Составить программу для вычисления значения площади заштрихованной области, если точка с заданными координатами (x, y) попадает в эту область. В противном случае вывести соответствующее сообщение.

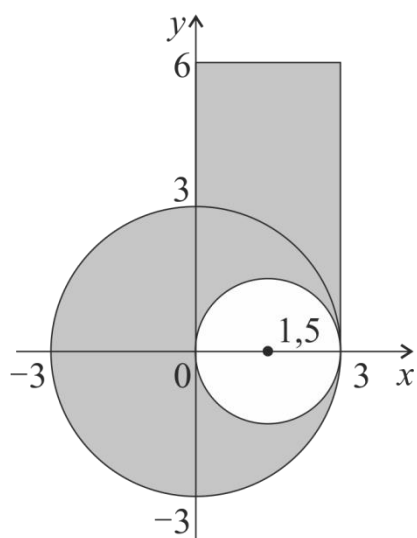
1.



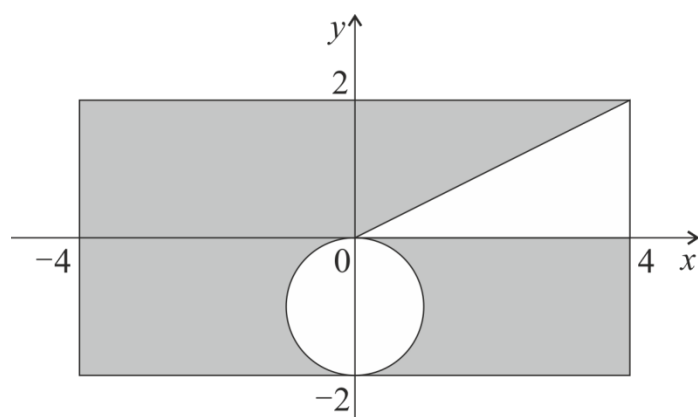
2.



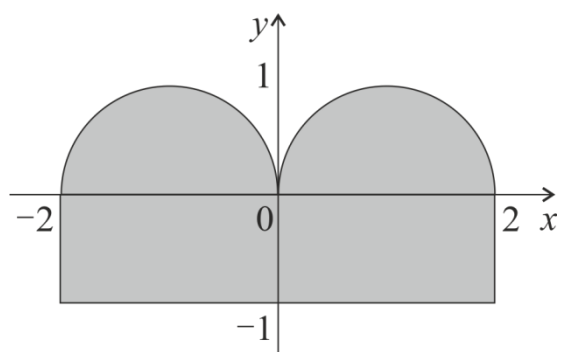
3.



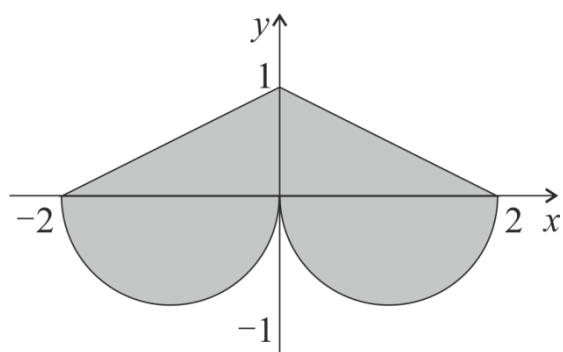
4.



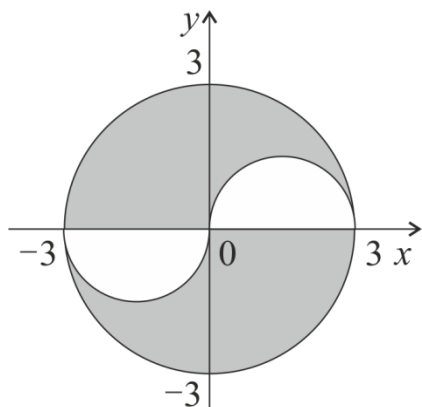
5.



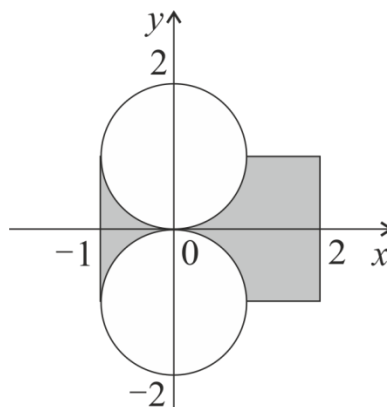
6.



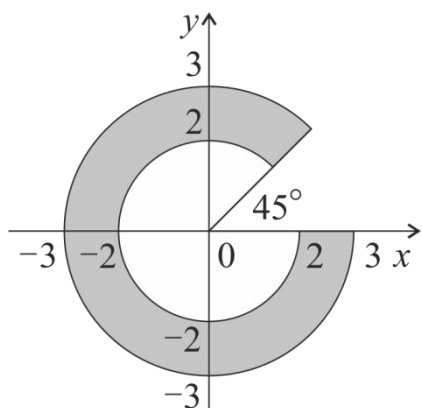
7.



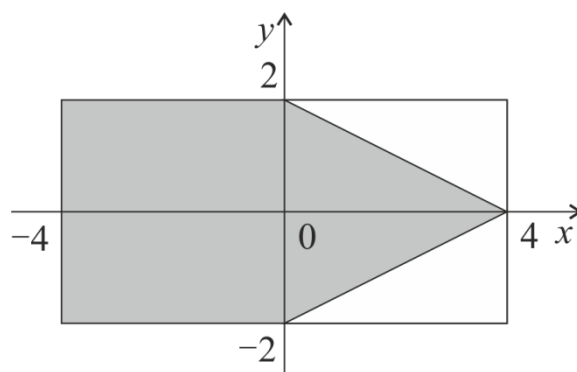
8.



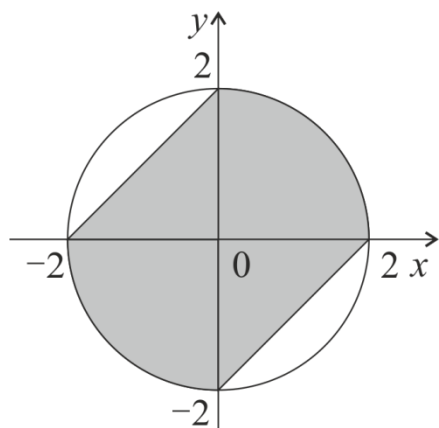
9.



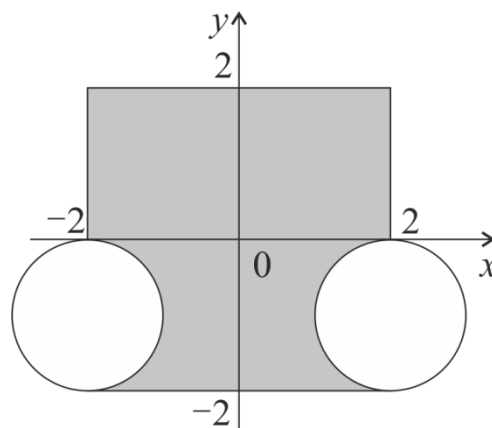
10.



11.



12.



2.2.3. Задание 3

1. Введите три числа. Если они могут быть длинами сторон прямоугольного треугольника, выведите их в порядке возрастания и вычислите площадь полученного треугольника.

2. Введите три числа. Если они могут быть длинами сторон остроугольного треугольника, выведите их в порядке убывания и вычислите площадь полученного треугольника.

3. Введите три числа. Если они могут быть длинами сторон тупоугольного треугольника, выведите их в порядке убывания и вычислите площадь полученного треугольника.

4. Введите три числа. Если они могут быть сторонами равностороннего треугольника, вычислите его площадь и длину высоты. Выведите длины сторон треугольника, площадь и длину высоты в порядке возрастания.

5. Введите три числа. Если они могут быть длинами сторон равнобедренного треугольника, вычислите длины его высот. Выведите длину основания и длины высот в порядке возрастания.

6. Введите три числа. Если они могут быть длинами сторон разностороннего тупоугольного треугольника, выведите их в порядке возрастания и вычислите площадь полученного треугольника.

7. Введите три числа. Если они могут быть длинами сторон равнобедренного тупоугольного треугольника, вычислите его площадь. Выведите длины сторон и площадь в порядке возрастания значений.

8. Введите три числа. Если они могут быть длинами сторон равнобедренного остроугольного треугольника, вычислите его площадь. Выведите длины сторон и площадь в порядке возрастания значений.

9. Введите три числа. Если они могут быть длинами сторон разностороннего остроугольного треугольника, выведите их в порядке возрастания и вычислите площадь полученного треугольника.

10. Пусть даны координаты трех точек на плоскости. Если они могут быть вершинами прямоугольного треугольника, вычислите его площадь.

12. Пусть даны координаты трех точек на плоскости. Если они могут быть вершинами тупоугольного треугольника, вычислите его площадь. Выведите длины сторон в порядке убывания их значений.

13. Пусть даны координаты трех точек на плоскости, Если они могут быть вершинами равностороннего треугольника, вычислите его площадь и длину высоты. Выведите длины сторон, площадь и длину высоты в порядке возрастания их значений.

14. Пусть даны координаты трех точек на плоскости. Если они могут быть вершинами равнобедренного треугольника, вычислите длины его высот. Выведите длины основания и высот в порядке возрастания их значений.

15. Пусть даны координаты трех точек на плоскости. Если они могут быть вершинами разностороннего тупоугольного треугольника, вычислите его площадь.

16. Пусть даны координаты трех точек на плоскости, Если они могут быть вершинами равнобедренного тупоугольного треугольника, вычислите

его площадь. Выведите длины сторон и площадь в порядке возрастания их значений.

17. Пусть даны координаты трех точек на плоскости. Если они могут быть вершинами равнобедренного остроугольного треугольника, вычислите его площадь. Выведите длины сторон и площадь в порядке возрастания их значений.

18. Пусть даны координаты трех точек на плоскости. Если они могут быть вершинами разностороннего остроугольного треугольника, вычислите его площадь.

19. Пусть даны три числа. Если они могут быть длинами сторон треугольника, определите его вид (разносторонний, равнобедренный, равносторонний). Вычислите длины его высот и выведите их в порядке убывания.

20. Пусть даны три числа. Если они могут быть длинами сторон треугольника, определите его вид (прямоугольный, тупоугольный, остроугольный). Выведите длины высот треугольника в порядке убывания.

21. Пусть даны координаты трех точек на плоскости. Если они могут быть вершинами треугольника, определите его вид (разносторонний, равнобедренный, равносторонний). Вычислите длины его высот и выведите их в порядке убывания.

22. Пусть даны координаты трех точек на плоскости. Если они могут быть вершинами треугольника, определите его вид (прямоугольный, тупоугольный, остроугольный). Вычислите длины его высот и выведите их в порядке возрастания.

23. Составьте программу для определения вида треугольника (равносторонний, равнобедренный, разносторонний, прямоугольный, тупоугольный, остроугольный), если по данным трем отрезкам его можно построить.

24. Пусть даны координаты трех точек на плоскости. Составьте программу для определения вида треугольника (равносторонний, равнобедренный, разносторонний, прямоугольный, тупоугольный, остроугольный), если данные координаты вершин позволяют его построить.

25. Пусть даны координаты вершин четырехугольника. Составьте программу для определения, является ли четырехугольник прямоугольником.

26. Пусть даны координаты трех вершин прямоугольника. Определите координаты четвертой вершины.

2.3. Решения

1.

```
#include <iostream.h>
using namespace std;
int main ()
{
```

```

float R, x, y;
int kol;
cout << "Input coordinates\n";
cin >> x >> y;
R = x*x+y*y; kol = 0;
if (R<1) kol = 3;
else
    if (R<4) kol = 2;
    else
        if (R<9) kol = 1;
cout << kol;
return 0;
system ("pause");
}

```

2. Существует несколько вариантов решения задачи.

Первый способ

```

#include <iostream.h>
using namespace std;
int main ()
{
float a, b, max;
cin >> a >> b;
if (a>b) max = a;
else max = b;
cout << max;
return 0;
system ("pause");
}

```

Второй способ

```

#include <iostream.h>
using namespace std;
int main ()
{
float a, b, max;
cin >> a >> b;
max = (b>a)? b : a;
cout << max;
return 0;
system ("pause");
}

```


3. Первый способ

```
#include <iostream.h>
using namespace std;
int main ()
{
    float a, b, c, max;
    cin >> a >> b >> c;
    if (a>b) max = a;
    else max = b;
    if (c>max) max = c;
    cout << max;
    return 0;
    system ("pause");
}
```

Второй способ

```
#include <iostream.h>
using namespace std;
int main ()
{ float a, b, c, max;
  cin >> a >> b >> c;
  if (a>b && a>c) cout << a;
  else
    if (b>a && b>c) cout << b;
    else
      if (c>a && c>b) cout << c;
  return 0;
  system ("pause");
}
```

4.

```
#include <iostream.h>
using namespace std;
int main ()
{
    float a, x, y, z;
    const float pi = 3.14159;
    z = a - cos(2*x - pi/12);
    if (x < pi/4) y = z+1;
    else
        if (x < pi/2) y = 2*z*z;
    else
        if (x < pi) y = 1/z+3;
        else y = 3*z+7;
}
```

```

cout << y;
return 0;
system ("pause");
}

```

5. Вывести значение максимального из трех вещественных чисел a , b , c .

6. *Первый* вариант решения с использованием оператора if

```

#include <iostream.h>
using namespace std;
int main ()
{ int n;
  cin >> n;
  if (10+5 == n) cout <<"Yes";
  else
    if (5+50 == n) cout <<"Yes";
    else
      if (10+50 == n) cout <<"Yes";
      else cout << "No";
  return 0;
  system ("pause");
}

```

Второй вариант решения с использованием оператора выбора switch

```

#include <iostream.h>
using namespace std;
int main ()
{int n;
  cin >> n;
  switch (n)
  { case 10+5: cout << "Yes"; break;
    case 50+5: cout << "Yes"; break;
    case 50+10: cout << "Yes"; break;
    default: {cout << "No"; break};
  }
  return 0;
  system("pause");
}

```

7.

```

#include <iostream.h>
using namespace std;
int main ()
{

```

```

float a, b, c, x, y;
cin >> a >> b >> c;
cin >> x >> y;
int q = 0;
if (a>x && a<y) {cout << a; q += 1 ;}
if (b>x && b<y) {cout << b; q++;}
if (c>x && c<y) {cout << c; q += 1;}
switch (q)
{case 0: cout << "No number is in interval\n"; break;
  case 1: cout << "One number is in interval\n"; break;
  case 2: cout << "Two numbers are in interval\n"; break;
  case 3: cout << "All numbers are in interval\n"; break;
}
return 0;
system("pause");
}

```

2.4. Указания к решению

4. Использовать операцию нахождения остатка от целочисленного деления.

5. Перевертыш – число, запись которого одинаково читается как слева направо, так и справа налево. В общем случае этот термин (а также «палиндром») применяется к любой строке символов, одинаково читающейся при прямом и обратном просмотре.

10. Например, для числа 48 ответ будет положительным, для числа 52 – отрицательным.

21.

<i>Назначение клавиши</i>	<i>Код клавиши</i>
Стрелка вправо	77
Стрелка влево	75
Стрелка вверх	72
Стрелка вниз	80

28. Воспользуйтесь формулами

$$r = \sqrt{x^2 + y^2}$$

$$\varphi = \begin{cases} \arctg y / x, & \text{если } x > 0, y > 0; \\ \pi / 2, & \text{если } x = 0, y > 0; \\ \pi + \arctg y / x, & \text{если } x < 0; \\ 3\pi / 2, & \text{если } x = 0, y < 0; \\ 2\pi + \arctg y / x, & \text{если } x > 0, y < 0. \end{cases}$$

Глава 3. ЦИКЛИЧЕСКИЕ АЛГОРИТМЫ

3.1. Задачи с решениями

1. По заданному целому положительному числу n вычислить его факториал $n!$

2. Какая задача решается с помощью следующего фрагмента программы

```
float xN, xk, dx;  
cin >> xN >> xk >> dx;  
float x;  
x = xN;  
while (x<=xk)  
{ cout << x*x+1;  
  x+=dx;  
}
```

3. Вычислить значение произведения для заданного числа N :

$$\prod_{k=2}^N \left[1 - \frac{2}{k(k+1)} \right].$$

4. Для заданных чисел a и p вычислить $x = \sqrt[p]{a}$ по рекуррентному соотношению (формула Ньютона)

$$x_{n+1} = \frac{1}{p} [(p-1)x_n + a / x_n^{p-1}]; \quad x_0 = a.$$

Сколько итераций надо выполнить, чтобы для заданной погрешности E выполнялось соотношение $|x_{n+1} - x_n| \leq E$?

5. Составить программу, моделирующую работу простого калькулятора. Пользователь поочередно вводит числа и знаки операций (+, -, *, /) над этими числами. Операции выполняются последовательно до тех пор, пока не будет введен знак равенства.

6. Сто вещественных чисел вводятся по одному. Найти разность между максимальным и минимальным из них.

7. Сто вещественных чисел вводятся по одному. Найти два наибольших среди введенных чисел.

3.2. Варианты заданий для самостоятельной работы

3.2.1. Задание 1

1. Начав тренировки, спортсмен в первый день пробежал 5 км. Каждый следующий день он увеличивал пробег на 10 % от пробега предыдущего дня. Определить

- 1) Пробег спортсмена за каждый из первых 10 дней тренировки.
 - 2) Суммарный пробег за первую неделю тренировки.
 - 3) В какой день он пробежал более 20 км, если спортсмен начал тренировки в среду, 3 июня.
 - 4) В какой день суммарный пробег станет больше 70 км, если первый день тренировок был в четверг, 20 апреля.
 2. На отрезке $[2, n]$ найти натуральное число с максимальной суммой делителей.
 3. По одной оценке вводятся результаты контрольной работы по информатике 25 студентов группы. Определить:
 - а) – число студентов, получивших отличную оценку,
 - число студентов, получивших оценку хорошо,
 - число студентов, получивших удовлетворительную оценку,
 - число студентов, получивших неудовлетворительную оценку;
 - б) среднюю оценку группы;
 - в) какую оценку получило наибольшее число студентов группы.
 4. Предприниматель, начав дело, взял кредит в размере k рублей под p процентов годовых. По прогнозу его дело должно давать прибыль r рублей в год. Сможет ли он накопить сумму, достаточную для погашения кредита и, если да, то через сколько лет?
 5. Для каждого посетителя парикмахерской (с одним мастером) известны следующие величины: t – момент его прихода (в часах и минутах) и t_1 – продолжительность его обслуживания (в минутах). Сколько клиентов обслужит мастер за смену продолжительностью T часов? Сколько рабочего времени он потратит на обслуживание?
 6. Дано натуральное число. Получить из него новое число, заменив в исходном все цифры 8 на 3 и удалив все единицы.
 7. Дано натуральное число $k < 99$. Получить все способы выплаты суммы денег в размере k копеек с помощью монет достоинством в 1, 5, 10 и 50 копеек.
 8. Составить программу нахождения корня уравнения $x + \sqrt{x} + \sqrt[3]{x} - 2,5 = 0$ методом деления отрезка $[0.4, 1]$ пополам с заданной точностью $E=10^{-5}$.
 9. По заданному натуральному числу получить новое число, поменяв в исходном порядок следования цифр на противоположный.
 10. В 2011 году на участке в 100 гектаров средняя урожайность ячменя составила 20 центнеров с гектара. После этого каждый год площадь участка увеличивалась на 5 %, а средняя урожайность – на 2 %.
- Определить
- а) урожайность за 2012, 2013, ..., 2018 год;
 - б) площадь участка в 2014, 2015, 2016, 2017 годах;

- в) общий урожай за шесть лет: с 2011 по 2016 годы;
 г) в каком году урожайность превысит 22 центнера с гектара;
 д) в каком году площадь участка станет больше 120 гектаров;
 е) в каком году общий урожай, собранный за все время, начиная с 2012 года, превысит 800 центнеров.

11. Дано натуральное число n . Вычислить значение выражения:

а) $\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \sin 2 + \dots + \sin n}$;

б) $\frac{\cos 1}{\sin 1} + \frac{\cos 1 + \cos 2}{\sin 1 + \sin 2} + \dots + \frac{\cos 1 + \dots + \cos n}{\sin 1 + \dots + \sin n}$.

12. Гражданин Х открыл 1 марта счет в банке, вложив 10 000 рублей. Каждый месяц размер вклада увеличивается на 2 % от имеющейся суммы.

Определить:

- а) прирост суммы за первый, второй, ..., десятый месяц;
 б) сумму вклада через три, четыре, ..., двенадцать месяцев;
 в) за какой по счету месяц величина ежемесячного увеличения вклада превысит 300 руб.;

г) через сколько месяцев размер вклада превысит 1200 руб.

13. Даны три натуральных числа x, y, z . Вывести делители большего из них.

14. Найти совершенные числа, меньшие заданного N .

15. Для заданного вещественного x вычислить значение многочлена

$$4x^5 - 2x^4 + 3x^3 + 5x^2 - 2x + 3,$$

используя схему Горнера.

16. Для заданных a и p вычислить $x = \sqrt[p]{a}$, используя рекуррентную формулу

$$x_{n+1} = \frac{x_n}{p^2} \left[(p^2 - 1) + \frac{1}{2}(p + 1) \frac{a}{x_n^p} - \frac{1}{2}(p - 1) \frac{x_n^p}{a} \right].$$

Сколько итераций надо выполнить, чтобы для заданной погрешности E было справедливо соотношение

$$|x_{n+1} - x_n| \leq E?$$

При каких начальных приближениях x_0 процесс сходится?

17. Вычислить $x = \sqrt[3]{a}$ для заданного значения a , используя рекуррентную формулу

$$x_{n+1} = \frac{1}{3} \left(x_n + 2\sqrt[3]{\frac{a}{x_n}} \right).$$

Сколько итераций надо выполнить для достижения заданной точности E , используя условие

$$|x_{n+1} - x_n| \leq E?$$

18. Для заданного $x > 1$ вычислить $y = \sqrt{x}$ по итерационной формуле

$$y_{n+1} = \frac{1}{2} \left(y_n + \frac{x}{y_n} \right)$$

с заданной погрешностью E , задав начальное приближение $y_0 = x$. Сравнить с результатом вызова встроенной функции вычисления квадратного корня. Сколько итераций пришлось выполнить?

19. Из пунктов А и В, удаленных друг от друга на расстояние $a = 600$ км, навстречу друг другу одновременно отправляются два поезда; скорость первого $V_1 = 40$ км/ч, скорость второго $V_2 = 60$ км/ч. В то же время из пункта А вылетает сверхбыстрая муха со скоростью $V = 200$ км/ч и летит навстречу поезду, следующему из пункта В; встретившись с ним, она летит к поезду, следующему из пункта А и т. д. до тех пор, пока поезда не встретятся. Определить общее расстояние, которое пролетит муха.

20. Способ последовательных приближений позволяет находить корни пятой степени из положительного числа a приближенно по формуле

$$x_{n+1} = \frac{4}{5}x_n + \frac{a}{5x_n^4}.$$

При этом разность между x_{n+1} и значением корня по абсолютной величине не превосходит значения $5/4a \cdot |x_{n+1} - x_n|$. Составить программу вычисления корня пятой степени из заданного числа a с точностью до 10^{-k} (k также задается вводом), принимая в качестве начального приближения:

$$x_0 = \begin{cases} \min(2a, 0.95), & \text{при } a \leq 1; \\ \frac{a}{5}, & \text{при } 1 < a \leq 25; \\ \frac{a}{25}, & \text{при } a > 25. \end{cases}$$

21. Составить программу вычисления значения k по формуле

$$k = \begin{cases} \frac{n(n-2)(n-1)}{6}, & \text{если } n \leq 9; \\ 2^{29-n}, & \text{если } 9 < n \leq 29; \\ 1, & \text{если } n = 29; \\ (n-9)!, & \text{если } n > 29. \end{cases}$$

где n – натуральное число, задаваемое вводом.

22. Составить программу вычисления при заданных p , x и a значения функции y вида:

$$\text{а) } y = (((((((((x+a)^2 + a)^2 + a)^2 + a)^2 + a)^2 + a)^2 + a)^2 + a)^2;$$

$$\text{б) } y = a + (a + (a + (a + (a + (a + (a + (a + (a + x)^9)^9)^9)^9)^9)^9;$$

$$\text{в) } y = \frac{(x-2)(x-4) \dots (x-64)}{(x-1)(x-3) \dots (x-65)};$$

$$\text{г) } y = (((((x-p)x-p)x-p)x-p)x-p)x-p).$$

23. Вычислите:

$$\text{а) } y = \sqrt{3 + \sqrt{6 + \sqrt{9 + \dots + \sqrt{96 + \sqrt{99}}}}};$$

$$\text{б) } y = \frac{1}{1 + \frac{1}{3 + \frac{1}{5 + \dots \frac{1}{109 + \frac{1}{111}}}}};$$

24. Напишите программу вычисления при заданном x величины y по формуле:

$$y = \frac{x}{x^2 + \frac{2}{x^2 + \frac{4}{x^2 + \frac{8}{x^2 + \dots \frac{256}{x^2}}}}}.$$

25. Вычислите значения следующих выражений, считая заданным значение x :

$$\text{а) } y = \cos x + \cos x^2 + \cos x^3 + \dots + \cos x^{30};$$

$$\text{б) } y = 1! + 2! + 3! + \dots + n!, \quad n > 1;$$

$$\text{в) } y = \sin 1 + \sin 1.1 + \sin 1.2 + \dots + \sin 2;$$

д) найдите первое из чисел $\sin x$, $\sin \sin x$, $\sin \sin \sin x$, ..., меньшее по модулю, чем 10^{-4} .

26. Вычислите сумму квадратов всех целых чисел, попадающих в интервал $(\ln x, e^x)$, $x > 1$.

27. Напишите программу нахождения первого члена последовательности

$$a_n = (-1)^n \left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} \right), \quad n = 1, 2, \dots,$$

который не принадлежит заданному отрезку $[a, b]$.

28. Пусть дан прямоугольник, длины сторон которого a и b – натуральные числа. На сколько квадратов, стороны которых выражены нату-

ральными числами, можно разрезать данный прямоугольник, если от него каждый раз отрезать квадрат максимально возможной площади?

29. В соревнованиях по фигурному катанию оценки выставляют несколько судей. При выведении единой оценки за выступление одного спортсмена из всей совокупности оценок удаляется наиболее высокая и наиболее низкая, и для оставшихся оценок вычисляется среднее арифметическое. Если наиболее низкую или наиболее высокую оценки выставили несколько судей, то из совокупности оценок удаляется только одна такая оценка. Напишите программу для вычисления оценки спортсмена, если оценки вводятся для каждого спортсмена последовательно по одной, и известно количество судей.

30. Пусть даны натуральное число n и вещественное число x . Вычислите:

- a) $\left(\frac{1}{1!} + \sqrt{|x|}\right) + \left(\frac{1}{2!} + \sqrt{|x|}\right) + \dots + \left(\frac{1}{n!} + \sqrt{|x|}\right);$
- b) $\left(1 + \frac{\sin x}{1!}\right) + \left(1 + \frac{\sin 2x}{2!}\right) + \dots + \left(1 + \frac{\sin nx}{n!}\right);$
- c) $\left(\frac{1}{2} + \cos |x|\right) + \left(\frac{2}{3} + \cos^2 |x|\right) + \dots + \left(\frac{n}{n+1} + \cos^n |x|\right);$
- e) $\sin x + \sin^2 x + \dots + \sin^n x;$
- f) $\sin x + \sin x^2 + \dots + \sin x^n;$
- g) $\sin x + \sin \sin x + \dots + \sin \sin \sin \dots \sin x.$

31. Пусть даны натуральное число n и вещественные числа $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Рассматривая пары x_i, y_i как координаты точек на плоскости, определите радиус наименьшего круга с центром в начале координат, внутрь которого попадают все эти точки.

32. Даны натуральное число n ($n > 2$) вещественные числа $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Рассматривая пары x_i, y_i как координаты левых и правых концов отрезков числовой прямой, определите концы отрезка, являющегося пересечением всех этих отрезков.

33. Пусть интервал (a, b) разбит точками на n равных частей; в каждой точке вычисляется значение функции

$$y = \frac{x^2 - 3x + 2}{\sqrt{2x^3 - 1}}.$$

Найдите точку, в которой функция достигает наибольшего значения.

34. Пусть отрезок $[a, b]$ разбит точками на n равных частей. В каждой точке вычисляется значение функции

$$y = \frac{x^2 - 3x + 2}{\sqrt{2x^3 - 1}}.$$

Найдите наибольшее и наименьшее значения функции на этом отрезке.

35. Пусть отрезок $[a, b]$ разбит точками на n равных частей. В каждой точке вычисляется значение функции

$$y = \frac{x^2 - 3x + 2}{\sqrt{2x^3 - 1}}.$$

Найдите точки, в которых функция принимает наибольшее и наименьшее значения.

36. Пусть отрезок $[a, b]$ разбит точками на n равных частей. В каждой точке вычисляется значение функции

$$y = \frac{x^2 - 3x + 2}{\sqrt{2x^3 - 1}}.$$

Найдите среднее арифметическое положительных значений функции на отрезке $[a, b]$.

37. Вычислите приближенное значение $\int x^2 dx$ на отрезке $[a, b]$, используя формулу прямоугольников, если известно, что отрезок $[a, b]$ разбит на n частей.

38. Пусть даны целые числа x_1, x_2, \dots, x_{55} . Вычислите величину

$$x_1(x_2 + x_3)(x_4 + x_5 + x_6) \dots (x_{46} + x_{47} + \dots + x_{55}).$$

39. Пусть $x_1 = 0,3$; $x_2 = -0,3$; $x_i = i + \sin(x_{i-1})$, $i = 3, 4, \dots, 100$. Среди чисел x_1, x_2, \dots, x_{100} найдите ближайшее к какому-нибудь целому числу.

40. Пусть $a_i = \frac{(i-1)}{(i+1)} + \sin \frac{(i-1)^2}{(i+1)}$, $i = 1, 2, \dots, n$, где n задано. Найдите сумму всех положительных чисел a_i .

41. Пусть дано натуральное число n и вещественное число x . Среди чисел:

$$e^{\cos(kx)} \sin(x^3 / k), \text{ где } k = 1, 2, \dots, n,$$

найдите ближайшее к какому-нибудь целому числу.

42. Последовательность чисел a_1, a_2, \dots, a_{100} задана формулой

$$a_i = \sin^2(3i + 5) - \cos(i^2 - 15), i = 1, 2, \dots, 100.$$

Определите, сколько членов последовательности с номерами 1, 2, 4, 8, 16, ... имеют значение, меньшее 0.25.

43. Дано вещественное положительное число b . Последовательность a_1, a_2, a_3, \dots образована по правилу:

$$a_1 = b, a_i = a_{i-1} - \frac{1}{\sqrt{i}}, i = 2, 3, \dots$$

Найдите первый отрицательный член последовательности.

44. Дано вещественное отрицательное число b . Последовательность a_1, a_2, a_3, \dots образована по правилу:

$$a_1 = b, a_i = a_{i-1} - \frac{1}{\sqrt{i}}, i = 2, 3, \dots$$

Найдите первый неотрицательный член последовательности.

3.2.2. Задание 2

При некоторых заданных x , N и E , определяемых вводом, вычислите

а) сумму N слагаемых заданного вида;

б) сумму тех слагаемых, которые по абсолютной величине больше E .

Вычисление второй суммы выполните для двух значений E , отличающихся на порядок, при этом определите количество слагаемых, включенных в сумму, вычисляемую для каждого значения E . Сравните результаты со значением функции, для которой данная сумма определяет приближенное значение при x , лежащем в интервале $(-R, R)$, вычисленным с помощью встроженных функций компилятора.

$$1. \frac{\sin(x)}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \dots \quad (R = \infty).$$

$$2. e^{-x^2} = 1 - \frac{x^2}{1!} + \frac{x^4}{2!} - \frac{x^6}{3!} + \dots \quad (R = \infty).$$

$$3. \ln\left(x + \sqrt{x^2 + 1}\right) = x - \frac{1}{2} \cdot \frac{x^3}{3} + \frac{1 \cdot 3}{2 \cdot 4} \cdot \frac{x^5}{5} - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot \frac{x^7}{7} + \dots \quad (R = 1).$$

$$4. \operatorname{arctg}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots \quad (R = 1).$$

$$5. \operatorname{arcctg}(x) = \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5} + \frac{x^7}{7} - \dots \quad (R = 1).$$

$$6. \arcsin(x) = x + \frac{1}{2} \cdot \frac{x^3}{3} + \frac{1 \cdot 3}{2 \cdot 4} \cdot \frac{x^5}{5} + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot \frac{x^7}{7} + \dots \quad (R = 1).$$

$$7. \arccos(x) = \frac{\pi}{2} - \left(x + \frac{1}{2} \cdot \frac{x^3}{3} + \frac{1 \cdot 3}{2 \cdot 4} \cdot \frac{x^5}{5} + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot \frac{x^7}{7} + \dots \right) \quad (R = 1).$$

$$8. \frac{1}{\sqrt{1-x^2}} = x + \frac{1}{2} \cdot x^2 + \frac{1 \cdot 3}{2 \cdot 4} \cdot x^4 + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot x^6 + \dots \quad (R = 1).$$

9. $\frac{1}{\sqrt{1+x}} = x - \frac{1}{2} \cdot x + \frac{1 \cdot 3}{2 \cdot 4} \cdot x^2 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot x^3 + \dots$ $(R=1).$
10. $\sqrt{1+x} = 1 + \frac{1}{2} \cdot x - \frac{1}{2 \cdot 4} \cdot x^2 + \frac{1 \cdot 3}{2 \cdot 4 \cdot 6} \cdot x^3 - \dots$ $(R=1).$
11. $\frac{1}{(1+x)^3} = 1 - \frac{2 \cdot 3}{2} \cdot x + \frac{3 \cdot 4}{2} \cdot x^2 - \frac{4 \cdot 5}{2} \cdot x^3 + \dots$ $(R=1).$
12. $\frac{1}{(1+x)^2} = 1 - 2 \cdot x + 3 \cdot x^2 - 4 \cdot x^3 + 5 \cdot x^4 - \dots$ $(R=1).$
13. $\frac{1}{1+x} = 1 - x + x^2 - x^3 + x^4 - \dots$ $(R=1).$
14. $\ln \frac{1+x}{1-x} = 2 \cdot \left(x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots \right)$ $(R=1).$
15. $\ln(1-x) = -\frac{x}{1} - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \dots$ $(R=1).$
16. $\ln(1+x) = \frac{x}{1} - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$ $(R=1).$
17. $\text{ch}(x) = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots$ $(R=\infty).$
18. $\text{sh}(x) = 1 + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots$ $(R=\infty).$
19. $\text{arth}(x) = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots$ $(R=1).$
20. $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$ $(R=\infty).$
21. $\sin(x) = 1 - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$ $(R=\infty).$
22. $e^{-x} = 1 - \frac{x}{1!} + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots$ $(R=\infty).$
23. $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ $(R=\infty).$

3.3. Решения

1.

```
#include <iostream.h>
using namespace std;
int main ()
```

```

{
    int i, res = 1, n;
    cin >> n;
    i = 1;
    while (i <= n)
    {
        res *= i;
        i++;
    }
    cout << "res=" << res;
    return 0;
    system ("pause");
}

```

2. Выводятся значения функции $y = x^2 + 1$ в заданном диапазоне изменений аргумента x от x_N до x_k с заданным шагом dx .

3.

```

#include <iostream.h>
using namespace std;
int main()
{
    int N; //верхняя граница
    int k; // индексная переменная
    double s = 1; // произведение
    cout << "Enter N=";
    cin >> N;
    for (k = 2; k <= N; k++)
        s *= (1- (double)2/(k*(k+1)));
    cout << "Product =" << s << endl;
    return 0;
    system("pause");
}

```

4.

```

#include <iostream.h>
using namespace std;
int main()
{
    float e, a, xn, xn1;
    int p;
    cin >> a >> p >> e;
    xn = a;
    int k = 1;
    xn1 = ((p-1)*xn - a/xn)/p;

```

```

while (abs(xn1 - xn)<=e)
{
    xn = xn1;
    xn1 = ((p-1)*xn - a/xn)/p;
    k++;
}
cout << "Число итераций -" << k;
cout << "x=" << xn;
return 0;
system("pause");
}

```

5.

```

#include <iostream>
using namespace std;
int main()
{
    double x, s;
    //вводимое число и результат
    char op; //вводимая операция
    cout << ">>";
    cin >> x; s = x;
    while(true)
    {
        cout << ">>";
        cin >> op; // ввод операции
        if (op == "=")
        {
            cout << "=>" << s << endl;
            exit(0);
        }
        cout << ">>"; cin >> x; s = x;
        switch (op)
        {
            case '+': s += x; break;
            case '-': s -= x; break;
            case '*': s *= x; break;
            case '/': s /= x; break;
        }
    }
    return 0;
    system("pause");
}

```

6.

```
#include <iostream>
using namespace std;
int main()
{
    float x, min, max;
    int i = 1;
    cin >> min;
    max = min;
    while (i<=99)
    {
        cin >> x;
        min = x<min ? x : min;
        if (x>max) max = x;
        i ++;
    }
    cout << max - min;
    return 0;
    system("pause");
}
```

3.4. Указания к решению

8. Уравнение $f(x) = 0$ на заданном отрезке $[a, b]$ имеет единственный корень, если $f(a) \cdot f(b) \leq 0$ и функция непрерывна на отрезке.

14. Число называется совершенным, если оно равно сумме своих делителей.

15. Для вычисления полинома n -й степени

$$y = a_1 x^n + a_2 x^{n-1} + \dots + a_n x + a_{n+1}$$

используется формула Горнера

$$(\dots((a_1 x + a_2)x + a_3)x + \dots + a_n)x + a_{n+1}.$$

19. Если, например, при встрече мухи с поездом, следующим из пункта А, расстояние между поездами было y , то время полета до поезда, следующего из пункта В, определится по формуле $t = y / (v + v_2)$. Следовательно, муха пролетит расстояние $x = t \cdot v$, а расстояние между поездами станет $y = t(v_1 + v_2)$. При полете до поезда, следующего из пункта А, $t = y / (v + v_1)$.

36. Формула прямоугольников при вычислении интеграла имеет вид

$$\int_a^b f(x) dx \approx h \left[f(a + h/2) + f(a + h/2 + h) + \dots + f(a + h/2 + (n-1)h) \right],$$

где $h = (b - a) / n$.

Во всех последующих заданиях исходные значения элементов массивов вводятся через стандартный входной поток.

Глава 4. СТАТИЧЕСКИЕ ОДНОМЕРНЫЕ МАССИВЫ

4.1. Условия заданий с решениями

1. Подсчитать количество неотрицательных элементов второй половины одномерного целочисленного массива.

2. В одномерном массиве целых чисел найти среднее арифметическое положительных элементов.

3. В одномерном массиве целых чисел могут быть равные элементы. Обменять местами значения первого максимального и последнего минимального элементов.

4. Найти скалярное произведение двух n -мерных векторов, координаты которых – вещественные числа.

Указание. Скалярное произведение двух векторов

$$a = (a_1, a_2, \dots, a_n) \text{ и } b = (b_1, b_2, \dots, b_n)$$

вычисляется по формуле

$$a * b = \sum_{i=1}^n a_i \cdot b_i.$$

5. В одномерном массиве из n целых чисел найти сумму цифр первого положительного элемента.

6. Отсортировать элементы одномерного целочисленного массива по возрастанию методом пузырька.

Указание. Каждый элемент массива, начиная с первого, сравнивается с соседним, и, если первый из сравниваемых элементов больше второго, то элементы обмениваются своими значениями. Процесс повторяется.

7. Сформировать случайным образом целочисленный массив a из 20 элементов. Ввести координаты концов целочисленного отрезка $[c, b]$. Вывести сообщение, в какую из трех частей числовой оси попадает больше всего точек, заданных массивом a .

4.2. Решения

Для корректного вывода на экран кириллического текста необходимо: подключить к программе заголовочный файл `windows.h`, включить в текст программы настройки `SetConsoleCP(1251); SetConsoleOutputCP(1251);` и установить в окне вывода результатов шрифт `Lucida Console`.

1.

```
#include <iostream>
#include <windows.h>
#include <conio.h>
using namespace std;
void main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int n=20;
    int i, a[n];
    system ("cls") ;
    // ввод элементов одномерного массива с клавиатуры
    cout << "Введите элементы массива с клавиатуры: \n";
    for (i=0; i<n; i++)
        cin >> a[i];
    cout << "Введен массив:\n";
    for (i=0; i<n; i++)
        cout << a[i] << ' ';
    cout << "\n";
    //Подсчет col_neotr - числа неотрицательных
    int col_neotr = 0;
    //элементов второй половины массива
    for (i=n/2; i<n; i++)
        if (a[i] >= 0) col_neotr++;
    if (col_neotr != 0)
        cout << "Количество неотрицательных элементов второй
                половины массива равно" << col_neotr << "\n";
    else
        cout << "Во второй половине массива все элементы
                неотрицательны";
    getch();
}
```

2.

```
#include <iostream>
#include <windows.h>
using namespace std;
void main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int n = 20;
    int i, a[n];
```

```

int s=0, k=0;
system("cls");
for (i=0; i<n; i++)
    cin >> a[i];
cout << "Исходный массив:\n";
for (i=0; i<n; i++)
    cout << a[i] << ' ';
cout << "\n";
for (i=0; i<n; i++)
    if (a[i]>0)
    {
        s+=a[i];
        k++;
    }
if (k>0)
    cout << "Среднее арифметическое положительных элементов
        равно" << s/k;
else
    cout << "В массиве нет положительных элементов";
system("pause");
}

```

3.

```

#include <iostream>
#include <windows.h>
using namespace std;
void main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int n=20;
    int i, a[n];
    system ("cls");
    // ввод элементов одномерного массива с клавиатуры
    cout << "Введите элементы массива с клавиатуры: \n";
    for (i=0; i<n; i++)
        cin >> a[i];
    cout << "Введен массив:\n";
    for (i=0; i<n; i++)
        cout << a[i];
    cout << "\n";
    int max, min, k, p;
    max = a[0]; k=0;
    min = a[0]; p=0;
}

```

```

for (i=1; i<n; i++)
{
    if (a[i] > max)
    {
        max = a[i]; k=i;
    }
    if (a[i] < min)
    {
        min = a[i]; p=i;
    }
}
a[k] = min; a[p] = max;
cout << "Получен массив:\n";
for (i=0; i<n; i++)
    cout << a[i] << ' ';
cout << "\n";
system("pause");
}

```

4.

```

#include <iostream>
#include <windows.h>
#include <conio.h>
using namespace std;
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int n=10; // размерность векторов
    int i; // индексная переменная
    double a[n]; // первый массив (вектор)
    double b[n]; // второй массив (вектор)
    double scalar = 0; // переменная для результата
    cout << "Введите координаты первого вектора";
    for (i=0; i<n; i++)
        cin >> a[i];
    cout << "Введите координаты второго вектора";
    for (i=0; i<n; i++)
        cin >> b[i];
    // Вычисление скалярного произведения
    for (i=0; i<n; i++)
        scalar += a[i] * b[i];
    system("cls");
}

```

```

    cout << "Скалярное произведение векторов равно " << scalar <<
endl;
    getch();
    return 0;
}

```

5.

```

#include <iostream>
#include <windows.h>
#include <conio.h>
using namespace std;
void main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int n=20;
    int i, a[n];
    system ("cls");
    //ввод элементов одномерного массива с клавиатуры
    cout << "Введите элементы массива с клавиатуры: \n";
    for (i=0; i<n; i++)
        cin >> a[i];
    cout << "Введен массив:\n";
    for (i=0; i<n; i++)
        cout << a[i] << ' ';
    cout << "\n";
    int k; // номер искомого положительного элемента
    cout << "Введите номер для поиска k=";
    cin >> k;
    if (k<=n)
    {
        int kp = 0; //количество положительных элементов
        int p = -1; // индекс текущего положительного элемента
        for (i=0; i<n; i++)
        {
            if (a[i] > 0)
            {
                kp++;
                p = i;
                if (kp == k) break;
                // положительный элемент с заданным
                // порядковым номером найден
            }
        }
        if (kp == k)

```

```

    {
        cout << "Элемент a[" << p << "]" "=" << a[p] <<
            "является" <<
            kp << "-м положительным элементом массива\n";
        //вычисление s - суммы цифр найденного элемента
        int b = a[p];
        int s = 0;
        while (b>0)
        {
            s += b % 10; b /= 10; }
        cout << "Сумма цифр найденного положительного
            элемента массива равна " <<s << endl;
        }
        else
            cout << "В массиве нет положительного элемента
                с порядковым номером " << k << "\n";
    }
    getch();
}

```

6.

```

#include <iostream>
#include <windows.h>
#include <stdlib.h>
#include <conio.h>
#include <time.h>
using namespace std;
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int n=20;
    int i, j, b;
    // формирование исходного массива
    srand(time(NULL));
    for (i=0; i<n; i++)
    {
        a[i] = rand() % 10;
        cout << a[i] << ' ';
    }
    //сортировка сформированного массива
    for (j=1; j<=n-1; j++)
        for (i=0; i<n-j; i++)
        {

```

```

        if (a[i] > a[i+1])
        {
            b = a[i+1]; a[i+1] = a[i]; a[i] = b;
        }
    }
    cout << "Массив после сортировки\n";
    for (i=0; i<n; i++)
    cout << a[i] << ' ';
    cout << "\n";
    getch();
    return 0;
}

```

7.

```

#include <iostream>
#include <string>
#include <windows.h>
#include <time.h>
using namespace std;
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    srand(time(NULL));
    const int n=20;
    int i, j, b, a[n];
    // формирование исходного массива
    for (i=0; i<n; i++)
    {
        a[i] = rand() % 10;
        cout << a[i] << ' ';
    }
    int c;
    cout << "\nВведите концы целочисленного отрезка" << endl;
    cin >> c >> b;
    if (b<c)
    {
        i=b; b=c; c=i;
    }
    int k1=0, k2=0, k3=0;
    // k1 – количество элементов массива a, принадлежащих
    // полуинтервалу  $(-\infty, c]$ 
    // k2 – количество элементов массива a, принадлежащих
    // интервалу  $[c, b]$ 

```

```

// k3 – количество элементов массива a, принадлежащих
// полуинтервалу [b, +∞)
for (i=0; i<n; i++)
    if (a[i] < c)
        k1++;
    else
        if (a[i] <= b)
            k2++;
        else
            k3++;
string s; int max;
if (k1 > k2)
{
    s = "слева от c"; max = k1;
}
else
{
    max = k2;
    s = "внутри отрезка";
}
if (k3 > max)
s = "справа от b";
cout << "Максимальное количество элементов массива a
        попадает" << s;
system("pause");
return 0;
}

```

Глава 5. ИСПОЛЬЗОВАНИЕ АДРЕСНОЙ АРИФМЕТИКИ ПРИ РАБОТЕ С ОДНОМЕРНЫМИ МАССИВАМИ

5.1. Условия заданий с решениями

1. Сформулировать условие задачи, программа решения которой приведена ниже

```
#include <iostream>
#include <windows.h>
using namespace std;
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int a[15], * b;
    b = a;
    for (int i=0; i<15; i++)
    {
        b[i] = 15 - i;
        cout << *(b+i) << ' ';
    }
    cout << "\n";
    return 0;
}
```

Каков результат работы этой программы?

2. Массив A состоит из целых чисел: 957, -10, 1851, -1000, -1, 7, -251. Перенести элементы массива A в массив B , сложив каждый положительный элемент с его младшей цифрой.

3. Составить программу для вычисления полинома

$$y = 2x^8 - x^6 + 4x^5 - 5x^2 + 6x + 1,$$

используя формулу Горнера.

Указание. Формула Горнера для вычисления многочлена n -й степени

$$y = a_1x^n + a_2x^{n-1} + \dots + a_nx + a_{n+1}$$

имеет вид

$$y = (\dots(((a_1x + a_2)x + a_3)x + \dots + a_n)x + a_{n+1}.$$

4. Без использования дополнительного массива сдвинуть все 10 элементов массива a вправо на k элементов ($k > 1$).

5. Составить программу для определения количества дней невисокосного года, прошедших с начала года до даты, введенной пользователем.

6. Ввести объем продаж изделий за каждый день недели (кроме воскресенья). Вывести название дня недели и отклонение от среднего арифметического объема продаж за неделю для каждого рабочего дня недели.

5.2. Решения

1. В программе оператором `int a[15]` объявляется целочисленный массив из 15 элементов. Оператор `int * b` служит для объявления указателя на поле памяти целочисленного типа. Далее переменной `b` оператором `b = a` присваивается адрес начального элемента массива `a`. Оператор цикла, заголовок которого имеет вид

```
for (int i = 0; i < 15; i++) ,
```

производит заполнение массива `a` числами от пятнадцати до одного и вывод этих значений на экран оператором

```
cout << *(b+i) << ' ' ;
```

Результат работы приведенной программы имеет вид

```
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

2.

```
#include <iostream>
#include <windows.h>
using namespace std;
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int A[] = {957, -10, 1851, - 1000, -1, 7, -251}, B[7];
    int i;
    for (i = 0; i<7; i++)
        if (*(A+i) > 0)
            *(B+i) = *(A+i) + *(A+i) % 10;
        else
            *(B+i) = *(A+i);
    int * pt; pt = B;
    for (i = 0; i<7; i++)
        cout << *(pt++) << endl;
    system("pause");
    return 0;
}
```

3.

```
#include <iostream>
#include <windows.h>
using namespace std;
int main()
```

```

{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int n = 10;
    int i, col, a[n];
    float x, y;
    cout << "Введите количество коэффициентов" << endl;
    cin >> col;
    if (col > n)
        cout << "Превышена размерность массива коэффициентов"
            << endl;
    else
    {
        cout << "Введите коэффициенты полинома, начиная со
                свободного члена" << endl;
        for (i=0; i<col; i++)
            cin >> *(a+i);
        cout << "Коэффициенты полинома:" << endl;
        for (i=0; i<col; i++)
            cout << *(a+i) << ' ';
        cout << "Введите x" << endl;
        cin >> x;
        y = *(a+1);
        for (i=1; i<col; i++)
            y = y*x + *(a+i);
        cout << "Результат:" << endl;
        cout << "y = " << y << endl;
    }
    system("pause");
    return 0;
}

```

4.

```

#include <iostream>
#include <windows.h>
using namespace std;
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int n = 10;
    int i, j, k, c;
    int a[n];
    int * x;
    x = a;

```

```

cout << "Введите элементы исходного массива"<< endl;
for (i=0; i<n; i++)
    cin >> *(x++);
cout << "Введите k" << endl;
cin >> k;
x = a;
if (k>1)
{
    for (i=0; i<k; i++)
    {
        c= *(x + n -1);
        for (j=n -1; j<=1; j- -)
            *(x++) = *(x- - );
        *x = c;
    }
    cout << "Полученный массив:" << endl;
    x = a;
    for (i=0; i<n; i++)
        cout << *(x++) <<' ';
}
system("pause");
return 0;
}

```

5.

```

#include <iostream>
#include <windows.h>
using namespace std;
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int month, day, skolko;
    int days[12]={31,28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    cout << "\nВведите месяц (число от 1 до 12):";
    cin >> month;
    cout << "\nВведите день (число от 1 до 31):";
    cin >> day;
    skolko = day;
    for (int j = 0; j < month-1; j++)
        skolko += *(days + j);
    cout << "Общее число дней с начала года:" << skolko << endl;
    system("pause");
    return 0;
}

```

6.

```
#include <iostream>
#include <windows.h>
using namespace std;
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int n = 6;
    float a[n];
    cout << "Введите объем продаж за каждый из шести дней
            недели \n";
    float * x;
    x = a; int j;
    for ( j = 0; j < n; j++)
        cin >> *(x++);
    cout << "Объем продаж за каждый рабочий день недели \n";
    x = a;
    for ( j = 0; j < n; j++)
        cout << *(x++);
    float s = 0; x = a;
    for ( j = 0; j < n; j++)
        s += *(x++);
    s /= n;
    cout << "\nСредний объем продаж за неделю" << s << endl';
    // отклонение от среднего арифметического для каждого
    // рабочего дня получим в массиве a
    x = a;
    for ( j = 0; j < n; j++)
    {
        *(a+j) = (*(x+j) - s)* 100 / *(x+j);
        switch(j)
        {
            case 0: cout << "понедельник " << a[0] << "%\n"; break;
            case 1: cout << "вторник " << a[1] << "%\n"; break;
            case 2: cout << "среда " << a[2] << "%\n"; break;
            case 3: cout << "четверг " << a[3] << "%\n"; break;
            case 4: cout << "пятница " << a[4] << "%\n"; break;
            case 5: cout << "суббота " << a[5] << "%\n"; break;
        }
    }
    system("pause");
    return 0;
}
```

Глава 6. ДИНАМИЧЕСКИЕ ОДНОМЕРНЫЕ МАССИВЫ

6.1. Условия заданий с решениями

1. Используя динамический массив, найти среднее арифметическое его положительных элементов, расположенных после последнего отрицательного.

2. Составить программу для вычисления математического ожидания для дискретной случайной величины.

Указание. Математическое ожидание M для дискретной случайной величины x , которая может принимать значения x_1, x_2, \dots, x_n с вероятностями p_1, p_2, \dots, p_n вычисляется по формуле

$$M = \sum_{i=1}^n x_i \cdot p_i, \text{ причем } \sum_{i=1}^n p_i = 1.$$

Чему равно математическое ожидание для случайной величины, которая принимает значения

0.5, 1.2, 2.5, 3.1, 4.9

с вероятностями

0.1, 0.2, 0.1, 0.25, 0.35.

3. Составить программу сортировки по возрастанию одномерного массива из n различных целочисленных элементов методом простого выбора.

Указание. Алгоритм сортировки методом выбора состоит в следующем. Находится наименьший элемент массива и меняется местами с первым элементом. Далее рассматриваются элементы массива, начиная со второго. Наименьший из них меняется местами со вторым элементом и так далее $n-1$ раз.

6.2. Решения

1.

```
#include <iostream>
#include <windows.h>
using namespace std;
int main ()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int i, n;
    cout << "Введите количество элементов массива\n";
    cin >> n;
    float s, * a = new float [n];
```

```

cout << "Введите элементы массива\n";
for (i=0; i<n; i++)
cin >> *(a+i);
bool flag = false; //отрицательный элемент массива не найден
s = 0; // сумма положительных
int k = 0; // количество положительных
for (i = n -1; i >= 0; i--)
{
    if (*(a+i) < 0)
    {
        flag = true; break;
    };
    if (*(a+i) > 0)
    {
        s += *(a+i); k++;
    }
}
if (flag)
    if (k>0)
        cout << "Среднее арифметическое положительных равно"
            << s/k << endl;
    else
        cout << "Нет положительных"<< endl;
else
    cout << "Нет отрицательных"<< endl;
delete [] a;
system("pause");
return 0;
}

```

2.

```

#include <iostream>
#include <windows.h>
using namespace std;
int main ()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int n, i;
    cout << "Введите количество различных значений случайной
        величины" << endl;
    cin >> n;
    double M, * x = new double[n], * p = new double [n];
    cout << "Введите значения случайной величины"<< endl;
}

```

```

for (i=0; i<n; i++)
    cin >> *(x+i);
cout << "Введите вероятности случайной величины"<< endl;
for (i=0; i<n; i++)
    cin >> *(p+i);
double s = 0;
for (i=0; i<n; i++)
    s += *(p+i);
if (abs (s - 1) <= 0.000000001)
{
    for (i=0; i<n; i++)
        M += *(p+i) * (*(x+i));
    // Вывод результата
    cout << "M = " << M << endl;
}
else
    cout << "Неверный ввод вероятностей" << endl;
delete [] x; delete [] p;
system("pause");
return 0;
}

```

Для предложенных входных данных ответ: $M = 3.03$

3.

```

#include <iostream>
#include <windows.h>
using namespace std;
int main ()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int n, i, j;
    cout << "Введите количество элементов массива"<< endl;
    cin >> n;
    int * a = new int [n];
    cout << "Введите элементы массива" << endl;
    for (i=0; i<n; i++)
        cin >> *(a+i);
    for (i=0; i<n -1; i++)
    {
        int imin = i;
        for (j=i+1; j<n; j++)
            if (*(a+j) < *(a+imin))
                imin = j;
    }
}

```

```

    int b = *(a+i);
    *(a+i) = *(a+imin);
    *(a+imin) = b;
}
cout << "Упорядоченный по возрастанию массив:" << endl;
for (i=0; i<n; i++)
    cout << *(a+i) << ' ';
delete [] a;
system("pause");
return 0;
}

```

6.3. Тренировочные задания

1. Массив x содержит $n = 30$ элементов вещественного типа. Вычислить значение переменной y .

а) $y = \sqrt[n]{|x_1 \cdot x_2 \cdots x_n|}$;

б) $y = \max_i \{x_i\} + \min_i \{x_i\}$;

в) $y = x_1 - x_2 + x_3 - \dots - x_{n-1} + x_n$;

г) $y = x_1 \cdot x_n + x_2 \cdot x_{n-1} + \dots + x_n \cdot x_1$;

д) $y = x_1^2 + x_2^2 + \dots + x_n^2$;

е) $y = x_1 + 2x_2 + 3x_3 + \dots + nx_n$;

ж) $y = x_n \cdot (x_n + x_{n-1}) \cdot (x_n + x_{n-1} + x_{n-2}) \cdots (x_n + \dots + x_1)$;

з) $y = (x_1 - 1) \cdot (x_2 - 2) \cdots (x_n - n)$.

2. Даны 50 целых чисел. Вывести их на экран в порядке, обратном порядку ввода, по 5 чисел в каждой строке экрана.

3. Даны 50 целых чисел. Определить, сколько из них отличны от полусуммы первого и последнего числа.

4. В одномерном массиве из n вещественных чисел определить произведение отрицательных, среднее арифметическое положительных и количество нулевых элементов.

5. Создать таблицу значений функции $y = \sin(x^2 - 1) + 2$ в точках отрезка $[a, b]$, полученных разбиением этого отрезка на 30 равных частей.

6. В массиве из n элементов вещественного типа найти последнюю цифру целой части среднего арифметического элементов, расположенных в массиве на нечетных местах.

7. В одномерном массиве из n целых чисел к каждому максимальному элементу прибавить его младшую цифру.

8. Если первый из максимальных элементов одномерного целочисленного массива A находится в первой половине массива, сдвинуть циклически все элементы массива влево на одну позицию.

9. Если в одномерном массиве A нет ни одного отрицательного элемента, обменять местами значения элементов первой и второй половин массива.

10. В одномерном массиве A каждый элемент, кроме первого и последнего, заменить полусуммой двух его соседних элементов. К первому и последнему элементам прибавить их младшие цифры.

11. Найти сумму цифр максимального элемента целочисленного одномерного массива A , если известно, что в этом массиве все числа положительные.

12. Если в одномерном целочисленном массиве все элементы упорядочены по возрастанию, то сменить порядок их следования на противоположный.

13. Найти произведение положительных элементов одномерного массива целых чисел с четными номерами, учитывая лишь те из них, которые кратны трем.

14. Найти произведение четных элементов с нечетными индексами в одномерном массиве, который состоит из положительных целых чисел. Каждый элемент с четным индексом увеличить на это произведение.

15. Если в одномерном массиве вещественных чисел два соседних элемента с нечетным и четным индексом имеют одинаковые знаки, обменять значения этих элементов местами, в противном случае к меньшему из них прибавить младшую цифру большего.

16. Пусть в числовом массиве содержатся результаты измерений температуры воздуха, которые проводились ежедневно один раз в день в течение декабря. Определите:

- a) среднемесячную температуру декабря;
- b) сколько раз температура была выше 0°C ;
- c) день (число), когда температура была наибольшей;
- d) день (число), когда температура первый раз поднялась выше 0°C и увеличьте эту температуру на 1°C ;
- e) любой из самых холодных дней (число) декабря;
- f) сколько дней в декабре температура была выше средней;
- g) день (число), когда температура была ближе всего к средней температуре декабря;
- h) максимальную температуру второй декады декабря;
- i) минимальную температуру тех дней декабря, которые следуют после последнего из самых теплых дней в этом месяце;

ж) температуры любого из самых холодных и самых теплых дней и поменяйте местами их значения;

к) среднюю температуру тех дней, которые предшествуют первому из самых холодных дней в декабре;

л) любые два самых холодных дня (числа);

м) сколько раз в декабре температура меняла знак;

н) первую из минимальных температур и поместите ее на последнее место в массиве, сдвинув остальные в направлении начала массива;

о) среднюю температуру тех дней декабря, которые расположены между последним днем с максимальной температурой и первым днем с минимальной температурой;

р) сколько дней в наибольшем периоде непрерывного роста температуры;

с) среднюю температуру выходных дней (суббота, воскресенье), если известно, что 1 декабря – пятница;

т) последнюю из максимальных температур. Обменяйте местами найденное значение с первой отрицательной температурой;

у) среднюю температуру последней декады декабря и поместите ее на первое место в массиве, сдвинув все остальные значения температур на одну позицию вправо.

17. Известны номера выигрышных билетов x_1, x_2, \dots, x_{40} . С клавиатуры вводится номер билета s . Определите, является ли он выигрышным.

18. Список содержит года рождения 50 сотрудников предприятия. Каков возраст самого молодого из них? Через сколько лет возраст самого молодого сотрудника превысит средний возраст коллектива, вычисленный на текущий момент?

19. В двух таблицах содержатся номера выигрышных лотерейных билетов: первая содержит n элементов, вторая – m элементов. Требуется сформировать новую таблицу, в которой номера билетов этих двух таблиц располагались бы в возрастающем порядке.

20. Одномерный массив содержит сведения о количестве выпущенных изделий по каждому из цехов некоторого предприятия: положительные значения означают количество изделий, выпущенных сверх плана, отрицательные – недовыполнение плановых заданий, ноль – выполнение плана.

Определите:

а) сколько цехов перевыполнило план, сколько не выполнило и сколько выполнило плановое задание;

б) сколько всего изделий выпущено сверх плана в цехах, где достигнуто перевыполнение плана, и сколько изделий недополучено в цехах, где имеет место невыполнение плана;

Переставьте значения элементов массива так, чтобы сначала располагались сведения о перевыполнении плана (в любом порядке), затем все остальные.

21. В первой строке заказа, поступившей из столовой на продуктовую базу, указано, сколько кг картофеля, риса, муки, сахара, моркови, манной крупы, сколько банок соков и сколько бутылок минеральной воды требуется привезти, во второй строке указана стоимость в рублях единицы каждого продукта. Требуется определить общую стоимость заказанных продуктов (в рублях).

22. Справочник содержит список из 100 номеров внутренних телефонов предприятия. Известно, что в ряде отделений и цехов телефоны были сняты, их номера переданы на АТС, а в таблице на их места занесены нулевые номера. Уплотнить таблицу справочника номеров телефонов.

Указание. Уплотнение таблицы сводится к «вытеснению» нулевых номеров и к сдвигу оставшихся номеров в направлении начала таблицы.

23. Одномерный массив A содержит n целочисленных значений. Требуется положительные элементы массива, отличные от максимального, поместить в массив B , отрицательные – в массив C .

24. Массив A содержит n целочисленных значений. Требуется создать новый массив из суммы цифр соответствующих элементов исходного массива.

25. Одномерный массив содержит результаты контрольной работы по информатике 25 студентов группы. Определить:

- а) число студентов, получивших оценку 5;
- б) число студентов, получивших положительную оценку;
- в) число студентов, получивших оценку 5 или 4;
- г) среднюю оценку группы.

26. Одномерный массив x содержит 20 вещественных чисел, каждое из которых больше 1. Создать два новых одномерных массива A и B . В массив A поместите результаты вычисления y_n по итерационной формуле

$$y_{n+1} = \frac{1}{2} \left(y_n + \frac{x_i}{y_n} \right)$$

с заданной погрешностью E , задав начальное приближение $y_0 = x_i$. В массив B поместите результаты вызова встроенной функции вычисления квадратного корня из соответствующего элемента заданного массива.

Указание. Задаваемая погрешность E удовлетворяется, когда

$$\text{abs}(y_{n+1} - y_n) \leq E.$$

27. Даны координаты x, y точки плоскости и одномерный массив A из n целых чисел. Считая элементы массива радиусами окружностей с центром в начале координат, определить:

- а) минимальный радиус круга, в который попадает точка;

б) радиусы двух ближайших окружностей, между которыми находится точка с координатами x, y ;

с) сколько кругов содержат точку с координатами x, y .

28. Даны два одномерных массива A и B целых чисел, каждый из которых содержит n элементов. Требуется сформировать содержимое третьего массива C по следующему правилу:

а) в начало массива C помещаются элементы массива A , затем элементы массива B ;

б) в начало массива C помещаются элементы массива A , причем в порядке, обратном исходному, затем элементы массива B ;

с) элементы массивов A и B пересылаются в массив C , причем порядок следования элементов в массивах A и B меняется на противоположный;

д) в начало массива C пересылаются элементы того массива, который содержит меньшее количество нулевых элементов, считая, что хотя бы в одном массива A или B есть по крайней мере один нулевой элемент;

е) сначала в массив C пересылаются положительные элементы обоих массивов A и B , затем все остальные.

29. Известен список, содержащий годы рождения 20 посетителей поликлиники. Добавить в этот список свой год рождения и отсортировать его по возрастанию возраста посетителей.

30. Соревнования по фигурному катанию оценивают 12 судей. Сначала они выставляют оценку за сложность элементов катания, затем оценку за артистичность исполнения. При выведении двух итоговых оценок из всей совокупности оценок удаляется самая высокая оценка и самая низкая оценка, для оставшихся оценок вычисляется среднее арифметическое. Если окажется, что самую высокую или самую низкую оценку выставили несколько судей, то из совокупности оценок удаляется только одна из них. Требуется создать программу для вычисления оценки за сложность и оценки за артистичность для одного фигуриста.

31. Для заданного массива x из 20 целых чисел составить таблицу значений функции, если p и a задаются вводом.

а) $y = 5((((x+a)^2 + a)^2 + a)^2 + a) - 2a$;

б) $y = a + (a + (a + (a + (a + (a + (a + x)^6)^6)^6)^6)^6 + p$;

с) $y = ((((((x+a)^2 + 2a)^2 + 3a)^2 + 4a)^2 + 5a)^2 + 6a)^2 - px$;

д) $y = \frac{p(x-2)(x-4)\cdots(x-64)}{((x-1)(x-3)\cdots(x-65))}$;

е) $y = a((((((x-p)x-p)x-p)x-p)x-p)x-p)$;

ф) $y = p((((((x-p)x-2p)x-3p)x-4p)x-5p)x-6p)x+a$.

32. Справочник содержит упорядоченный по возрастанию массив из 100 телефонов предприятия. Известно, что в ряде отделений и цехов теле-

фоны были сняты, номера таких телефонов хранятся в упорядоченном по возрастанию номеров массиве. Уплотнить таблицу номеров телефонов предприятия, удалив из нее номера снятых телефонов.

33. Найти угол φ между двумя n -мерными векторами, используя формулу

$$\cos \varphi = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2 \sum_{i=1}^n b_i^2}},$$

где a_i – координаты вектора a , b_i – координаты вектора b , $i = 1, 2, \dots, n$.

34. Пусть даны вещественные числа a_1, a_2, \dots, a_{20} . Вычислить:

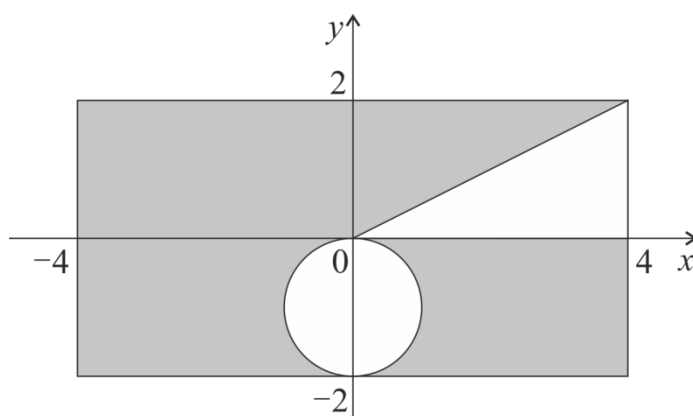
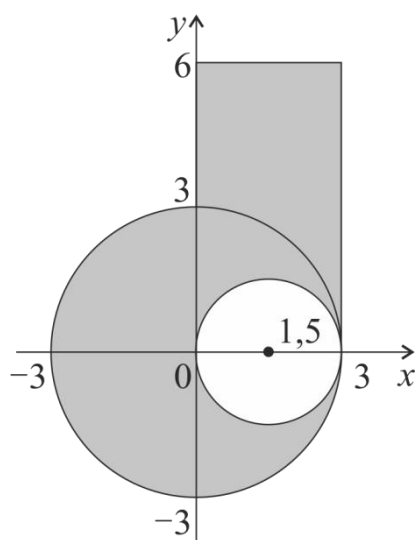
$$b = \frac{1}{20} \sum_{i=1}^{20} a_i, \quad S = \sqrt{\frac{1}{20} \sum_{i=1}^{20} a_i^2}.$$

6.4. Задания для самостоятельной работы

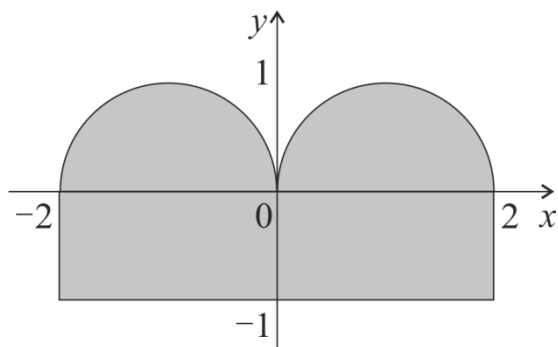
6.4.1. Задание № 1

1. Массив X содержит целочисленные абсциссы n точек на плоскости, массив Y – целочисленные ординаты этих точек. Сколько точек попадают в заштрихованную область? (рис. 1–8)

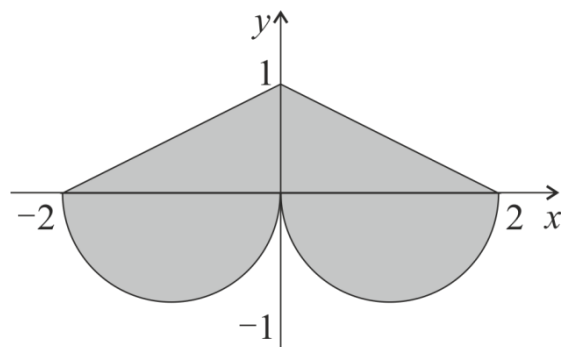
1.



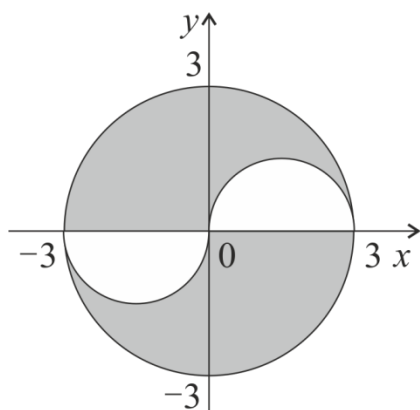
3.



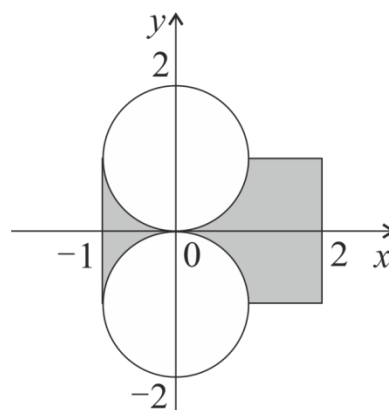
4.



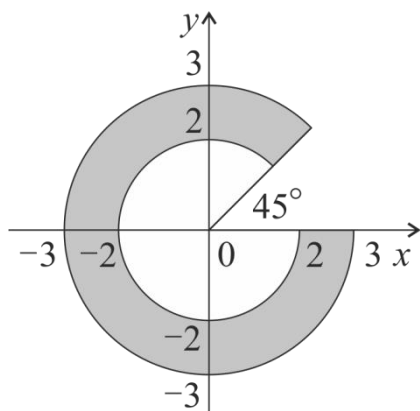
5.



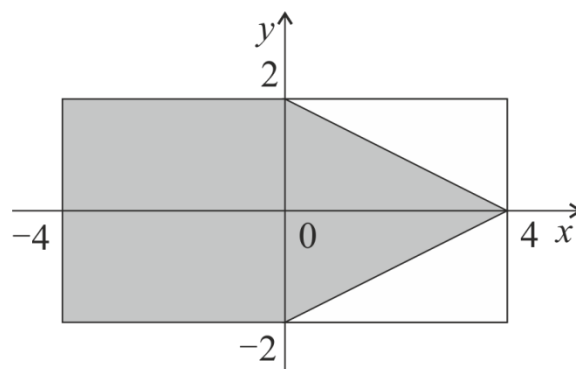
6.



7.



8.



2. Массив X содержит целочисленные абсциссы n точек на плоскости, массив Y – целочисленные ординаты этих точек. Определите количество точек, не попадающих в заштрихованную область (рис. 1–8).

3. Массив X содержит целочисленные абсциссы n точек на плоскости, массив Y – целочисленные ординаты этих точек. Найдите номер точки, попавшей в заштрихованную область и наиболее удаленной от начала координат (рис. 1–8).

4. Массив X содержит целочисленные абсциссы n точек на плоскости, массив Y – целочисленные ординаты этих точек. Найдите координаты точ-

ки, самой близкой к началу координат, среди тех точек, которые не попадают в заштрихованную область (рис. 1–8).

5. Массив A содержит целочисленные координаты n точек плоскости $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Определите количество точек, попадающих в заштрихованную область (рис. 1–8), лежащих на оси X или на оси Y .

6. Массив A содержит целочисленные координаты n точек плоскости $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Определите количество точек, не попадающих в заштрихованную область (рис. 1–8) и не лежащих на оси X или на оси Y .

7. Массив A содержит целочисленные координаты n точек плоскости $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Определите количество точек, попадающих в заштрихованную область (рис. 1–8) и количество точек максимально удаленных от начала координат.

8. Массив A содержит целочисленные координаты n точек плоскости $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Создайте одномерный массив из координат точек x_i, y_i , попадающих в заштрихованную область (рис. 1–8).

9. Массив A содержит целочисленные координаты n точек плоскости $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Создайте одномерный массив из расстояний от начала координат до тех точек плоскости x_i, y_i , которые попадают в заштрихованную область (рис. 1–8).

10. Массив A содержит целочисленные координаты n точек плоскости $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Создайте два одномерных массива X и Y , в которые поместите абсциссы (массив X) и ординаты (массив Y) тех точек плоскости, которые не попадают в заштрихованную область (рис. 1–8).

11. Массив A содержит целочисленные координаты n точек плоскости $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Определите количество точек, расстояние от которых до начала координат больше минимального из таких расстояний.

12. Даны два массива X и Y из n элементов вещественного типа. Изменить порядок следования значений элементов на противоположный в том массиве, среднее арифметическое положительных элементов которого, расположенных между первым из минимальных и последним из максимальных, больше.

13. Дан массив A из n положительных элементов целого типа. Создать новый массив B , в который занести суммы цифр тех элементов массива A , которые не совпадают с наибольшим и с наименьшим элементами.

14. Дан массив A из n положительных элементов целого типа. Создать новый массив B , в который на нечетные места поместить старшие цифры каждого элемента массива A , а на четные – младшие цифры того же элемента.

15. Дан массив A из n положительных элементов целого типа. В каждом элементе этого массива, содержащем более одной цифры, за исключе-

нием максимального и минимального элементов, обменять местами последнюю и предпоследнюю цифры.

16. В одномерном массиве из n вещественных чисел вычислить наибольшее значение модуля разности между двумя соседними элементами. Создать новый массив из исходного, заменив отрицательные и нулевые элементы исходного массива на вычисленное значение.

17. Дан массив A из n вещественных чисел, среди которых могут быть одинаковые. Перенести в новый массив B значения тех элементов массива A , которые отличаются от значения второго элемента по модулю на заданную величину $e < 1$.

18. Дан массив A из n различных целочисленных элементов. Найти два наибольших элемента массива и создать новый массив B из старших цифр тех элементов массива A , которые расположены между двумя найденными элементами массива A .

19. Даны два массива A и B из n целых чисел каждый. Создать новый массив C , в который сначала поместить положительные элементы того из массивов A или B , в котором больше четных значений элементов, расположенных на нечетных местах, затем – отрицательные элементы другого массива.

20. Массив A состоит из n вещественных чисел. Все отрицательные элементы массива, большие минимального, перенести в его начало, а все остальные – в конец массива, сохраняя исходное взаимное расположение как среди отрицательных, так и среди остальных элементов.

21. Все элементы массива A , содержащего n вещественных чисел, расположенные между минимальным и максимальным, циклически сдвинуть вправо на k позиций.

22. Все элементы массива A , содержащего n вещественных чисел, расположенные между минимальным и максимальным, циклически сдвинуть влево на k позиций.

23. Даны целые числа a_1, a_2, \dots, a_{20} . Составить новую последовательность по правилу: то из чисел a_i и a_{i+10} , сумма цифр которого больше, принять за значение b_i , другое считать новым значением b_{i+10} .

24. Даны два массива целых чисел A и B , содержащие по n элементов. Переслат в массив C сначала значения элементов того массива, где больше элементов с максимальной суммой цифр, затем значения элементов другого массива.

25. Дан массив из n различных целых чисел. Создать два новых массива. В первый поместить, сохраняя порядок следования, значения элементов исходного массива, расположенные между двумя наибольшими, в другой – все остальные.

27. В массиве, состоящем из n целочисленных элементов, в каждой группе подряд идущих одинаковых элементов заменить каждый элемент на среднее арифметическое всех элементов массива.

6.4.2. Задание № 2

1. Дан одномерный массив из N вещественных чисел. Определить количество троек соседних элементов, которые могут быть длинами сторон треугольника. Создать массив из площадей таких треугольников, упорядочить его элементы по возрастанию, используя алгоритм сортировки простым выбором.

2. Дан одномерный массив из N вещественных чисел. Определить количество троек соседних элементов, которые могут быть длинами сторон прямоугольного треугольника. Создать массив из площадей таких треугольников, упорядочить его элементы по возрастанию, используя алгоритм сортировки простым обменом.

3. Дан одномерный массив из N вещественных чисел. Определить количество троек соседних элементов, которые могут быть длинами сторон треугольника. Создать массив из периметров таких треугольников, упорядочить его элементы по возрастанию, используя алгоритм сортировки простым выбором.

4. Дан одномерный массив из N вещественных чисел. Определить количество троек соседних элементов, которые могут быть длинами сторон равнобедренного треугольника. Создать массив из периметров таких треугольников, упорядочить его элементы по убыванию, используя алгоритм сортировки простыми включениями.

5. Дан одномерный массив из N вещественных чисел. Определить количество троек соседних элементов, которые могут быть длинами сторон треугольника. Создать массив из наибольших сторон таких треугольников, упорядочить его элементы по убыванию, используя алгоритм сортировки методом пузырька.

6. Дан массив из n целых чисел. Создать новый массив из элементов исходного, значения которых встречаются в нем один раз. Упорядочить элементы нового массива по возрастанию младших цифр, используя сортировку методом пузырька.

7. Дан массив из n целых чисел. Создать новый массив из элементов исходного, значения которых встречаются в нем один раз. Упорядочить элементы нового массива по убыванию старших цифр, используя сортировку методом простых включений.

8. Дан массив из n вещественных чисел. С клавиатуры вводится целое число k . Удалить из массива все элементы, целая часть которых больше k .

Отсортировать по убыванию элементы полученного массива, используя сортировку методом простого выбора.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении начала массива и поместить нулевое значение в последний элемент массива.

9. Даны три одномерных массива вещественных чисел:

$$x_1, x_2, \dots, x_n;$$

$$y_1, y_2, \dots, y_n;$$

$$z_1, z_2, \dots, z_n.$$

Определить количество троек элементов x_i, y_i, z_i ($1 \leq i \leq n$), которые могут быть длинами сторон треугольника. Создать массив из площадей таких треугольников, упорядочить его элементы по возрастанию, используя алгоритм сортировки простым выбором.

10. Даны три одномерных массива вещественных чисел:

$$x_1, x_2, \dots, x_n;$$

$$y_1, y_2, \dots, y_n;$$

$$z_1, z_2, \dots, z_n.$$

Определить количество троек элементов x_i, y_i, z_i ($1 \leq i \leq n$), которые могут быть длинами сторон треугольника. Создать массив из площадей таких треугольников, упорядочить его элементы по возрастанию, используя алгоритм сортировки простым обменом.

11. Даны три одномерных массива вещественных чисел:

$$x_1, x_2, \dots, x_n;$$

$$y_1, y_2, \dots, y_n;$$

$$z_1, z_2, \dots, z_n.$$

Определить количество троек элементов x_i, y_i, z_i ($1 \leq i \leq n$), которые могут быть длинами сторон треугольника. Создать массив из периметров таких треугольников, упорядочить его элементы по возрастанию, используя алгоритм сортировки простым обменом.

12. Даны три одномерных массива вещественных чисел:

$$x_1, x_2, \dots, x_n;$$

$$y_1, y_2, \dots, y_n;$$

$$z_1, z_2, \dots, z_n.$$

Определить количество троек элементов x_i, y_i, z_i ($1 \leq i \leq n$), которые могут быть длинами сторон равнобедренного треугольника. Создать массив из периметров таких треугольников, упорядочить его элементы по убыванию, используя алгоритм сортировки простыми включениями.

13. Даны три одномерных массива вещественных чисел:

$$x_1, x_2, \dots, x_n;$$

$$y_1, y_2, \dots, y_n;$$

$$z_1, z_2, \dots, z_n.$$

Определить количество троек элементов x_i, y_i, z_i ($1 \leq i \leq n$), которые могут быть длинами сторон треугольника. Создать массив из максимальных сторон таких треугольников, упорядочить его элементы по убыванию, используя алгоритм сортировки методом пузырька.

14. Даны две последовательности вещественных чисел:

$$R_1, R_2, \dots, R_n;$$

$$l_1, l_2, \dots, l_n.$$

Создать одномерный массив X из площадей колец с центром в начале координат, внутренний радиус которых равен l_i , внешний – R_i . Упорядочить элементы, стоящие на нечетных местах в массиве X по возрастанию, используя сортировку простыми включениями.

15. Дана последовательность вещественных чисел $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Создать новый массив A из площадей прямоугольников со сторонами x_i, y_i ($1 \leq i \leq n$), отличных от нулевой, максимальной и минимальной площадей. Упорядочить элементы массива A , стоящие на нечетных местах по убыванию, используя сортировку обменом.

16. Дана последовательность n целых чисел. Каждый элемент этой последовательности, начиная со второго, нужно заменить на сумму предшествующих ему элементов. Удалить из этой последовательности все четные элементы, затем отсортировать по убыванию, используя сортировку обменом.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении начала массива и дальнейшую работу производить с частью массива (с 1-го по $n - k$ -й элементы, где n – размер исходного массива, k – количество удаленных элементов).

17. Даны две последовательности целых чисел:

$$x_1, x_2, \dots, x_n;$$

$$y_1, y_2, \dots, y_n.$$

Создать новую последовательность, каждый элемент которой равен тому из чисел x_i, y_i , сумма цифр которого больше. Удалить из второй половины новой последовательности все нечетные элементы, затем отсортировать полученную последовательность методом пузырька.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении

начала массива и дальнейшую работу производить с частью массива (с 1-го по $n - k$ -й элементы, где n – размер исходного массива, k – количество удаленных элементов).

18. Даны две последовательности целых чисел:

$$x_1, x_2, \dots, x_n;$$

$$y_1, y_2, \dots, y_n.$$

Создать новую последовательность, каждый элемент которой равен тому из чисел x_i, y_i , сумма старшей и младшей цифр которого больше. Удалить из первой половины новой последовательности все нечетные элементы, затем отсортировать полученную последовательность методом простых вставок.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении начала массива и дальнейшую работу производить с частью массива (с 1-го по $n - k$ -й элементы, где n – размер исходного массива, k – количество удаленных элементов).

19. Дана последовательность n целых чисел. Каждый элемент этой последовательности, кроме последнего, нужно заменить на сумму следующих за ним элементов. Удалить из этой последовательности все четные элементы, затем отсортировать по убыванию, используя сортировку выбором.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении начала массива и дальнейшую работу производить с частью массива (с 1-го по $n - k$ -й элементы, где n – размер исходного массива, k – количество удаленных элементов).

20. Даны две последовательности из 30 целых чисел каждая. Создать новый массив из тех чисел первой последовательности, которые не входят во вторую. Удалить из новой последовательности все элементы, среди цифр которых есть цифра 5. Отсортировать полученную последовательность по возрастанию методом включений.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении начала массива и дальнейшую работу производить с частью массива (с 1-го по $n - k$ -й элементы, где n – размер исходного массива, k – количество удаленных элементов).

21. Даны две последовательности из n целых чисел каждая. Создать новый массив из тех чисел второй последовательности, которые не входят в первую. Удалить из новой последовательности все числа, среди цифр которых есть младшая цифра максимального элемента первой последовательности, затем отсортировать полученную последовательность по убыванию методом пузырька.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении начала массива и дальнейшую работу производить с частью массива (с 1-го по $n - k$ -й элементы, где n – размер исходного массива, k – количество удаленных элементов).

22. Даны две последовательности из n целых чисел каждая. Создать новый массив из тех чисел второй последовательности, которые не входят в первую. Удалить из новой последовательности все числа, среди цифр которых есть старшая цифра минимального элемента первой последовательности, затем отсортировать полученную последовательность по убыванию методом пузырька.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении начала массива и дальнейшую работу производить с частью массива (с 1-го по $n - k$ -й элементы, где n – размер исходного массива, k – количество удаленных элементов).

23. Даны три одномерных числовых массива A , B , C из n элементов каждый. Сформировать массив K такого же размера, элементы которого вычисляются по формуле

$$k_i = \frac{a_i + b_i^2}{1 + c_i} - i,$$

где a_i , b_i , c_i , k_i – i -е элементы соответственно массива A , массива B , массива C и массива K . В массиве K удалить все отрицательные и минимальные элементы, затем полученный массив отсортировать по убыванию, используя метод выбора.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении начала массива и дальнейшую работу производить с частью массива (с 1-го по $n - k$ -й элементы, где n – размер исходного массива, k – количество удаленных элементов).

24. Даны два массива A и B из n целых чисел. Любой из них, элементы которого неупорядочены по возрастанию, скопировать в новый массив C , затем удалить из него все нулевые и отрицательные элементы и отсортировать по убыванию методом простых включений.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении начала массива и дальнейшую работу производить с частью массива (с 1-го по $n - k$ -й элементы, где n – размер исходного массива, k – количество удаленных элементов).

25. Даны два массива A и B из n целых чисел. Любой из них, в котором нет нулевых элементов и при этом положительные элементы чередуют-

ся с отрицательными, скопировать в новый массив C . Удалить из массива C все отрицательные элементы и упорядочить оставшиеся по возрастанию, используя сортировку обменом.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении начала массива и дальнейшую работу производить с частью массива (с 1-го по $n - k$ -й элементы, где n – размер исходного массива, k – количество удаленных элементов).

26. Дан массив A из n целых неотрицательных чисел. Каждый его элемент, кроме первого и последнего, нужно заменить полусуммой цифр двух его соседних элементов, затем удалить из полученного массива четные элементы и отсортировать по возрастанию, используя сортировку методом пузырька.

Указание. Удалить элемент из массива означает сдвинуть расположенные после него элементы массива на одну позицию в направлении начала массива и дальнейшую работу производить с частью массива (с 1-го по $n - k$ -й элементы, где n – размер исходного массива, k – количество удаленных элементов).

Глава 7. ДВУМЕРНЫЕ МАССИВЫ

7. 1. Задания с решениями

7.1.1. Статические двумерные массивы

1. Сформировать квадратную матрицу, в которой все элементы нулевой и последней строк, все элементы нулевого и последнего столбцов равны -1 , все остальные элементы матрицы равны 0 .

```
#include <iostream>
#include <Windows.h>
using namespace std;
const int n=10;
int main()
{
    SetConsoleOutputCP(1251);
    int a[n][n];
    for (int i=0; i<n; i++)
    {
        a[0][i]=-1;
        a[i][0]=-1;
        a[n-1][i]=-1;
        a[i][n-1]=-1;
    }
    for (int i=1; i<n-1; i++)
        for (int j=1; j<n-1; j++)
            a[i][j]=0;
    // печать результатов
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<n; j++)
            cout << a[i][j] << " ";
        cout << endl;
    }
    getchar();
    return 0;
}
```

2. Обменять местами элементы строк с номерами 2 и $n-1$ вещественной прямоугольной матрицы, в которой n строк и n столбцов.

```
#include <iostream>
#include <Windows.h>
using namespace std;
const int m=10, n=20;
```

```

int main()
{
    SetConsoleOutputCP(1251);
    float a[m][n], b;
    // ввод элементов матрицы
    for (int i=0; i<m; i++)
        for (int j=0; j<n; j++)
            cin >> a[i][j];
    for (int i=0; i<n; i++)
    {
        b=a[2][i];
        a[2][i]=a[n-1][i];
        a[n-1][i]=b;
    }
    // печать результатов
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<n; j++)
            cout << a[i][j] << " ";
        cout << endl;
    }
    getchar();
    return 0;
}

```

3. В таблице а представлены результаты 12 контрольных работ по информатике 25 первокурсников группы. Найти средний балл по информатике студентов этой группы.

```

#include <iostream>
#include <Windows.h>
using namespace std;
const int m=25, n=12;
int main()
{
    SetConsoleOutputCP(1251);

    int a[m][n];
    float Sr=0;
    // ввод элементов матрицы
    for (int i=0; i<m; i++)
        for (int j=0; j<n; j++)
            cin >> a[i][j];
    for (int i=0; i<m; i++)
        for (int j=0; j<n; j++)

```



```

        Sr+=a[i][j];
    Sr/=(m*n);
    // печать результатов
    cout << "Средний балл по информатике = " << Sr << endl;
    getchar();
    return 0;
}

```

4. Что получится при инициализации двумерного массива?

```

int a[2][3], i, j;
i=0;
while (i<2)
{
    j=0;
    while (j<3)
        a[i][j++]=3*i+j+1;
    i++;
}

```

Ответ: a[0][0]=1 a[0][1]=2 a[0][2]=3
 a[1][0]=4 a[1][1]=5 a[1][2]=6

5. Сформировать матрицу целых чисел из m строк и n столбцов. Найти максимальный из минимальных элементов каждой строки.

```

#include <iostream>
#include <Windows.h>
using namespace std;
const int M=16, N=10;
int main()
{
    SetConsoleOutputCP(1251);
    int a[M][N], i, j,max, min;
    // ввод элемнетов матрицы a
    j=0;
    while (j<M)
    {
        i=0;
        while (i<N)
        {
            cout << "введите элемент" << j << i << endl;
            cin >> a[j][i];
            i++;
        }
    }
}

```

```

        j++;
    }
    // находим минимальный элемент нулевой строки
    min=a[0][0];
    i=0;
    while (i<N)
    {
        if (a[0][i]<min) min=a[0][i];
        i++;
    }
    j=1;
    max=min;
    while (j<M)
    {
        i=0;
        min=a[j][0];
        while(i<N)
        {
            if (a[j][i]<min) min=a[j][i];
            i++;
        }
        if (min>max) max=min;
        j++;
    }
    cout << "max=" << max << endl;
    getchar();
    return 0;
}

```

6. Ввести элементы целочисленной прямоугольной матрицы, содержащей m строк, в каждой из которых n элементов. В первой из строк, содержащей максимальное количество нулевых элементов, каждый положительный элемент заменить суммой его цифр.

```

#include <iostream>
#include <Windows.h>
using namespace std;
const int M=20, N=10;
int main()
{
    SetConsoleOutputCP(1251);
    int a[M][N], b;
    // ввод элементов матрицы a
    for (int i=0; i<M; i++)
        for (int j=0; j<N; j++)

```

```

        cin >> a[i][j];
// сформирован двумерный массив
for (int i=0; i<M; i++)
{
    for (int j=0; j<N; j++)
        cout << a[i][j] << " ";
    cout << endl;
}
// поиск Nstr номера строки, содержащей максимальное
// количество нулевых элементов
int Nstr=-1, max=0;
for (int i=0; i<M; i++)
{
    int kol=0;
    for (int j=0; j<N; j++)
        if (a[i][j]==0) kol++;
    if (kol>max)
    {
        Nstr=i;
        max=kol;
    }
}
if (Nstr==-1) cout<<"В матрице нет нулевых элементов"<<endl;
else
    for (int i=0; i<N; i++)
        if (a[Nstr][i]>0)
        {
            int s=0;
            b=a[Nstr][i];
            do
            {
                s+=b%10;
                b=b/10;
            }while (b!=0);
            a[Nstr][i]=s;
        }
cout << "результат:" << endl;
for (int i=0; i<M; i++)
{
    for (int j=0; j<N; j++)
        cout << a[i][j] << " ";
    cout << endl;
}
getchar();

```

```

    return 0;
}

```

7.1.2. Двумерные массивы и указатели

1. Что получится при инициализации двумерного массива?

```

int a[2][3], i, j;
i=0;
while (i<2*3)
    *((int*)a+i++)=i+1;

```

Ответ: здесь двумерный массив представлен как одномерный; результат инициализации 1 2 3 4 5 6 (такой же, как в примере 4 раздела 1.1).

2. Каков результат инициализации матрицы при работе следующего фрагмента программы?

```

int a[2][3], i, *pt=(int*)a;
i=0;
while (i<2*3)
    *pt++=++i;

```

Ответ: предложенный фрагмент программы идентичен рассмотренному в предыдущем примере этого раздела и в примере 4 раздела 1.1.

3. Вычислить норму матрицы a , содержащей m строк, в каждой из которых n вещественных элементов по формуле

$$\max_i \sum_j |a_{i,j}|$$

(максимальное значение из суммы модулей каждой строки).

```

#include <iostream>
#include <Windows.h>
using namespace std;
const int M=10, N=20;
int main()
{
    SetConsoleOutputCP(1251);
    float a[M][N];
    // ввод элемнетов матрицы a
    for (int i=0; i<M; i++)
        for (int j=0; j<N; j++)
            cin >> *((a+j)+i);
    float s, max=0;
    for (int i=0; i<M; i++)
    {

```

```

        s=0;
        for (int j=0; j<N; j++)
            s+=abs(*(a+i+j));
        if (s>max) max=s;
    }
    cout << "Норма введенной матрицы = " << s;
    getchar();
    return 0;
}

```

4. Пусть определен двумерный массив `int m[3][4]`. Каким из предложенных способов можно обратиться к элементу массива `m[2][3]`?

- 1) `*(m[2]+3)` 2) `*(*(m+2)+3)`
- 3) `3m[2]` 4) `3[2[m]]`

Ответ: все четыре способа эквивалентны.

5. Сформулировать, что вычисляет следующий фрагмент программы

```

int a[4][5]={1,2,3,4,5,6,7,8,9};
int *p=&a[0][0];
int s=0;
for (int i=0; i<4; i++)
    s+=*p++;

```

Ответ: вычисляется сумма первых четырех элементов нулевой строки массива *a*.

6. Что будет напечатано в результате выполнения следующего фрагмента программы

```

#include <iostream>
#include <Windows.h>
using namespace std;
int main()
{
    SetConsoleOutputCP(1251);
    int a[4][5], *p;
    for (int i=0; i<4; i++)
    {
        p=a[i];
        for (int j=0; j<5; j++)
        {
            *(p+j)=5*i+j+1;
            cout << a[i][j] << " ";
        }
        cout << "\n";
    }
}

```

```

    }
    getch();
    return 0;
}

```

Ответ:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

7. Ввести вещественные элементы квадратной матрицы из 10 строк. Вычислить скалярное произведение строки, в которой находится максимальный элемент, на столбец, в котором расположен минимальный элемент матрицы. Считать, что матрица состоит из различных вещественных чисел.

```

#include <iostream>
#include <Windows.h>
using namespace std;
const int N=10;
int main()
{
    SetConsoleOutputCP(1251);
    int a[N][N], max, min, k, p;
    // ввод элементов матрицы a
    int i=0;
    while (i<N*N)
        cin >> (*(a+i++));
    cout << "Исходный массив:\n";
    // печать в строку как одномерный массив
    i=0;
    while (i<N*N)
        cout << (*(a+i++));
    cout << endl;
    // обработка матрицы по строкам
    max=min=**a;
    k=p=0;
    for (int i=0; i<N; i++)
        for (int j=0; j<N; j++)
        {
            if (*(a+i)+j)>max)
            {
                max=*(a+i)+j;
                k=i;
            }
        }
}

```

```

        if (*(a+i)+j)<min)
        {
            min=*(a+i)+j;
            p=j;
        }
    }
    // вчисление скалярного произведения
    // строки k на столбец p
    int s=0;
    for (int i=0; i<N; i++)
        s+=(*(a+k)+i)*(*(a+i)+p));
    cout << "s=" << s << endl;
    getchar();
    return 0;
}

```

7.1.3. Динамические двумерные массивы

1. Дана программа, в которой формируются значения элементов двумерного массива, содержащего n строк и n столбцов.

```

#include <iostream>
#include <Windows.h>
using namespace std;
int main()
{
    SetConsoleOutputCP(1251);
    int n, k=1;
    cin >> n;
    int **a=new int*[n];
    for (int i=0; i<n; i++)
        a[i]=new int [n];
    for (int i=1; i<n; i++)
    {
        for (int j=1; j<n; j++)
            *(a+i)+j)=k;
        k=-k;
    }
    for (int i=0; i<n; i++)
    {
        *(a+0)+i)=i;
        *(a+i)+0)=i;
    }
    for (int i=0; i<n; i++)
        delete [] *(a+i);
}

```

```

delete []a;
getchar();
return 0;
}

```

A. Чему равен элемент двумерного массива $a[5][4]$, если введено $n > 7$?

B. Чему равна сумма элементов строки с номером 6, если введено $n = 8$?

C. Чему равен максимальный элемент столбца с номером 4?

D. Сколько столбцов матрицы a состоят из положительных элементов, если введено $n = 15$?

E. Сколько строк не содержит отрицательных элементов, если введено $n = 8$?

F. Сколько единиц находится в третьем столбце, если введено $n = 10$?

G. Пусть введено $n = 6$. Чему равна сумма элементов главной диагонали матрицы?

H. Пусть введено $n = 10$. Чему равен максимальный элемент матрицы?

I. Пусть введено $n = 9$. Чему равна сумма максимальных элементов нулевой строки и нулевого столбца?

Ответы:

A	B	C	D	E	F	G	H	I
1	-1	1	1	5	5	1	9	16

2. Дана матрица вещественных чисел. Составить вектор p из произведений элементов строк этой матрицы.

```

#include <iostream>
#include <Windows.h>
using namespace std;
int main()
{
    SetConsoleOutputCP(1251);
    int nstr, nstb;
    cout << "Введите количество строк и столбцов" << endl;
    cin >> nstr >> nstb;
    float **a=new float *[nstr];
    for (int i=0; i<nstr; i++)
        a[i]=new float [nstb];
    float *p=new float [nstr];
    for (int i=0; i<nstr; i++)
        for (int j=0; j<nstb; j++)
            cin >> (*(a+i)+j);
    for (int i=0; i<nstr; i++)
    {

```



```

        p[i]=1;
        for (int j=0; j<nstb; j++)
            if (*(a+i+j)==0)
            {
                p[i]=0;
                break;
            }
            else
                p[i]=*(a+i+j);
    }
    cout << "Полученный массив" << endl;
    for (int i=0; i<nstr; i++)
        cout << p[i] << " ";

    for (int i=0; i<nstr; i++)
        delete [] *(a+i);
    delete []a;
    delete []p;

    getchar();
    return 0;
}

```

2. Сформировать двумерный массив из m строк. В каждой строке этого массива находятся вещественные координаты трех точек плоскости

x_1	y_1	x_2	y_2	x_3	y_3
координаты первой точки		координаты второй точки		координаты третьей точки	

Найти количество строк этого массива, содержащих координаты тех точек, которые могут быть вершинами треугольника

```

#include <iostream>
#include <Windows.h>
using namespace std;
int main()
{
    SetConsoleOutputCP(1251);
    int m,n=6,k=0;
    cout << "Введите количество строк " << endl;
    cin >> m;
    float **A=new float *[m];
    for (int i=0; i<m; i++)
        A[i]=new float [n];
    float a,b,c,x1,x2,x3,y1,y2,y3;
    for (int i=0; i<m; i++)

```

```

        for (int j=0; j<n; j++)
            cin >> (*(A+i)+j);
cout << "Сформированная матрица" << endl;
for (int i=0; i<m; i++)
{
    for (int j=0; j<n; j++)
        cout << (*(A+i)+j) << " ";
    cout << endl;
}
for (int i=0; i<m; i++)
{
    x1=(*(A+i)+0);
    y1=(*(A+i)+1);
    x2=(*(A+i)+2);
    y2=(*(A+i)+3);
    x3=(*(A+i)+4);
    y3=(*(A+i)+5);
    a=sqrtf((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
    b=sqrtf((x1-x3)*(x1-x3)+(y1-y3)*(y1-y3));
    c=sqrtf((x3-x2)*(x3-x2)+(y3-y2)*(y3-y2));
    if ((a+b)>c && (a+c>b) && (b+c>a)) k++;
}
cout << "Количество треугольников =" << k << endl;

for (int i=0; i<n; i++)
    delete [] *(A+i);
delete []A;

getchar();
return 0;
}

```

7.2. Тренировочные задания

1. В поликлинику поступают следующие сведения: номер дома, число жильцов, число детей, число приезжих, число пенсионеров – по каждому из семи обслуживаемых домов. Сформировать прямоугольную таблицу, содержащую сведения обо всех домах.

2. Поменять местами элементы главной и побочной диагонали квадратной таблицы.

3. Вычислить суммы элементов в каждом столбце прямоугольной таблицы.

4. Найти суммы положительных элементов в каждой строке прямоугольной таблицы.

5. Найти сумму элементов, являющихся минимальными в каждом столбце прямоугольной таблицы.
6. Найти произведение элементов, являющихся максимальными в каждой строке прямоугольной таблицы.
7. Поменять местами элементы первой строки и третьего столбца квадратной таблицы.
8. На место каждого отрицательного элемента третьего столбца прямоугольной таблицы поставить сумму его индексов.
9. Поставить нули на место каждого элемента первой строки и первого столбца, последней строки и последнего столбца прямоугольной таблицы.
10. В каждой строке прямоугольной таблицы поменять местами первый минимальный и последний максимальный элементы.
11. В прямоугольной таблице подсчитать число положительных, отрицательных и нулевых элементов в каждой строке.
12. Составить таблицу Пифагора (квадратную таблицу, состоящую из 10 строк и 10 столбцов, каждый элемент которой определяется формулой $A_{ij} = i \cdot j$).
13. Вычислить сумму диагональных элементов таблицы, содержащей 10 строк и 10 столбцов.
14. Транспонировать квадратную матрицу порядка n , где n вводится.
15. В каждой из n строк прямоугольной таблицы указано содержание белков, жиров и углеводов в 1 г соответствующего продукта. В другой, линейной, таблице каждый элемент указывает количество соответствующего продукта в рационе. Сколько белков, жиров и углеводов содержится в рационе? *Указание.* Исходная таблица содержит n строк и 3 столбца, вторая таблица содержит n элементов, искомая – 3 элемента.
16. Первая строка таблицы содержит сведения о количестве жильцов, проживающих на первом этаже в каждом из 10 подъездов, вторая – те же сведения о жильцах второго этажа и т.д., всего n строк. На каком этаже и в каком подъезде проживает наибольшее число жильцов?
17. По таблице, описанной в задаче 16, найдите подъезд с максимальным количеством жильцов.
18. По таблице, описанной в задаче 16, определите, есть ли в доме лестничная площадка, в квартирах которой проживает более k жильцов, где k определяется вводом.
19. Таблица содержит 10 столбцов, в каждом из которых указаны номер продукта, содержание в 1 г. этого продукта белков, жиров и углеводов. В другой таблице указано, сколько граммов каждого продукта требуется для приготовления блюда. Определить содержание белков, жиров и углеводов в блюде. *Указание.* Задача сводится к нахождению произведения матрицы на вектор.

20. По таблице, описанной в задаче 19, определить номер продукта, который содержит наибольшее количество белков.

21. По таблице, описанной в задаче 19, определить, есть ли продукт, содержащий одновременно максимальное количество белков и минимальное количество жиров.

22. Сформировать целочисленную матрицу A из m строк, в каждой из которых n элементов. Создать двумерный массив B из тех строк матрицы A , которые удовлетворяют условию $A[i][0] > c$, где c – заданное число, определяемое вводом.

23. В прямоугольной таблице собраны отметки 30 учеников класса по 10 предметам.

А. Найти средний балл учеников класса.

В. Если ученик имеет только одну двойку, исправить ее на тройку.

С. Сколько в классе неуспевающих учеников, у которых нет ни одной пятерки.

Д. Составить новую таблицу, в которую поместить средний балл каждого ученика и отметки по информатике (третий столбец данной прямоугольной таблицы).

Е. Обменять местами отметки любых двух учеников: одного с максимальным средним баллом, другого с минимальным среди успевающих учеников.

24. Прямоугольная таблица содержит зарплаты M сотрудников отдела предприятия за 12 месяцев прошлого года.

А. Какова средняя зарплата всех сотрудников в феврале месяце?

В. Какова максимальная зарплата в сентябре месяце?

С. Переставить минимальную и максимальную зарплаты в мае месяце.

Д. Изменить на противоположный порядок следования элементов последней строки таблицы.

Е. В каком месяце сотрудники получили суммарную максимальную зарплату?

Ф. Найти номер сотрудника и месяц, которым соответствует максимальная зарплата.

Г. Все зарплаты сотрудников с нечетными номерами заменить на среднюю зарплату в месяце, если эти зарплаты отличны от нуля.

Н. Найти среднюю зарплату всех сотрудников отдела за прошлый год.

І. Сколько сотрудников имеют зарплату, выше средней?

Ж. Сколько сотрудников имеют зарплату ниже прожиточного уровня, который равен P рублей в месяц?

К. Обменять местами ненулевые зарплаты первого и последнего сотрудников.

Л. Обменять местами зарплаты всех сотрудников за март и сентябрь месяцы, если среди этих зарплат нет максимальных за весь прошлый год.

М. Сколько сотрудников имеют зарплату выше прожиточного уровня, который равен P рублей в месяц, но меньше средней за весь год?

25. Таблица содержит результаты расхода электроэнергии N предприятиями города за каждый месяц года.

А. В каком месяце было израсходовано меньше всего электроэнергии?

В. Найти суммарный расход электроэнергии каждого предприятия за год.

С. Сколько предприятий превысили минимальный расход суммарной электроэнергии за год в M киловатт?

Д. Найти расход энергии за год всеми предприятиями города.

26. В каждой строке прямоугольной таблицы изменить на противоположный порядок следования элементов.

27. К каждому отрицательному элементу прямоугольной таблицы добавить минимальный, отличный от нуля элемент всей таблицы.

28. В строке, содержащей максимальный элемент, найти произведение ненулевых элементов, расположенных на четных местах.

29. В линейной таблице размещена цена 50 молочных продуктов магазина. Прямоугольная таблица содержит сведения о количестве проданных молочных продуктов каждого из 50 видов в январе месяце.

А. Какова выручка магазина за январь месяц?

В. В какой день выручка магазина была минимальной?

30. Дана таблица из m строк, в каждой из которых n элементов. Поменять местами в каждой строке первый положительный элемент с первым максимальным.

31. Составить алгоритм формирования таблицы из n строк следующего вида

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ -1 & -2 & -3 & -4 & -5 & -6 & -7 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ -7 & -6 & -5 & -4 & -3 & -2 & -1 \end{pmatrix}.$$

32. Дана прямоугольная таблица вещественных чисел.

А. В строке, сумма элементов которой минимальна, найти среднее арифметическое положительных элементов, расположенных между двумя ее наибольшими элементами, включая сами эти элементы.

В. Составить линейную таблицу из количества минимальных элементов каждого столбца и строки.

33. Сформировать прямоугольную таблицу вида

$$\text{a) } \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 26 & 27 & 28 & 29 & 30 \end{pmatrix};$$

$$\text{b) } \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & -1 \end{pmatrix};$$

$$\text{c) } \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 0 & 0 & 0 & 0 & 5 \\ 3 & 0 & 0 & 0 & 0 & 4 \\ 4 & 0 & 0 & 0 & 0 & 3 \\ 5 & 0 & 0 & 0 & 0 & 2 \\ 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix};$$

$$\text{d) } \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 11 \\ -2 & -2 & -2 & -2 & -2 & 12 \\ 3 & 3 & 3 & 3 & 3 & 13 \\ -4 & -4 & -4 & -4 & -4 & 14 \\ 5 & 5 & 5 & 5 & 5 & 15 \\ -6 & -6 & -6 & -6 & -6 & 16 \end{pmatrix}.$$

34. Заданы координаты центра круга (a,b) , его радиус R , таблица C , содержащая координаты N вещественных чисел.

А. Сколько точек из таблицы C попадает внутрь круга радиуса R с центром (a,b) ?

В. Найти координаты первой точки из таблицы C , которая не попадает внутрь заданного круга.

С. Найти номер точки таблицы C , наиболее удаленной от центра круга.

35. Дана квадратная таблица, содержащая $N \times N$ вещественных чисел.

А. Обменять местами элементы первой строки и последнего столбца.

В. Обменять местами элементы диагоналей.

С. Найти минимальный элемент столбца, сумма элементов которого максимальна.

Д. Образовать линейную таблицу из средних арифметических каждой строки.

Е. Каждый элемент последнего столбца увеличить на значение максимального элемента соответствующей строки.

Ф. Каждый максимальный элемент таблицы, не являющийся диагональным, заменить суммой его индексов.

Г. Составить линейную таблицу, каждый элемент которой равен количеству нулевых элементов строки.

Н. Каждый положительный элемент таблицы, который расположен не в первом и последнем столбцах, не на первой и последней строках, увеличить на значение среднего арифметического всей таблицы.

36. В вещественной матрице поменять местами первую строку, содержащую минимальный элемент и последнюю строку, содержащую максимальный элемент.

37. Дана целочисленная матрица из m строк и n столбцов. В строках, содержащих наибольшее количество положительных элементов, все четные элементы увеличить на младшую цифру первого элемента этой строки.

38. Дана квадратная вещественная матрица. Заменить элементы главной диагонали заданной матрицы на 1, если в соответствующей строке все элементы положительные и на 0 в противном случае.

39. Дана целочисленная матрица из m строк и n столбцов. Найти и напечатать номера тех строк матрицы, произведение элементов которых, совпадает с суммой элементов последней строки.

40. Получите вещественную матрицу A размером 7×7 , первая строка которой задается формулой $a_{1j} = 2j + 3$ ($j = 1, \dots, 7$), вторая строка – формулой $a_{2j} = j - \frac{3}{2 + 1/j}$ ($j = 1, \dots, 7$), а каждая следующая строка есть сумма двух предыдущих.

41. Пусть дано натуральное число n . Выясните, сколько положительных элементов содержит матрица A размером $n \times n$ при выполнении перечисленных ниже условий:

A. $a_{ij} = \sin(i + j / 2)$;

B. $a_{ij} = \cos(i^2 + n \cdot j)$;

C. $a_{ij} = \sin\left(\frac{i^2 - j^2}{n}\right)$.

42. Пусть дана вещественная матрица размером $n \times m$, в которой не все элементы равны нулю. Получите новую матрицу путем деления всех элементов данной матрицы на ее наибольший по модулю элемент.

43. Пусть даны натуральное число m , целые числа a_1, \dots, a_m и целочисленная квадратная матрица порядка m . Строку с номером i назовем отмеченной, если $a_i > 0$, и неотмеченной – в противном случае:

A. Все элементы, расположенные в отмеченных строках матрицы, преобразуйте по правилу: отрицательные элементы заменить на -1 , положительные на 1 , а нулевые оставить без изменения.

B. Подсчитайте число отрицательных элементов матрицы, расположенных в отмеченных строках.

44. Все элементы с наибольшим значением в данной целочисленной квадратной матрице порядка 10 заменить нулями, если эти элементы не принадлежат первой или последней строкам.

45. Пусть дана вещественная матрица размером $n \times n$. Найдите среднее арифметическое наибольшего и наименьшего значений ее элементов, расположенных ниже главной диагонали.

46. Пусть дана вещественная матрица размером $n \times n$. Найдите значение наибольшего по модулю элемента матрицы и укажите его местоположение в матрице.

47. В данной квадратной целочисленной матрице порядка n укажите индексы всех элементов с наибольшим значением, не принадлежащих главной и побочной диагоналям.

48. Пусть дана вещественная матрица размером $n \times m$, все элементы которой различны. В каждой строке выберите элемент с наименьшим значением, затем среди этих чисел выберите наибольшее. Укажите индексы найденного элемента.

49. Пусть дана целочисленная квадратная матрица порядка n . Найдите наименьшее из значений элементов столбца, который обладает наибольшей суммой модулей элементов. Если таких столбцов несколько, надо взять первый из них.

50. Пусть дана целочисленная квадратная матрица порядка n . Найдите номера строк:

- а) все элементы которых нули;
- б) элементы в каждой из которых одинаковы;
- с) все элементы которых четны;
- д) элементы каждой из которых образуют монотонную последовательность (монотонно убывающую или монотонно возрастающую);
- е) элементы которых образуют симметричные последовательности.

51. Пусть дана вещественная квадратная матрица порядка n . В строках с отрицательным элементом на главной диагонали найдите:

- а) сумму элементов;
- б) наибольший из элементов.

52. Пусть дана вещественная квадратная матрица порядка n . Отноительно элементов, расположенных в строках, начинающихся с отрицательного элемента, найдите суммы тех элементов, которые расположены соответственно ниже, выше и на главной диагонали.

53. Пусть дана вещественная квадратная матрица порядка n . Получите целочисленную квадратную матрицу того же порядка, в которой элемент равен единице, если соответствующий ему элемент исходной матрицы больше элемента, расположенного в его строке на главной диагонали, и равен нулю в противном случае.

54. Пусть дана вещественная квадратная матрица порядка n . Вычислите сумму тех из ее элементов, расположенных на главной диагонали и выше ее, которые превосходят по величине все элементы, расположенные

ниже главной диагонали. Если на главной диагонали и выше ее нет элементов с указанным свойством, ответом должно служить сообщение об этом.

55. Пусть A, B, C – квадратные матрицы порядка n из вещественных чисел, x, y – одномерные массивы из n вещественных чисел. Вычислить:

- a) $C = A + B$;
- b) $y = A \cdot x$;
- c) $C = A \cdot B$;
- d) $B = B^T$ (транспонировать).

56. Даны два двумерных массива A и B , в каждом из которых m строк и n столбцов. Присвоить переменной t значение true, если массивы A и B равны и false в противном случае.

57. Пусть даны натуральное число n и элементы вещественной квадратной матрицы A порядка $m < 7$. Вычислить n -ую степень этой матрицы ($A_1 = A$, $A_2 = A \cdot A_1$, $A_3 = A \cdot A_2$ и т. д.).

58. Дан вектор x из n целых чисел и квадратная матрица A порядка n . Составить программы для решения следующих задач.

А. Строки матрицы A заменить вектором x .

В. Элементы столбцов с четными номерами увеличить на минимальный элемент вектора x .

С. Строки матрицы A , в которых есть хотя бы один четный элемент заменить элементами вектора x .

Д. Если в векторе x более двух минимальных элементов, обменять местами строки матрицы A с четными и нечетными номерами.

Е. Каждый минимальный и максимальный элементы вектора x заменить на минимальный и максимальный элементы матрицы A .

59. В двумерном массиве A заданы координаты m точек плоскости.

А. Найти наибольшее расстояние между этими точками.

В. Найти координаты любых двух точек, расстояние между которыми минимально.

С. Найти количество точек, расположенных выше оси x .

60. Пусть дана вещественная матрица порядка $m \times n$, все элементы которой различны. Переставить ее строки и столбцы так, чтобы ее наибольший элемент оказался в левом верхнем углу.

61. Пусть дана вещественная матрица порядка $m \times n$, все элементы которой различны. Переставить ее строки и столбцы так, чтобы ее наименьший элемент оказался в правом нижнем углу.

62. Определить, является ли заданная квадратная матрица порядка n магическим квадратом, то есть такой, в которой суммы элементов во всех строках и столбцах одинаковы.

63. Дан двумерный массив из m строк, в каждой из которых записаны координаты трех точек плоскости:

а) найти количество строк, в которых записаны координаты вершин треугольника;

б) составить одномерный массив из площадей треугольников, координаты вершин которых записаны в строках матрицы;

с) если строка матрицы содержит координаты вершин треугольника, поместить длину большей его стороны в одномерный массив;

д) если строка матрицы содержит координаты вершин треугольника, поместить длину меньшей стороны в одномерный массив;

е) если строка матрицы содержит координаты вершин прямоугольного треугольника, поместить длины его катетов в одномерный массив.

7.3. Задания для самостоятельной работы

7.3.1. Задание 1

1. Двумерный массив A содержит m строк, в каждой из которых n целых чисел.

В нулевой столбец занесены номера участников олимпиады. Остальные столбцы содержат баллы (от 0 до 100) за выполнение олимпиадных заданий.

Составить одномерный массив из номеров участников, у которых не было ни одного нулевого балла.

2. Двумерный массив A содержит m строк, в каждой из которых n целых чисел.

В нулевой столбец занесены номера участников олимпиады. Остальные столбцы содержат баллы (от 0 до 100) за выполнение олимпиадных заданий.

Составить одномерный массив из номеров участников, средний балл которых за выполнение всех заданий превышает средний балл всех участников.

3. Двумерный массив A содержит m строк, в каждой из которых n целых чисел.

В нулевой столбец занесены номера участников олимпиады. Остальные столбцы содержат баллы (от 0 до 100) за выполнение олимпиадных заданий.

Составить одномерный массив из номеров участников, суммарный балл которых ниже максимального суммарного балла, но выше минимального.

4. Двумерный массив A содержит m строк, в каждой из которых n целых чисел.

В нулевой столбец занесены номера участников олимпиады. Остальные столбцы содержат баллы (от 0 до 100) за выполнение олимпиадных заданий.

Составить одномерный массив из номеров заданий, которые выполнили наибольшее число участников.

5. Двумерный массив A содержит m строк, в каждой из которых n целых чисел.

В нулевой столбец занесены номера участников олимпиады. Остальные столбцы содержат баллы (от 0 до 100) за выполнение олимпиадных заданий.

Составить одномерный массив из номеров заданий, которые не выполнили более половины участников.

6. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.1).

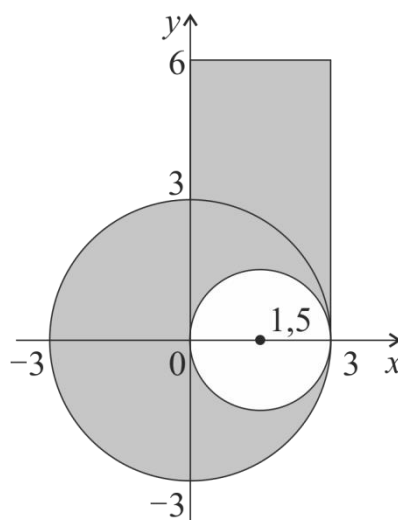


Рис. 7.1

7. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.2).

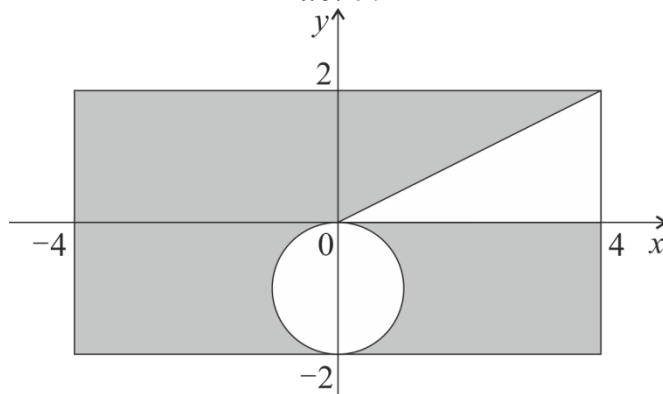


Рис. 7.2

8. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.3).

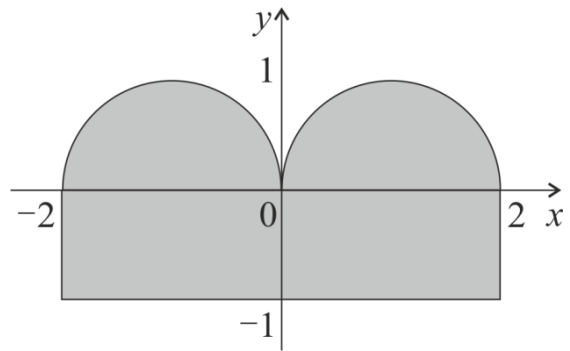


Рис. 7.3

9. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.4).

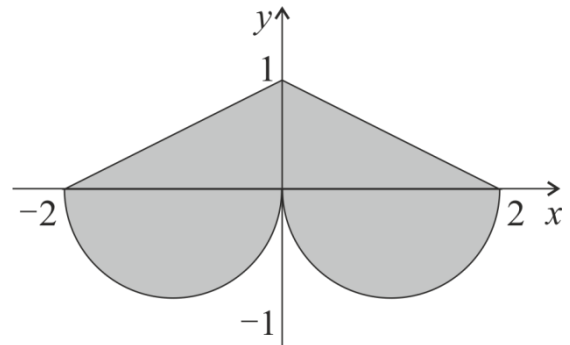


Рис. 7.4

10. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.5).

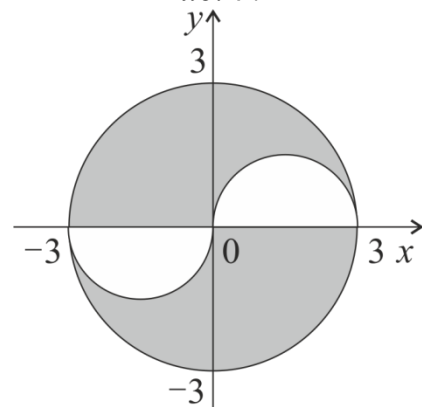


Рис. 7.5

11. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.6).

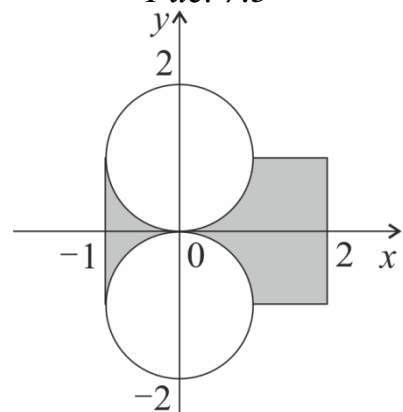


Рис. 7.6

12. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.7).

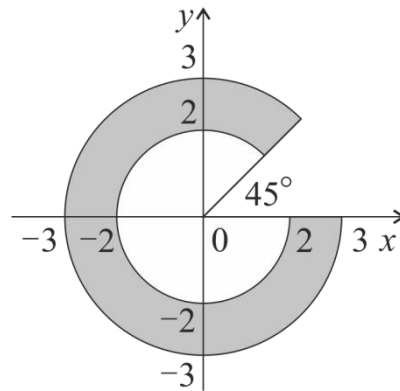


Рис. 7.7

13. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.8).

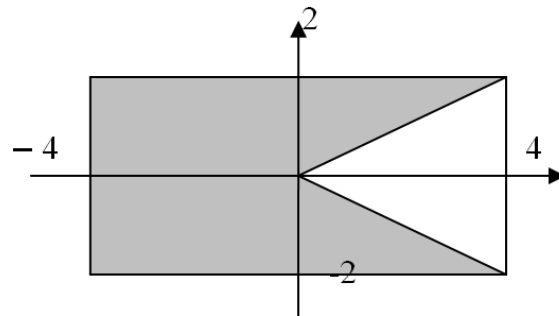


Рис. 7.8

14. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.9).

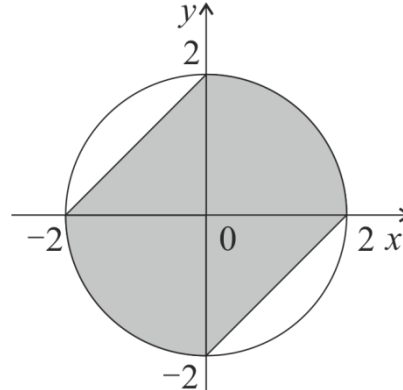


Рис. 7.9

15. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.10).

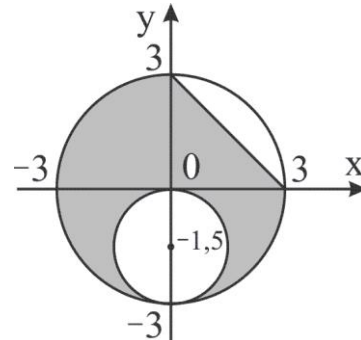


Рис. 7.10

16. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.11).

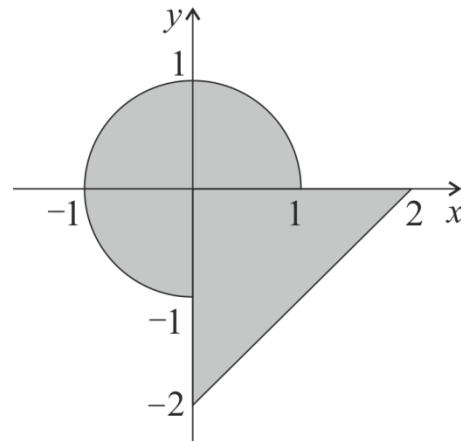


Рис. 7.11

17. В двумерном массиве заданы координаты m точек плоскости. Составить одномерный массив из номеров точек, которые попадают внутрь заштрихованной области (рис. 7.12).

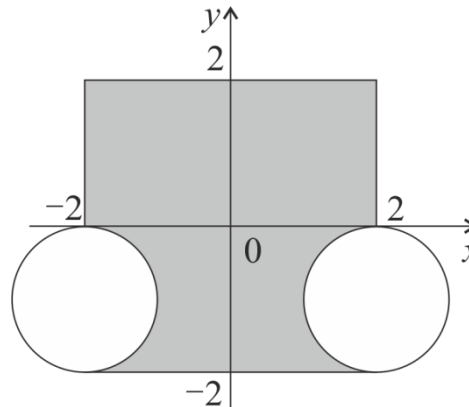


Рис. 7.12

18. Даны две таблицы A и B одинаковой размерности $m \times n$. Каждая строка таблицы A содержит количество различных изделий, выпущенных определенным цехом завода. Каждый цех выпускает n различных изделий. Структура таблицы B идентична таблице A , но содержит количество изделий, которое надо выпустить по плану. Номер строки соответствует номеру цеха завода.

Составить одномерный массив из номеров цехов, которые перевыполнили план по общему количеству выпущенных изделий.

19. Даны две таблицы A и B одинаковой размерности $m \times n$. Каждая строка таблицы A содержит количество различных изделий, выпущенных определенным цехом завода. Каждый цех выпускает n различных изделий. Структура таблицы B идентична таблице A , но содержит количество изделий, которое надо выпустить по плану. Номер строки соответствует номеру цеха завода.

Составить одномерный массив из номеров цехов, которые недовыполнили план более чем на 10 % по общему количеству выпущенных изделий.

20. Даны две таблицы A и B одинаковой размерности $m \times n$. Каждая строка таблицы A содержит количество различных изделий, выпущенных определенным цехом завода. Каждый цех выпускает n различных изделий. Структура таблицы B идентична таблице A , но содержит количество изде-

лий, которое надо выпустить по плану. Номер строки соответствует номеру цеха завода.

Составить одномерный массив из номеров изделий, которые не выпустил хотя бы один цех.

7.3.2. Задание 2

1. Результаты забега в массовом кроссе представлены целочисленной матрицей $A(n, 4)$. Здесь n – число участников, A_{i0}, A_{i1} – момент старта i -го участника в минутах и секундах, A_{i2}, A_{i3} – момент финиша, $i = 1, 2, \dots, n$. За один просмотр матрицы найти номера трех призеров.

2. На клеточном поле размером $n \times n$ представлен результат игры двух студентов в крестики-нолики: один рисовал крестики, второй – нолики. Проверить, кто из них выиграл. Считается, что выиграл первый (второй), если на поле найдется по горизонтали, вертикали или диагонали цепочка, состоящая подряд из n крестиков (ноликов).

3. Дан двумерный массив целых чисел. Вставить в него

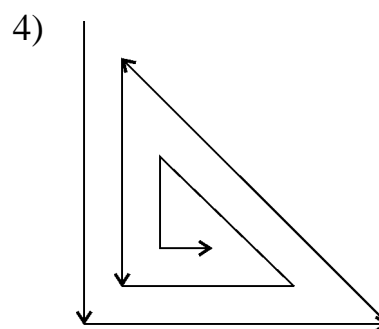
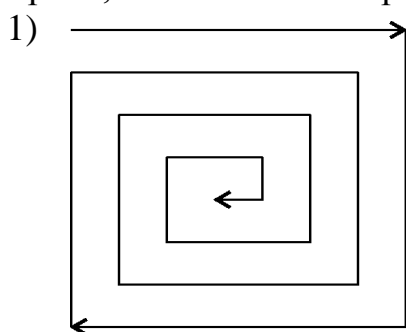
а) строку из нулей между всеми строками, в которых количество положительных элементов равно количеству отрицательных;

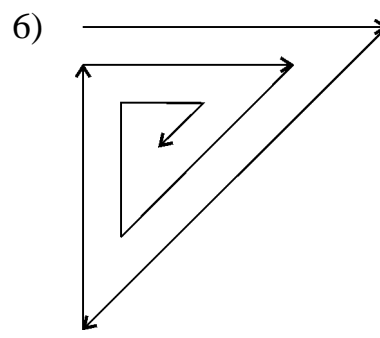
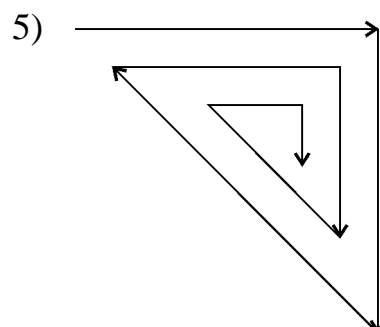
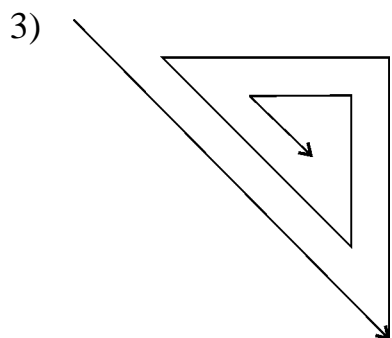
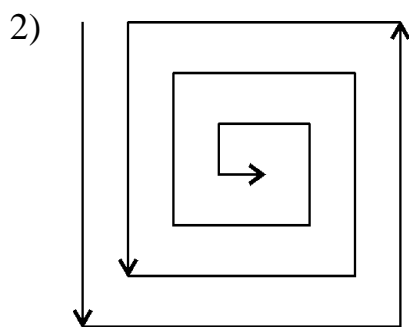
б) столбец, каждый элемент которого равен полусумме индексов первого минимального элемента данной матрицы, между всеми столбцами, в которых количество положительных элементов превышает количество отрицательных;

с) строку из минимальных элементов каждого столбца перед каждой строкой, в которой нет ни одного нуля.

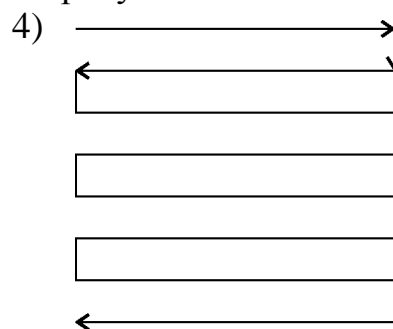
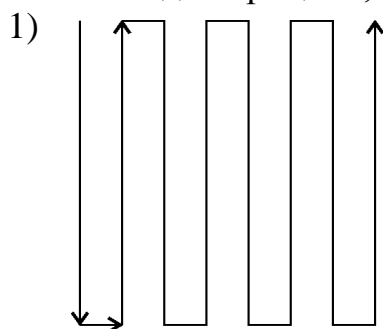
д) столбец из первых отрицательных элементов каждой строки перед каждым столбцом, в котором совпадают первый и последний элементы.

4. Дана квадратная матрица A порядка $n \times n$. Составить одномерный массив из положительных элементов матрицы A , выполнив обход матрицы по спирали, как показано на рисунках





5. Дана квадратная матрица A порядка $n \times n$. Составить одномерный массив из отрицательных элементов матрицы A , больших ее минимального, выполнив обход матрицы A , как показано на рисунках



6. Матрица A содержит m строк, в каждой из которых n элементов. Первый элемент каждой строки – шифр некоторой детали, остальные элементы – числовые характеристики этой детали. Удалить из матрицы строки с совпадающими шифрами и несовпадающими характеристиками.

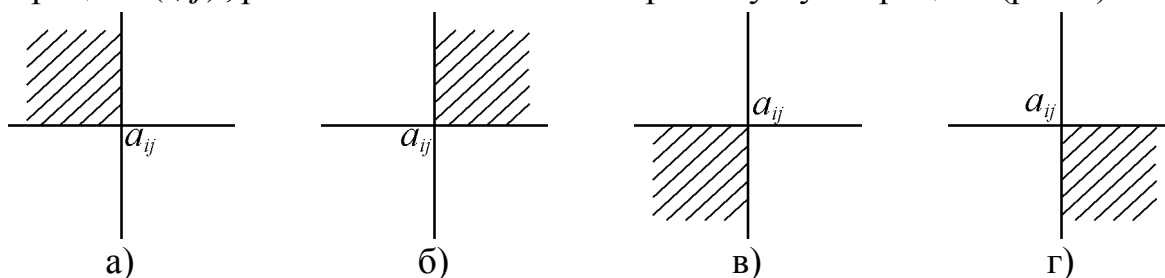
7. На плоскости заданы n материальных точек и масса каждой точки. С некоторого момента точка с наименьшей массой исчезает, передавая свою массу ближайшей к ней точке. Так продолжается до тех пор, пока останется только одна точка. Составить программу, реализующую этот процесс и найти оставшуюся точку.

8. Просуммировать элементы квадратной матрицы по каждой из линий, параллельных главной диагонали. Из полученных сумм составить одномерный массив.

9. Матрицу из m строк и n столбцов заполнить натуральными числами от 1 до $m \cdot n$ по спирали, начинающейся в левом верхнем углу и закрученной по часовой стрелке.

10. Прямоугольная матрица состоит из нулей и единиц. Найти в ней квадрат заданного размера (квадратную подматрицу), состоящий целиком из нулей. Составить одномерный массив из восьми элементов – индексов угловых точек квадрата.

11. В матрице вещественных чисел $A(m,n)$, содержащей m строк и n столбцов, каждый элемент a_{ij} заменить минимальным среди элементов подматрицы $A(i, j)$, расположенной в левом верхнем углу матрицы A (рис. а).



12. Каждый положительный элемент данной матрицы A заменить суммой элементов, расположенных в правом верхнем углу этой матрицы (рис. б).

13. Каждый нулевой элемент данной матрицы A заменить максимальным из элементов, расположенных в левом нижнем углу этой матрицы (рис. в).

14. Каждый элемент данной матрицы A заменить средним арифметическим положительных элементов, расположенных в правом нижнем углу этой матрицы (рис. г).

15. В матрице из m строк и n столбцов каждый элемент кроме граничных заменить суммой непосредственно примыкающих к нему элементов по вертикали, горизонтали и диагоналям.

16. Дана матрица из 12 строк, в каждой из которых n целых чисел. Переставить строки так, чтобы они располагались следующим образом: первая, двенадцатая, вторая, одиннадцатая, ..., пятая, восьмая, шестая, седьмая.

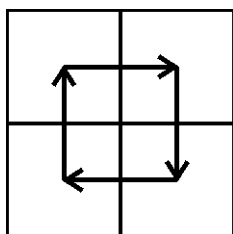
17. Дан двумерный массив из m строк и 18 столбцов. Переставить столбцы так, чтобы они располагались следующим образом: первый, восемнадцатый, второй, семнадцатый, ..., восьмой, одиннадцатый, девятый, десятый.

18. Построить квадратную матрицу порядка n по заданному образцу

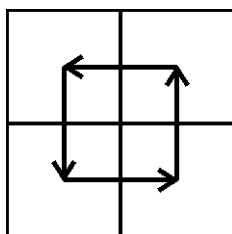
$$\begin{pmatrix} 1 & 2 & 3 & \cdots & n-2 & n-1 & n \\ 2 & 1 & 2 & \cdots & n-2 & n-2 & n-1 \\ 3 & 2 & 1 & \cdots & n-4 & n-3 & n-2 \\ \vdots & & & & & & \\ n-1 & n-2 & n-3 & \cdots & 2 & 1 & 2 \\ n & n-1 & n-2 & \cdots & 3 & 2 & 1 \end{pmatrix}$$

19. Матрицу A заполнить следующим образом. Для данных k и l элементу $A[k][l]$ присвоить значение 1. Элементам, окаймляющим его (соседним с ним по вертикали, горизонтали и диагоналям) присвоить значение 2. Элементам следующего окаймления присвоить значение 3 и так далее до заполнения всей матрицы.

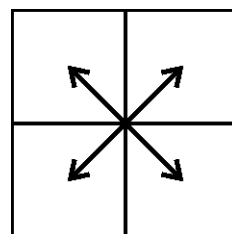
20. Квадратная матрица порядка $2n$ состоит из четырех блоков. Написать программу, которая формирует новую матрицу, переставляя блоки исходной согласно схемам



а)



б)



в)

21. В двумерный массив A записаны оценки 25 первокурсников дневного отделения факультета ПММ по алгебре, информатике, истории, математическому анализу, дискретной математике. В одномерном массиве B хранятся фамилии первокурсников. Изменить массивы A и B так, чтобы внести на место S (целое число, задаваемое вводом) фамилию и оценки студента, переведенного с вечернего отделения, а все остальные строки матрицы A и массива B сдвинуть на одну позицию вниз.

7.3.3. Задание 3

Сформировать двумерный массив A случайных целых чисел из m строк и n столбцов.

1. Если среди элементов сформированного массива нет ни одного нулевого элемента, упорядочить строки массива A в порядке возрастания средних арифметических каждой строки.

2. Упорядочить строки массива A в порядке убывания сумм цифр первого элемента каждой строки.

3. Упорядочить строки матрицы A в порядке возрастания минимальных элементов каждой ее строки все строки матрицы A .

4. Упорядочить, в порядке убывания максимальных среди элементов с нечетными номерами в каждой строке, все строки матрицы A .
5. Упорядочить, в порядке возрастания элементов главной диагонали каждой строки, все строки матрицы A , в которой $m = n$.
6. Если максимальный элемент матрицы A не находится на побочной диагонали, упорядочить, в порядке убывания элементов побочной диагонали, все строки матрицы A с четными номерами, считая $m = n$.
7. Упорядочить, в порядке возрастания полусуммы элементов главной и побочной диагонали каждой строки, все строки матрицы A , считая $m = n$.
8. Если среди цифр максимального элемента матрицы A нет ни одной цифры 5, упорядочить все столбцы матрицы в порядке убывания последних элементов каждого столбца.
9. Если минимальный элемент матрицы A находится выше главной диагонали, упорядочить столбцы матрицы A в порядке убывания сумм элементов каждого столбца, считая $m = n$.
10. Если в матрице A больше, чем один максимальный элемент, упорядочить столбцы матрицы в порядке возрастания сумм цифр первого и последнего элементов столбца.
11. Если сумма элементов, расположенных выше главной диагонали, больше суммы элементов, расположенных ниже главной диагонали, упорядочить все строки матрицы A в порядке убывания минимальных элементов каждой строки матрицы A , в которой $m = n$.
12. Упорядочить строки матрицы A в порядке убывания элементов той строки, в которой находится минимальный элемент всей матрицы, если он является единственным минимальным элементом матрицы A .
13. Если произведение элементов нулевой и последней строк больше произведения элементов первого и последнего столбцов, упорядочить столбцы матрицы в порядке убывания сумм старшей и младшей цифр последнего элемента каждого столбца.
14. Найти в матрице A первую из строк, сумма элементов которой максимальна. Упорядочить строки матрицы в порядке убывания элементов найденной строки.
15. Найти в матрице A первую из строк, в которой более двух максимальных элементов. Пусть это строка с номером k . Упорядочить строки матрицы A в порядке возрастания элементов строки k .
16. Найти в матрице A первый из столбцов с нечетным номером, который содержит максимальное количество четных элементов. Пусть это будет столбец с номером p . Упорядочить столбцы матрицы A в порядке возрастания элементов столбца с номером p .

17. Найти в матрице A любую из строк, содержащую наибольшее количество четных элементов. Пусть это строка с номером k . Упорядочить строки матрицы A в порядке убывания элементов строки с номером k .

18. Найти в матрице A любую строку, содержащую минимальное количество четных элементов. Пусть это будет строка с номером k . Упорядочить все строки матрицы A в порядке убывания элементов строки с номером k .

19. Найти в матрице A любую строку, не содержащую элементов кратных 5. Пусть это будет строка с номером k . Упорядочить все строки матрицы A с четными номерами в порядке убывания элементов строки с номером k .

20. Найти в матрице A столбец, в котором находится элемент, минимально отклоняющийся от среднего арифметического положительных элементов матрицы, расположенных выше главной диагонали. Пусть это будет столбец с номером p . Упорядочить все столбцы матрицы A в порядке убывания элементов столбца p , считая $m = n$.

21. Пусть для матрицы A $m = n$. Если произведение ненулевых элементов, расположенных в матрице A выше побочной диагонали, меньше произведения ненулевых элементов, расположенных ниже побочной диагонали, упорядочить все строки в порядке возрастания сумм цифр последних элементов каждой строки.

22. Найти в матрице A номер любой строки, в которой расположен элемент с максимальной суммой цифр. Пусть это будет строка с номером k . Упорядочить строки с нечетными номерами в порядке убывания элементов с четными индексами строки с номером k .

23. Упорядочить по возрастанию элементы тех строк с четными номерами, среди цифр элементов которых нет ни одной нечетной цифры.

24. Пусть в матрице A $m = n$. Если элементы главной диагонали упорядочены по возрастанию, упорядочить строки матрицы A по убыванию сумм первых элементов каждой строки.

25. Пусть в матрице A $m = n$. Если элементы побочной диагонали упорядочены по убыванию, упорядочить столбцы матрицы A по возрастанию сумм последних элементов каждого столбца.

Глава 8. СИМВОЛЫ И СИМВОЛЬНЫЕ СТРОКИ

8.1. Условия заданий с решениями

1. В программе объявлена символьная переменная `ch`, которой присвоено значение `'a'`

```
. . .  
char ch;  
ch = 'a';  
. . .
```

Чему равно содержимое байта памяти, который соответствует переменной `ch`?

2. По одному символу до точки вводится последовательность символов. Составить программу, которая определит, сколько в этой последовательности четных цифр; знаков «запятая», «точка с запятой», «двоеточие»; знаков арифметических операций «+», «-», «*», «/» и количество других символов.

3. Предложение оканчивается точкой, состоит из слов, разделенных одним пробелом. В начале предложения может быть любое количество пробелов. Найти количество символов в первом слове. Символы предложения вводятся по одному.

4. Составить программу для подсчета количества слов и непробельных символов в строке, вводимой пользователем с клавиатуры, с использованием библиотечной функции `getche()`. Слова разделены одним пробелом. Введение строки заканчивается нажатием клавиши **Enter**.

Напомним, что библиотечная функция `getche()` не имеет аргументов. Для ее использования в программе необходим оператор

```
char a = getche();
```

где *a* – идентификатор переменной символьного типа. Значение, возвращаемое функцией `getche()`, присваивается переменной *a* и выводится на экран. Описание функции `getche()` содержится в заголовочном файле `conio.h`.

5. Символы предложения вводятся с клавиатуры по одному символу до нажатия клавиши **Enter**. Предложение состоит из слов, разделенных любым количеством пробелов. Найти число слов этого предложения. Составить программу.

6. В переменной *x* записано предложение, оканчивающееся точкой. Составить программу для определения, сколько в нем слов, если слова разделены одним пробелом и сколько в нем символов до точки. Анализируемое предложение начинается не с пробела.

7. Найти ошибку в программе.

```
int main()
```

```

{
    char str1[] = "массив";
    char *str2 = "указатель";
    cout << str1 << std::endl;
    cout << str2 << std::endl;
    str1++;
    str2++;
    cout << str2 << std::endl;
    return 0;
}

```

8. Составить программу, позволяющую скопировать символьную строку s1 в символьную строку s2.

9. В символьной строке *a* представлена информация об участнице конкурса красоты, номером которой являются первые два символа этой строки. Составить программу для печати номера участницы, ее роста (height) и веса (weight).

10. Составить программу для заполнения массива элементами типа double из символьной строки.

11. Каков результат работы программы?

```

#include <iostream>
using std::cout;
int main()
{
    const int days = 7;
    const int length = 10;
    char day[days][length] = {"Sunday", "Monday", "Tuesday",
    "Wednesday", "Thursday", "Friday", "Saturday"};

    for (int j = 0; j < days; j++)
        cout << day[j] << std::endl;

    cin.get();
    return 0;
}

```

12. Составить программу, позволяющую определить, сколько цифр в строке, длина которой определяется вводом.

13. Составить программу, которая проверяет упорядоченность по возрастанию символов строки, количество элементов которой определяется вводом.

14. Сформулировать условие задачи, программа которой приведена ниже.

```

#include <iostream>
using std::cin;

```

```

using std::cout;
int main()
{
    char *str1 = new char [10];
    char *str2 = new char [10], *d = str2;
    cin >> str1;
    while (*str1 != '\0')
        if (*(str1) >= 'a' && *(str1++) <= 'z')
            *(d++) = '\xFE';
        else
            *(d++) = *(str1++);
    *(d) = '\0';
    cout << str2 << std::endl;
    delete [] str2;
    cin.ignore().get();
    return 0;
}

```

15. В трех символьных строках a , b , c хранится информация об участниках конкурса красоты «Мисс ВГУ». Первые две цифры каждой из строк соответствуют номеру участницы, далее записан ее рост, вес и параметры. Составить программу, которая определит номер участницы с минимальным весом.

16. Известен список фамилий m человек. Максимальная длина каждой фамилии n символов. Составить программу, которая напечатает первую самую короткую фамилию списка, ее длину, порядковый номер, количество фамилий минимальной длины.

8.2. Решения

1. В программе объявлена символьная переменная ch , которой присвоено значение 'a'

```

. . .
char ch;
ch = 'a';
. . .

```

Чему равно содержимое байта памяти, который соответствует переменной ch ?

Решение.

Десятичный код символа 'a' равен 97. Переводим число 97 в двоичную систему счисления, получим 1100001. Это значение соответствующего байта памяти.

Ответ: 01100001.

2. По одному символу до точки вводится последовательность символов. Составить программу, которая определит, сколько в этой последовательности четных цифр; знаков «запятая», «точка с запятой», «двоеточие»; знаков арифметических операций «+», «-», «*», «/» и количество других символов.

Решение.

```
#include <iostream>
#include <Windows.h>
using std::cin;
using std::cout;
int main()
{
    SetConsoleOutputCP(1251);
    char x; // вводимый символ
    int k = 0; // количество четных цифр
    int p = 0; // количество , ; :
    int r = 0; // количество + - * /
    int q = 0; // количество остальных символов
    cin >> x;
    while (x!='.')
    {
        switch (x)
        {
            case '0':
            case '2':
            case '4':
            case '6':
            case '8':
                k++;
                break;
            case ',':
            case ';':
            case ':':
                p++;
                break;
            case '+':
            case '-':
            case '*':
            case '/':
                r++;
                break;
            default:
                q++;
                break;
        }
    }
}
```



```

    }
    cin >> x;
}
cout << "Во введенной последовательности\n";
cout << "Количество четных цифр " << k << std::endl;
cout << "Количество знаков препинаний " << p << std::endl;
cout << "Количество арифметических операций " << r <<
std::endl;
cout << "Число остальных символов " << q << std::endl;
cin.ignore().get();
return 0;
}

```

3. Предложение оканчивается точкой, состоит из слов, разделенных одним пробелом. В начале предложения может быть любое количество пробелов. Найти количество символов в первом слове. Символы предложения вводятся по одному.

Решение.

```

#include <iostream>
#include <Windows.h>
using std::cin;
using std::cout;
int main()
{
    SetConsoleOutputCP(1251);
    char c; // вводимый символ
    cout << "Введите предложение:\n";
    c = cin.get();
    if (c!='.') // если предложение не пустое
    {
        while (c==' ') // пропуск пробелов в начале
            // предложения
            c = cin.get();
        if (c!='.') // отсекаем случай, когда предложение
            // состоит из одних пробелов
        {
            int k = 0; // искомое количество символов
            // в первом слове предложения
            while (c!='.' && c!=' ')
            {
                k++;
                c = cin.get();
            }
            cout << "Длина первого слова " << k <<

```

```

" символов\n";
    }
    else
        cout << "Нет первого слова\n";
}
else
    cout << "Нет предложения\n";
cin.ignore(cin.rdbuf()->in_avail()).get();
return 0;
}

```

Функция `rdbuf()` позволяет обратиться к буферу чтения потока ввода. Функция `in_avail()` возвращает количество символов, которое содержится в буфере. Это и есть количество символов, которое необходимо проигнорировать функцией `ignore()`.

4. Составить программу для подсчета количества слов и непробельных символов в строке, вводимой пользователем с клавиатуры, с использованием библиотечной функции `getche()`. Слова разделены одним пробелом. Введение строки заканчивается нажатием клавиши **Enter**.

Напомним, что библиотечная функция `getche()` не имеет аргументов. Для ее использования в программе необходим оператор

```
char a = getche();
```

где *a* – идентификатор переменной символьного типа. Значение, возвращаемое функцией `getche()`, присваивается переменной *a* и печатается на экран. Описание функции `getche()` содержится в заголовочном файле `conio.h`.

Решение.

```

#include <iostream>
#include <conio.h>
#include <Windows.h>
using std::cin;
using std::cout;
int main()
{
    SetConsoleOutputCP(1251);

    int chcount = 0; // количество непробельных символов
    int probel = 1; // число пробелов
    char ch = ' '; // очередной вводимый символ
    cout << "Введите строку\n";
    while (ch != '\r') // пока не нажата клавиша Enter
    {
        ch = getche(); // считывание очередного символа строки
    }
}

```

```

        if (ch == ' ') // если символом оказался пробел,
            probel++; // увеличиваем число пробелов
        else // в противном случае
            chcount++; // увеличиваем число непробельных
                    // символов
    }
    cout << "\nСлов " << probel << std::endl;
    cout << "Символов, отличных от пробела " << chcount-1 <<
std::endl;
    cin.get();
    return 0;
}

```

5. Символы предложения вводятся с клавиатуры по одному символу до нажатия клавиши **Enter**. Предложение состоит из слов, разделенных любым количеством пробелов. Найти число слов этого предложения. Составить программу.

Решение.

```

#include <iostream>
#include <conio.h>
#include <Windows.h>
using std::cin;
using std::cout;

enum SLOVO {NO, YES}; // для getch() NO=0, YES=1

int main()
{
    SetConsoleOutputCP(1251);
    SLOVO isSLOVO = NO;
    int count = 0; // число слов, которое необходимо найти
    char ch = ' ';
    do
    {
        ch = getch(); // ввод очередного символа
        if (ch == ' ' || ch == '\r') // если введен пробел,
        {
            if (isSLOVO == YES) // а до этого вводились
                                // символы слова, значит
                                // слово закончилось
            {
                count++; // учет количества слов
                isSLOVO = NO; // сброс флага
            }
        }
    }
}

```

```

    }
    else // в противном случае ввод символов продолжается
        if (isSLOVO == NO) // если начался ввод
            // очередного слова
            isSLOVO = YES; // изменяем значение флага
    } while (ch != '\r'); // выход из цикла по нажатию клавиши
        // Enter
    cout << "\n*** Число слов " << count << " ***\n";
    cin.get();
    return 0;
}

```

6. В переменной *x* записано предложение, оканчивающееся точкой. Составить программу для определения, сколько в нем слов, если слова разделены одним пробелом и сколько в нем символов до точки. Анализируемое предложение начинается не с пробела.

Решение.

```

#include <iostream>
#include <conio.h>
#include <Windows.h>
using std::cin;
using std::cout;
enum SLOVO {NO, YES}; // для getch() NO=0, YES=1
int main()
{
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);

    char x[] = {'Г','о','р','о','д',' ','В','о','р','о','
'н','е','ж',' ','о','с','н','о','в','а','н',' ','в',' ','
'1','5','8','6',' ','г','о','д','у','.'};
    int k = 0; // количество слов
    int i = 0; // количество символов в предложении

    while (x[i]!='.') // пока очередной символ предложения
        // не совпадает с точкой, являющейся
        // концом предложения
    {
        if (x[i] == ' ') // проверка на конец слова
            k++;
        i++; // для перехода к анализу очередного символа
    }
    cout << "В предложении " << k+1 << " слов\n";
    cout << "В предложении " << i << " символов\n";
}

```

```

        cin.get();
        return 0;
    }

```

7. Найти ошибку в следующем фрагменте программы.

```

int main()
{
    char str1[] = "массив";
    char *str2 = "указатель";
    cout << str1 << std::endl;
    cout << str2 << std::endl;
    str1++;
    str2++;
    cout << str2 << std::endl;
    return 0;
}

```

Ответ: так как `str1` – это адрес, т. е. указатель-константа, то `str1` не может изменять свое значение (в отличие от `str2`). Неверный оператор `str1++`.

8. Составить программу, позволяющую скопировать символьную строку `s1` в символьную строку `s2`.

Решение. Рассмотрим несколько вариантов фрагментов программ, осуществляющих копирование.

8.1. Пусть длина строки `s1` известна. В ней, например, 10 символов.

```

char s1[11], s2[11]; // длина указана с учетом нуля-терминатора
for (int i=0; i<11; i++)
    s2[i] = s1[i];

```

8.2. Если последний символ строки `s1` известен, то можно в заголовке цикла сменить условие проверки его окончания.

```

char s1[] = "1+2+3+4+5+0", s2[12];
for (int i=0; s1[i]; i++)
    s2[i] = s1[i];
s2[11] = '\0'; // завершающий ноль-символ

```

8.3. С использованием стандартной библиотечной функции `strlen(x)`, которая вычисляет длину строки `x` до нуля-терминатора.

```

char s1[] = "Воронеж - центр Черноземья";
const int length = 50; // максимальная длина строки
char s2[length];
for (int i=0; i<=strlen(s1); i++)
    // нестрогий знак отношения (<=) позволяет учесть
    // при копировании ноль-терминатор
    s2[i] = s1[i];

```

8.4. Пусть p – указатель на первый символ копируемой строки, q – указатель на результат.

```
char s1[] = "Москва", s2[30];
char *p=s1, *q=s2;
while (*p)
{
    *q = *p;
    q = q + 1;
    p = p + 1;
}
*q = '\0'; // завершающий ноль-символ
```

8.5. Копирование строки с использованием указателей

```
char *s1 = new char[10], *p = s1;
char *s2 = new char[10], *q = s2;

cin >> s1; // ввод исходной строки
while (*p != 0)
    *(q++) = *(p++);
*q = '\0'; // завершающий ноль-символ

cout << s2 << std::endl;
```

```
delete [] s1;
delete [] s2;
```

8.6. Если поставить оператор, выполняющий посимвольное присваивание элементов строки, с одновременной инкрементацией указателей

$*(q++) = *(p++)$

в заголовок цикла `while` на место условия, то цикл копирования строки примет вид

`while (*(q++) = *(p++));`

8.7. Стандартная библиотечная функция `strcpy` копирует все символы одной строки в другую, включая ноль-терминатор.

```
char s1[] = "Россия - Родина моя";
int m;
cin >> m;
char *s2 = new char[m];
```

```
strcpy(s2,s1);
```

```
cout << s2 << std::endl;
delete [] s2;
```

Заметим, что программист должен позаботиться, чтобы $m \geq 20$.

9. В символьной строке *a* представлена информация об участнице конкурса красоты, номером которой являются первые два символа этой строки. Составить программу для печати номера участницы, ее роста (height) и веса (weight).

```
#include <iostream>
#include <conio.h>
#include <Windows.h>
using std::cout;
int main()
{
    SetConsoleOutputCP(1251);
    char a[] = "16) Рост - 172 см; вес - 60.5 кг; параметры 95
61 93";
    int num; // номер
    long height; // рост
    double weight; // вес
    num = atoi(a); // преобразование символов 16 в целое число
    height = atol(&a[11]);
    weight = atof (&a[25]);
    cout << num << ' ' << height << ' ' << weight << std::endl;
    cin.get();
    return 0;
}
```

10. Составить программу для заполнения массива элементами типа double из символьной строки.

```
#include <iostream>
#include <conio.h>
using std::cout;
int main()
{
    char s[] = "5, 12, 37.58, 77, 60.5, 1777, 19";
    char *p = s;
    double a[10]; // фактическое количество элементов массива
                  // определим после окончания работы цикла
    int i = 0; // начальный номер элемента массива
    do
    {
        a[i++] = atof(p); // преобразование символьной
                          // числовой константы в элемент
                          // массива типа double
        if (i>9) break;
    }while (p = strchr(p, ','), p++);
    for (int k = 0; k<i; k++)
```

```

        cout << a[k] << ' ';
    cout << std::endl;
    cin.ignore().get();
    return 0;
}

```

11. Каков результат работы программы?

```

#include <iostream>
using std::cout;
int main()
{
    const int days = 7;
    const int length = 10;
    char day[days][length] = {"Sunday", "Monday", "Tuesday",
    "Wednesday", "Thursday", "Friday", "Saturday"};
    for (int j = 0; j < days; j++)
        cout << day[j] << std::endl;
    cin.get();
    return 0;
}

```

Ответ: вывод на экран названий семи дней недели.

12. Составить программу, позволяющую определить, сколько цифр в строке, длина которой определяется вводом.

```

#include <iostream>
#include <Windows.h>
using std::cin;
using std::cout;
int main()
{
    SetConsoleOutputCP(1251);
    int n;
    cout << "Введите длину строки: ";
    cin >> n;
    char *str = new char [n+1]; // включая конец строки '\0'
    cout << "Введите символы строки:\n";
    for (int i = 0; i < n; i++)
        cin >> *(str+i);
    str[n] = '\0';
    int k = 0;
    for (int i = 0; i < n; i++)
        if (*(str+i) >= '0' && *(str+i) <= '9')
            k++;
    cout << "В строке " << k << " цифр\n";
}

```



```

delete [] str;
cin.ignore().get();
return 0;
}

```

13. Составить программу, которая проверяет упорядоченность по возрастанию символов строки, количество элементов которой определяется вводом.

Вариант 1.

```

#include <iostream>
#include <Windows.h>
using std::cin;
using std::cout;
int main()
{
    SetConsoleOutputCP(1251);

    int n;
    cout << "Введите длину строки: ";
    cin >> n;
    char *str = new char [n+1]; // включая конец строки '\0'
    cout << "Введите символы строки:\n";
    for (int i = 0; i < n; i++)
        cin >> *(str+i);
    str[n] = '\0';
    bool t = true; // t = true, если два соседних символа
                  // упорядочены
    int i = 0; // номер очередного символа введенной строки
    while (t && i < n-1) // цикл продолжается пока не дошли
                        // до конца строки и предыдущие
                        // символы были упорядочены
    {
        t = str[i] < str[i+1]; // сравнение двух соседних
                              // символов
        if (t) i++; // переход к следующему символу, если
                  // предыдущие были упорядочены
    }

    if (t) cout << "\nСимволы строки упорядочены
                                     по возрастанию";
    else cout << "\nСимволы строки не упорядочены
                                     по возрастанию";

    delete [] str;
}

```

```

    cin.ignore().get();
    return 0;
}

```

Вариант 2.

С использованием цикла for, из тела которого предусмотрен выход (break) в случае, если предыдущий символ больше следующего за ним.

```

#include <iostream>
#include <Windows.h>
using std::cin;
using std::cout;
int main()
{
    SetConsoleOutputCP(1251);
    int n;
    cout << "Введите длину строки: ";
    cin >> n;
    char *str = new char [n+1]; // включая конец строки '\0'
    cout << "Введите символы строки:\n";
    for (int i = 0; i < n; i++)
        cin >> *(str+i);
    str[n] = '\0';
    bool t = true; // t = true, если два соседних символа
                  // упорядочены
    for (int i = 0; i < strlen(str) - 1; i++)
        if (*(str+i) > *(str+i+1)) // сравнение двух соседних
                                   // символов
        {
            t = false; // в случае, если они не упорядочены
                       // по возрастанию, выход из цикла
            break;
        }
    if (t) cout << "\nСимволы строки упорядочены
                                   по возрастанию";
    else cout << "\nСимволы строки не упорядочены
                                   по возрастанию";

    delete [] str;
    cin.ignore().get();
    return 0;
}

```

14. Сформулировать условие задачи, программа которой приведена ниже.

```
#include <iostream>
using std::cin;
using std::cout;
int main()
{
    char *str1 = new char [10];
    char *str2 = new char [10], *d = str2;
    cin >> str1;

    while (*str1 != '\0')
        if (*(str1) >= 'a' && *(str1++) <= 'z')
            *(d++) = '\xFE';
        else
            *(d++) = *(str1++);
    *(d) = '\0';

    cout << str2 << std::endl;

    delete [] str2;
    cin.ignore().get();
    return 0;
}
```

Ответ: символы строки str1 копируются в str2, все строчные латинские буквы (от 'a' до 'z') заменяются символом '■', десятичный код которого в таблице символов CP866 (ASCII, OEM, DOS) равен 254 (в шестнадцатеричной системе счисления FE).

15. В трех символьных строках *a*, *b*, *c* хранится информация об участниках конкурса красоты «Мисс ВГУ». Первые две цифры каждой из строк соответствуют номеру участницы, далее записан ее рост, вес и параметры. Составить программу, которая определит номер участницы с минимальным весом.

```
#include <iostream>
#include <Windows.h>
using std::cin;
using std::cout;
int main()
{
    SetConsoleOutputCP(1251);
    char a[] = " 25) Рост - 175 см; вес - 58.5 кг; параметры 95 61 98";
```

```

        char b[] = " 56) Рост - 170 см; вес - 52.8 кг; параметры 67
58 96";
        char c[] = " 62) Рост - 175 см; вес - 50.6 кг; параметры 90
62 92";
        int numa = atoi(a), numb = atoi(b), numc = atoi(c);
        double wa = atof(&a[25]), wb = atof(&b[25]), wc =
atof(&c[25]);
        double min;
        int num;
        if (wa < wb)
        {
            min = wa;
            num = numa;
        }
        else
        {
            min = wb;
            num = numb;
        }
        if (wc < min)
            num = numc;
        cout << "Номер участницы конкурса с минимальным весом: " <<
num << std::endl;

        cin.get();
        return 0;
}

```

16. Известен список фамилий m человек. Максимальная длина каждой фамилии n символов. Составить программу, которая напечатает первую самую короткую фамилию списка, ее длину, порядковый номер, количество фамилий минимальной длины.

```

#include <iostream>
#include <Windows.h>
using std::cin;
using std::cout;
int main()
{
    SetConsoleOutputCP(1251);
    const int m = 9; // количество строк в массиве
    const int n = 20; // максимальная длина строки
    // массив фамилий
    char a[m][n] = {"Гуденко", "Юркий", "Павловская", "Ахо",
"Лафоре", "Сорос", "Ким", "Франка", "Подбельский"};

```

```

int num, min = n;
int k = 0; // количество фамилий минимальной длины
for (int i = 0; i < m; i++)
    if (strlen(a[i]) < min)
    {
        min = strlen(a[i]);
        num = i;
        k = 1;
    }
    else
        if (strlen(a[i]) == min)
            k++;
cout << "Первая из фамилий минимальной длины: " << a[num]
<< std::endl;
cout << "ее номер: " << num << std::endl;
cout << "В самой короткой фамилии " << min <<
                                     " символов\n";
cout << "Количество фамилий минимальной длины: " << k <<
std::endl;

cin.get();
return 0;
}

```

8.3. Тренировочные задания

1. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить программу для определения длины второго слова.

2. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить программу копирования символов второго слова с использованием библиотечной функции `strcpy` в строковую переменную `str`, учитывая, что длина каждого слова предложения не более 20 символов.

3. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить программу для копирования символов одного любого слова, первый символ которого не совпадает с первым символом предложения. Напечатать полученное слово.

4. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить программу для определения количества слов, среди символов которых есть хотя бы одна цифра.

5. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить программу для определения количества слов, среди символов которых есть только одна цифра.

6. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить программу для определения количества слов, среди символов которых нет ни одной цифры.

7. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить программу для копирования символов одного любого слова, среди символов которого нет ни одного символа, совпадающего с последним символом предшествующего слова.

8. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить программу для копирования в переменную `str` любого слова, у которого совпадают первый и последний символы. Длина каждого слова не более 20 символов.

9. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить символьную строку из первых символов каждого слова. В предложении не более 255 символов.

10. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить символьную строку из последних символов каждого слова. В предложении не более 255 символов.

11. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить символьную строку из порядковых номеров тех слов, в которых нет ни одной цифры.

12. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить символьную строку из порядковых номеров тех слов, в которых есть хотя бы одна цифра.

13. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить символьную строку из порядковых номеров тех слов, в которых совпадают первый и последний символы.

14. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить программу для копирования в переменную str любого слова, у которого совпадают первый и последний символы с первым и последним символами первого слова. Длина каждого слова не более 20 символов

15. По одному символу до точки вводятся символы предложения, которое состоит из слов, разделенных одним пробелом. В начале предложения может быть несколько пробелов.

Составить символьную строку из порядковых номеров тех слов, длина которых превышает длину первого слова.

16. Массив из 20 символьных строк содержит информацию об изделиях мебельной фабрики. Каждая строка начинается с пробела. Следующие две цифры соответствуют номеру изделия, далее записано название изделия (Стол, Стул, Шкаф), затем параметры изделия (ширина, длина, высота). Пример одной из строк:

12) Стол ширина - 75.5 см; длина - 90.8 см; высота - 80 см

Составить программу для определения номеров столов, площадь которых минимальна.

17. Массив из 20 символьных строк содержит информацию об изделиях мебельной фабрики. Каждая строка начинается с пробела. Следующие две цифры соответствуют номеру изделия, далее записано название изделия (Стол, Стул, Шкаф), затем параметры изделия (ширина, длина, высота). Пример одной из строк:

12) Стол ширина - 75.5 см; длина - 90.8 см; высота - 80 см

Составить программу для определения количества столов и шкафов.

18. Массив из 20 символьных строк содержит информацию об изделиях мебельной фабрики. Каждая строка начинается с пробела. Следующие две цифры соответствуют номеру изделия, далее записано название изделия (Стол, Стул, Шкаф), затем параметры изделия (ширина, длина, высота). Пример одной из строк:

12) Стол ширина - 75.5 см; длина - 90.8 см; высота - 80 см

Составить программу для формирования символьной строки из номеров изделий, являющихся столами, площадь которых больше $0,8 \text{ м}^2$.

19. Массив из 20 символьных строк содержит информацию об изделиях мебельной фабрики. Каждая строка начинается с пробела. Следующие две цифры соответствуют номеру изделия, далее записано название изделия (Стол, Стул, Шкаф), затем параметры изделия (цвет, ширина, длина, высота). Пример одной из строк:

25) Стол цвет – белый; ширина - 60.8 см; длина - 80.8 см;
высота - 90 см

Составить программу для подсчета количества столов белого цвета, высота которых больше 60 см.

20. Массив из 20 символьных строк содержит информацию об изделиях мебельной фабрики. Каждая строка начинается с пробела. Следующие две цифры соответствуют номеру изделия, далее записано название изделия (Стол, Стул, Шкаф), затем параметры изделия (цвет, ширина, длина, высота). Пример одной из строк:

25) Стол цвет – белый; ширина - 60.8 см; длина - 80.8 см;
высота - 90 см

Сформировать символьную строку из номеров шкафов максимального объема.

21. Массив из 20 символьных строк содержит информацию об изделиях мебельной фабрики. Каждая строка начинается с пробела. Следующие две цифры соответствуют номеру изделия, далее записано название изделия (Стол, Стул, Шкаф), затем параметры изделия (цвет, ширина, длина, высота). Пример одной из строк:

25) Стол цвет – белый; ширина - 60.8 см; длина - 80.8 см;
высота - 90 см

Подсчитать количество шкафов бежевого цвета, объем которых минимальный среди таких шкафов.

22. Массив из 15 строк содержит информацию о результатах соревнований по фигурному катанию. Соревнования оценивают 8 судей, каждый из которых сначала выставляет оценку за сложность программы фигуриста, затем оценку за артистичность исполнения. Пример одной из строк:

19) Чернов Егор; сложность программы : 6 5,5 5,5 6 4,3 6 5 6;
артистизм : 6 6 5,7 5,1 5,8 6 6 5,3

Здесь первые две цифры (19) – номер фигуриста, далее через пробел фамилия, имя, затем 8 оценок за сложность программы и 8 оценок за артистизм исполнения. При выведении двух итоговых оценок из каждой совокупности (за сложность и артистизм) удаляются самая высокая и самая низкая оценка. Для оставшихся оценок вычисляется среднее арифметическое в каждой совокупности. Если окажется, что самую высокую или самую низкую оценку выставили несколько судей, то удаляется только одна из них.

Требуется составить программу для формирования массива из символьных строк, содержащих номер фигуриста, его фамилию, имя и итоговые оценки за сложность и артистизм.

Для приведенного выше примера элемент нового массива будет иметь вид

19) Чернов Егор; сложность программы : 5,67;
артистизм : 5,73

23. Массив из 15 строк содержит информацию о результатах соревнований по фигурному катанию. Соревнования оценивают 8 судей, каждый из которых сначала выставляет оценку за сложность программы фигуриста, затем оценку за артистичность исполнения. Пример одной из строк:

19) Котолевский Павел; сложность программы : 6 5,5 5,5 6 4,3 6 5 6; артистизм : 6 6 5,7 5,1 5,8 6 6 5,3

Здесь первые две цифры (19) – номер фигуриста, далее через пробел фамилия, имя, затем 8 оценок за сложность программы и 8 оценок за артистизм исполнения. При выведении двух итоговых оценок из каждой совокупности (за сложность и артистизм) удаляются самая высокая и самая низкая оценка. Для оставшихся оценок вычисляется среднее арифметическое в каждой совокупности. Если окажется, что самую высокую или самую низкую оценку выставили несколько судей, то удаляется только одна из них.

Сформировать символьную строку из номеров фигуристов, сложность программы которых судьи оценили ниже 5 баллов.

24. Массив из 15 строк содержит информацию о результатах соревнований по фигурному катанию. Соревнования оценивают 8 судей, каждый из которых сначала выставляет оценку за сложность программы фигуриста, затем оценку за артистичность исполнения. Пример одной из строк:

19) Золотухин Николай; сложность программы : 6 5,5 5,5 6 4,3 6 5 6; артистизм : 6 6 5,7 5,1 5,8 6 6 5,3

Здесь первые две цифры (19) – номер фигуриста, далее через пробел фамилия, имя, затем 8 оценок за сложность программы и 8 оценок за артистизм исполнения. При выведении двух итоговых оценок из каждой совокупности (за сложность и артистизм) удаляются самая высокая и самая низкая оценка. Для оставшихся оценок вычисляется среднее арифметическое в каждой совокупности. Если окажется, что самую высокую или самую низкую оценку выставили несколько судей, то удаляется только одна из них.

Сформировать символьную строку str из номеров фигуристов, показавших по артистизму три лучших результата. Если таких фигуристов больше 3, включить в строку str номера всех таких спортсменов.

Пример сформированной строки

17) первое место 6 баллов; 8) второе место 5,8 баллов;
1), 2) 8) третье место 5 баллов

25. Массив из 15 строк содержит информацию о результатах соревнований по фигурному катанию. Соревнования оценивают 8 судей, каждый из которых сначала выставляет оценку за сложность программы фигуриста, затем оценку за артистичность исполнения. Пример одной из строк.

19) Прохоров Дмитрий; сложность программы : 6 5,5 5,5 6 4,3 6 5 6;
артистизм : 6 6 5,7 5,1 5,8 6 6 5,3

Здесь первые две цифры (19) – номер фигуриста, далее через пробел фамилия, имя, затем 8 оценок за сложность программы и 8 оценок за артистизм исполнения. При выведении двух итоговых оценок из каждой совокупности (за сложность и артистизм) удаляются самая высокая и самая низкая оценка. Для оставшихся оценок вычисляется среднее арифметическое в каждой совокупности. Если окажется, что самую высокую или самую низкую оценку выставили несколько судей, то удаляется только одна из них.

Сформировать символьную строку, в которую включить фамилию, имя победителя соревнований, его средние баллы по сложности и артистизму. Если победителей будет несколько, включить в формируемую строку информацию о каждом из них.

26. На разноцветной бумаге формата А4 заданы координаты трех точек. Меньшая сторона листа считается осью x , большая осью y , левый нижний угол началом координат (т. е. весь лист бумаги – это первая координатная четверть). Информация о каждом листе хранится в символьной строке, которая является элементом массива, состоящим из 90 строк вида:

42) цвет : зеленый ; $x_1=5$, $y_1=20$; $x_2=12$, $y_2=20.8$;
 $x_3=40,6$, $y_3=32$;

Здесь первые две цифры (42) – это номер листа бумаги зеленого цвета, (x_1, y_1) , (x_2, y_2) , (x_3, y_3) – координаты трех точек.

Составить символьную строку из номеров листов бумаги синего цвета, если точки (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , расположенные на этом листе, могут быть вершинами равнобедренного треугольника.

27. На разноцветной бумаге формата А4 заданы координаты трех точек. Меньшая сторона листа считается осью x , большая осью y , левый нижний угол началом координат (т. е. весь лист бумаги – это первая координатная четверть). Информация о каждом листе хранится в символьной строке, которая является элементом массива, состоящим из 90 строк вида:

42) цвет : зеленый ; $x_1=5$, $y_1=20$; $x_2=12$, $y_2=20.8$;
 $x_3=40,6$, $y_3=32$;

Здесь первые две цифры (42) – это номер листа бумаги зеленого цвета, (x_1, y_1) , (x_2, y_2) , (x_3, y_3) – координаты трех точек.

Составить символьную строку из номеров листов бумаги красного цвета, если точки (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , расположенные на этом листе, могут быть вершинами прямоугольного треугольника.

28. На разноцветной бумаге формата А4 заданы координаты трех точек. Меньшая сторона листа считается осью x , большая осью y , левый нижний угол началом координат (т. е. весь лист бумаги – это первая координатная четверть). Информация о каждом листе хранится в символьной строке, которая является элементом массива, состоящим из 90 строк вида:

42) цвет : зеленый ; $x_1=5$, $y_1=20$; $x_2=12$, $y_2=20.8$;
 $x_3=40,6$, $y_3=32$;

Здесь первые две цифры (42) – это номер листа бумаги зеленого цвета, (x_1, y_1) , (x_2, y_2) , (x_3, y_3) – координаты трех точек.

Подсчитать количество листов бумаги, на которых точки (x_1, y_1) , (x_2, y_2) , (x_3, y_3) являются вершинами треугольника с максимальной площадью.

29. На разноцветной бумаге формата А4 заданы координаты трех точек. Меньшая сторона листа считается осью x , большая осью y , левый нижний угол началом координат (т. е. весь лист бумаги – это первая координатная четверть). Информация о каждом листе хранится в символьной строке, которая является элементом массива, состоящим из 90 строк вида:

42) цвет : зеленый ; $x_1=5$, $y_1=20$; $x_2=12$, $y_2=20.8$;
 $x_3=40,6$, $y_3=32$;

Здесь первые две цифры (42) – это номер листа бумаги зеленого цвета, (x_1, y_1) , (x_2, y_2) , (x_3, y_3) – координаты трех точек.

Сформировать новый массив из символьных строк. Каждый элемент строки должен содержать номер листа, его цвет и значение площади треугольника, если точки (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , расположенные на этом листе, могут быть вершинами треугольника.

30. Из столовой на продуктовую базу поступил заказ в виде символьной строки

547) картофель : 5 кг , рис : 10 кг , сахар : 8 кг , морковь :
3 кг , свекла : 2 кг , капуста : 2 кг

в которой указан номер заказа (547) и сколько кг различных продуктов требуется привезти.

В символьной строке STR указана стоимость в рублях единицы каждого продукта. Например, строка STR может иметь следующий вид

картофель : 20 руб. , рис : 45 руб. , сахар : 53 руб. ,
морковь : 28 руб. , свекла : 30 руб. , капуста : 35 руб.

Составить программу для определения общей стоимости заказанных продуктов.

31. Из столовых города на продуктовую базу поступило 20 заказов, каждый из которых является символьной строкой, например, следующего вида

547) картофель : 5 кг , рис : 10 кг , сахар : 8 кг, морковь :
3 кг , свекла : 2 кг , капуста : 2 кг

Здесь 547 – номер столовой, а далее указано, сколько кг и каких продуктов требуется привезти столовой с номером 547.

В символьной строке STR указана стоимость в рублях единицы каждого продукта. Например, строка STR может иметь следующий вид

картофель : 20 руб. , рис : 45 руб. , сахар : 53 руб. ,
морковь : 28 руб. , свекла : 30 руб. , капуста : 35 руб.

Составить символьную строку из номеров столовых с максимальной общей стоимостью заказанных продуктов.

8.4. Задания для самостоятельной работы

1. На вход программе подаются строки, содержащие информацию о багаже N ($N \leq 100$) пассажиров авиарейса. Информация вводится построчно. Каждая строка состоит из символов и имеет вид:

<фамилия> <инициалы> <общий вес багажа> <количество вещей>

Пример входных строк:

Малежина И. И. 42 кг 2

Челядинов И. А. 54 кг 4

Составить программу, которая выведет в любом порядке фамилии пассажиров, средний вес одной вещи багажа которых больше среднего веса одной вещи всех пассажиров.

2. На вход программе подаются строки, содержащие информацию о багаже N ($N \leq 100$) пассажиров авиарейса. Информация вводится построчно. Каждая строка состоит из символов и имеет вид:

<фамилия> <инициалы> <общий вес багажа> <количество вещей>

Пример входных строк:

Глазунова В. О. 42 кг 2

Лапыгин В. В. 54 кг 4

Составить программу, которая выведет общий вес багажа и количество вещей всех пассажиров.

3. На вход программе подаются строки, содержащие информацию о багаже N ($N \leq 100$) пассажиров авиарейса. Информация вводится построчно. Каждая строка состоит из символов и имеет вид:

<фамилия> <инициалы> <общий вес багажа> <количество вещей>

Пример входных строк:

Семыкин Д. Д. 42 кг 2

Легеня Л. Н. 54 кг 4

Составить программу, которая выведет в любом порядке фамилии пассажиров, общий вес багажа которых минимален.

4. На вход программе подаются строки, содержащие информацию о багаже N ($N \leq 100$) пассажиров авиарейса. Информация вводится построчно. Каждая строка состоит из символов и имеет вид:

<порядковый номер> <фамилия> <общий вес багажа> <количество вещей>

Пример входных строк:

30) Курман Л. Л. 42 кг 2

17) Чернышова О. Б. 54 кг 4

Составить программу, которая сформирует строку из порядковых номеров пассажиров, общий вес багажа которых максимален.

5. На вход программе подаются строки, содержащие информацию о багаже N ($N \leq 100$) пассажиров авиарейса. Информация вводится построчно. Каждая строка состоит из символов и имеет вид:

<фамилия> <инициалы> <общий вес багажа> <количество вещей>

Пример входных строк:

Тугаев А. М. 42 кг 2

Кабанова Т. А. 54 кг 4

Составить программу, которая выведет количество пассажиров авиарейса с максимальным и минимальным общим весом.

6. На вход программе подаются в произвольном порядке символьные строки, содержащие утреннюю, дневную, вечернюю и ночную температуру в градусах за каждый день марта месяца 2016 года. Информация вводится построчно.

Пример входной строки

23) среда ; температура : утро 2 ; день 4 ; вечер 2 ; ночь -1
Здесь первые две цифры 23) – день марта месяца.

Найти среднюю из среднесуточных температур второй недели марта, если 1 марта был вторник.

7. На вход программе подаются в произвольном порядке символьные строки, содержащие утреннюю, дневную, вечернюю и ночную температуру в градусах за каждый день марта месяца 2016 года. Информация вводится построчно.

Пример входной строки

23) среда ; температура : утро 2 ; день 4 ; вечер 2 ; ночь -1
Здесь первые две цифры 23) – день марта месяца.

Найти первый от начала месяца день, когда среднесуточная температура была максимальной.

8. На вход программе подаются в произвольном порядке символьные строки, содержащие утреннюю, дневную, вечернюю и ночную температуру в градусах за каждый день марта месяца 2016 года. Информация вводится построчно.

Пример входной строки

23) среда ; температура : утро 2 ; день 4 ; вечер 2 ; ночь -1
Здесь первые две цифры 23) – день марта месяца.

Составить символьную строку из порядковых номеров дней марта месяца, когда дневная температура была максимальной.

9. На вход программе подаются в произвольном порядке символьные строки, содержащие утреннюю, дневную, вечернюю и ночную температуру в градусах за каждый день марта месяца 2016 года. Информация вводится построчно.

Пример входной строки

23) среда ; температура : утро 2 ; день 4 ; вечер 2 ; ночь -1
Здесь первые две цифры 23) – день марта месяца.

Составить символьную строку, в которую включить информацию о первом периоде начала и конца возрастания среднесуточных температур, причем длина этого периода должна быть более 2. Сформированная строка должна включать значение средней из среднесуточных температур за этот период.

Пример строки, которую должна сформировать программа

начало 9 марта конец 13 марта ; средняя из среднесуточных температур 8,5 градусов

10. На вход программе подаются в произвольном порядке символьные строки, содержащие утреннюю, дневную, вечернюю и ночную температуру в градусах за каждый день марта месяца 2016 года. Информация вводится построчно.

Пример входной строки

23) среда ; температура : утро 2 ; день 4 ; вечер 2 ; ночь -1
Здесь первые две цифры 23) – день марта месяца.

Составить символьную строку, в которую включить номера дней каждой недели марта месяца, когда вечерняя температура была максимальной.

11. На вход программе подаются в произвольном порядке символьные строки, содержащие утреннюю, дневную, вечернюю и ночную температуру в градусах за каждый день марта месяца 2016 года. Информация вводится построчно.

Пример входной строки

23) среда ; температура : утро 2 ; день 4 ; вечер 2 ; ночь -1
Здесь первые две цифры 23) – день марта месяца.

Составить символьную строку из номеров дней последней декады марта месяца, в которые среднесуточная температура превышала среднесуточную температуру за весь март месяц.

12. На вход программе подаются сведения о результатах зимней сессии студентов 1 курса факультета ПММ ВГУ, на котором учится 210 человек. Информация вводится построчно.

Пример входной строки

118) Винокуров экзамены : 5 4 5 5 4 ; зачеты : + + + + + - + +
Здесь первые три цифры 118) – трехзначное число, являющееся идентификационным номером первокурсника, далее результаты 5 экзаменов (оценки от 2 до 5), затем результаты зачетов («+» означает, что зачет сдан, «-» не сдан).
Оценки за экзамены и результаты сдачи зачетов разделяются во входной строке одним пробелом.

Составить программу, которая подсчитывает количество отличников и количество неуспевающих студентов.

Отличником считается студент, у которого сданы все зачеты и по всем экзаменам получена оценка 5.

У неуспевающего студента не сдан хотя бы один зачет или хотя бы один экзамен.

13. На вход программе подаются сведения о результатах зимней сессии студентов 1 курса факультета ПММ ВГУ, на котором учится 210 человек. Информация вводится построчно.

Пример входной строки

185) Анищев экзамены : 5 4 5 5 4 ; зачеты : + + + + + - + +
Здесь первые три цифры 185) – трехзначное число, являющееся идентификационным номером первокурсника, далее результаты 5 экзаменов (оценки от 2 до 5), затем результаты зачетов («+» означает, что зачет сдан, «-» не сдан).
Оценки за экзамены и результаты сдачи зачетов разделяются во входной строке одним пробелом.

Составить символьную строку из идентификационных номеров первокурсников, являющихся по результатам зимней сессии отличниками.

Отличником считается студент, у которого сданы все зачеты и по всем экзаменам получена оценка 5.

14. На вход программе подаются сведения о результатах зимней сессии студентов 1 курса факультета ПММ ВГУ, на котором учится 210 человек. Информация вводится построчно.

Пример входной строки

153) Панфилова экзамены : 5 4 5 5 4 ; зачеты : + + + + + - + +
Здесь первые три цифры 153) – трехзначное число, являющееся идентификационным номером первокурсника, далее результаты 5 экзаменов (оценки от 2 до 5), затем результаты зачетов («+» означает, что зачет сдан, «-» не сдан).
Оценки за экзамены и результаты сдачи зачетов разделяются во входной строке одним пробелом.

Составить символьную строку из идентификационных номеров неуспевающих первокурсников.

У неуспевающего студента не сдан хотя бы один зачет или хотя бы один экзамен.

15. На вход программе подаются сведения о результатах зимней сессии студентов 1 курса факультета ПММ ВГУ, на котором учится 210 человек. Информация вводится построчно.

Пример входной строки

153) Зайцев экзамены : 5 4 5 5 4 ; зачеты : + + + + + - + +
Здесь первые три цифры 153) – трехзначное число, являющееся идентификационным номером первокурсника, далее результаты 5 экзаменов (оценки от 2 до 5), затем результаты зачетов («+» означает, что зачет сдан, «-» не сдан). Оценки за экзамены и результаты сдачи зачетов разделяются во входной строке одним пробелом.

Составить символьную строку из идентификационных номеров первокурсников, которым назначена стипендия.

Стипендия назначается студентам, у которых все экзамены сданы на «4» и «5» и сданы все зачеты.

16. На вход программе подаются сведения о результатах зимней сессии студентов 1 курса факультета ПММ ВГУ, на котором учится 210 человек. Информация вводится построчно.

Пример входной строки

251) Рабаджан экзамены : 5 4 5 5 4 ; зачеты : + + + + + - + +
Здесь первые три цифры 251) – трехзначное число, являющееся идентификационным номером первокурсника, далее результаты 5 экзаменов (оценки от 2 до 5), затем результаты зачетов («+» означает, что зачет сдан, «-» не сдан). Оценки за экзамены и результаты сдачи зачетов разделяются во входной строке одним пробелом.

Составить одномерный массив из фамилий и инициалов отличников.

Отличником считается студент, у которого сданы все зачеты и по всем экзаменам получена оценка 5.

17. На вход программе подаются сведения о результатах зимней сессии студентов 1 курса факультета ПММ ВГУ, на котором учится 210 человек. Информация вводится построчно.

Пример входной строки

153) Зайцев экзамены : 5 4 5 5 4 ; зачеты : + + + + + - + +
Здесь первые три цифры 153) – трехзначное число, являющееся идентификационным номером первокурсника, далее результаты 5 экзаменов (оценки от 2 до 5), затем результаты зачетов («+» означает, что зачет сдан, «-» не сдан). Оценки за экзамены и результаты сдачи зачетов разделяются во входной строке одним пробелом.

Составить программу для вычисления стипендиального фонда первокурсников факультета ПММ ВГУ, если размер обычной стипендии составляет 1 800 руб., а стипендия отличников увеличена на 50 %.

18. На вход программе подаются сведения о результатах зимней сессии студентов 1 курса факультета ПММ ВГУ, на котором учится 210 человек. Информация вводится построчно.

Пример входной строки

127) Татаренко экзамены : 5 4 5 5 4 ; зачеты : + + + + + - + +
Здесь первые три цифры 127) – трехзначное число, являющееся идентификационным номером первокурсника, далее результаты 5 экзаменов (оценки от 2 до 5), затем результаты зачетов («+» означает, что зачет сдан, «-» не сдан). Оценки за экзамены и результаты сдачи зачетов разделяются во входной строке одним пробелом.

Составить символьную строку из идентификационных номеров успевающих первокурсников.

Успевающим считается студент, который сдал все зачеты и экзамены.

19. На вход программе подаются сведения о результатах городской олимпиады школьников по информатике, в которой принимало участие N человек ($N \leq 500$). Информация вводится построчно. Каждая строка состоит из символов и имеет вид:

<Фамилия> <Имя> <номер школы> <баллы>

Здесь <Фамилия> – фамилия участника олимпиады, <Имя> – его имя. Фамилия и имя состоят не более чем из 30 символов. <баллы> – количество набранных этим участником баллов. Известно, что максимальное количество баллов, которое можно набрать 50; число школ в городе не превосходит 100 и их номера изменяются в диапазоне от 1 до 99.

Пример входных строк

Масюков Александр 57 48

Викарчук Андрей 1 42

Все элементы одной строки разделены одним пробелом.

Составить программу, которая будет выводить на экран в порядке возрастания номеров школ строки, каждая из которых содержит номер школы и максимальный балл, набранный участниками олимпиады из этой школы. Номера школ, ученики которых не принимали участие в олимпиаде, не выводятся. Информация о номерах школ, отсутствующих в городе, не выводится.

20. На вход программе подаются сведения о результатах городской олимпиады школьников по информатике, в которой принимало участие N человек ($N \leq 500$). Информация вводится построчно. Каждая строка состоит из символов и имеет вид:

<Фамилия> <Имя> <номер школы> <баллы>

Здесь <Фамилия> – фамилия участника олимпиады, <Имя> – его имя. Фамилия и имя состоят не более чем из 30 символов. <баллы> – количество набранных этим участником баллов. Известно, что максимальное количе-

ство баллов, которое можно набрать 50; число школ в городе не превосходит 100 и их номера изменяются в диапазоне от 1 до 99.

Пример входных строк

Подакина Татьяна 57 48

Симонян Эрнест 1 42

Все элементы одной строки разделены одним пробелом.

Составить символьную строку из номеров школ, в которых есть хотя бы один ученик, набравший в городской олимпиаде школьников по информатике лучший среди соревнующихся балл.

21. На вход программе подаются сведения о результатах городской олимпиады школьников по информатике, в которой принимало участие N человек ($N \leq 500$). Информация вводится построчно. Каждая строка состоит из символов и имеет вид:

<Фамилия> <Имя> <номер школы> <баллы>

Здесь <Фамилия> – фамилия участника олимпиады, <Имя> – его имя. Фамилия и имя состоят не более чем из 30 символов. <баллы> – количество набранных этим участником баллов. Известно, что максимальное количество баллов, которое можно набрать 50; число школ в городе не превосходит 100 и их номера изменяются в диапазоне от 1 до 99.

Пример входных строк

Золотухин Алексей 57 50

Тишин Ярослав 1 42

Все элементы одной строки разделены одним пробелом.

Составить программу, которая будет выводить на экран в порядке возрастания номеров школ строки, каждая из которых содержит номер школы и количество учеников этой школы, принимавших участие в городской олимпиаде. Номера школ, ученики которых не принимали участие в олимпиаде, не выводятся. Информация о номерах школ, отсутствующих в городе, не выводится.

22. На вход программе подаются символьные строки, в которых содержится информация о количестве выпущенных изделий по каждому из цехов некоторого предприятия: положительные значения означают количество изделий, выпущенных сверх плана, отрицательные – недовыполнение плана, ноль – выполнение плана. Каждый цех выпускает однотипные изделия пяти видов. Количество цехов равно N ($N < 10$).

Пример входной строки

20A) 10 0 -100 0 5

Здесь первые три символа 20A) означают номер цеха.

Определить, сколько цехов перевыполнило план по второму изделию, не выполнило план по пятому изделию, выполнило плановое задание по первому изделию.

23. На вход программе подаются символьные строки, в которых содержится информация о количестве выпущенных изделий по каждому из цехов некоторого предприятия: положительные значения означают количество изделий, выпущенных сверх плана, отрицательные – недовыполнение плана, ноль – выполнение плана. Каждый цех выпускает однотипные изделия пяти видов. Количество цехов равно N ($N < 10$).

Пример входной строки

20A) 10 0 -100 0 5

Здесь первые три символа 20A) означают номер цеха.

Составить символьную строку, содержащую номера цехов, в которых было недовыполнение плана хотя бы по одному изделию.

24. На вход программе подаются символьные строки, в которых содержится информация о количестве выпущенных изделий по каждому из цехов некоторого предприятия: положительные значения означают количество изделий, выпущенных сверх плана, отрицательные – недовыполнение плана, ноль – выполнение плана. Каждый цех выпускает однотипные изделия пяти видов. Количество цехов равно N ($N < 10$).

Пример входной строки

20A) 10 0 -100 0 5

Здесь первые три символа 20A) означают номер цеха.

Определить, сколько всего изделий сверх плана было выпущено на предприятии по каждому изделию.

25. На вход программе подаются символьные строки, в которых содержится информация о количестве выпущенных изделий по каждому из цехов некоторого предприятия: положительные значения означают количество изделий, выпущенных сверх плана, отрицательные – недовыполнение плана, ноль – выполнение плана. Каждый цех выпускает однотипные изделия пяти видов. Количество цехов равно N ($N < 10$).

Пример входной строки

20A) 10 0 -100 0 5

Здесь первые три символа 20A) означают номер цеха.

Составить символьную строку из номеров цехов, в которых имело место недовыполнение плана хотя бы по одному изделию.

8.5. Библиотечные функции для работы с символами и символьными строками

Функция	Пояснение
strlen(s)	Определяет длину строки <i>s</i> , без учёта нуль-символа. strlen("Информатика") = 11
Копирование строк	
strcpy(s1,s2)	Выполняет побайтное копирование символов из строки <i>s2</i> в строку <i>s1</i> . Результат сохраняется в <i>s1</i> . strcpy(s1,"Информатика"); s1 = "Информатика"
strncpy(s1,s2,n)	Выполняет побайтное копирование <i>n</i> символов из строки <i>s2</i> в строку <i>s1</i> без учета нуль-символа. Результат сохраняется в <i>s1</i> . strncpy(s1,"Информатика",4); s1[4]='\0'; s1 = "Инфо"
Конкатенация строк	
strcat(s1,s2)	Объединяет строку <i>s2</i> со строкой <i>s1</i> . Результат сохраняется в <i>s1</i> . char s1[25]="Инфо"; strcat(s1,"рматика"); s1 = "Информатика"
strncat(s1,s2,n)	Объединяет <i>n</i> символов строки <i>s2</i> со строкой <i>s1</i> . Результат сохраняется в <i>s1</i> . char s1[25]="Инфо"; strncat(s1,"рматика***",7); s1 = "Информатика"
Сравнение строк	
strcmp(s1,s2)	Сравнивает строку <i>s1</i> со строкой <i>s2</i> и возвращает результат типа int: 0 – если строки эквивалентны, > 0 – если <i>s1</i> > <i>s2</i> , < 0 – если <i>s1</i> < <i>s2</i> , с учётом регистра. strcmp("самолет","самокат") > 0 strcmp("самокат","самолет") < 0 strcmp("самолет","самолет") = 0
strncmp(s1,s2,n)	Сравнивает <i>n</i> символов строки <i>s1</i> со строкой <i>s2</i> и возвращает результат типа int: 0 – если строки эквивалентны, > 0 – если <i>s1</i> < <i>s2</i> , < 0 – если <i>s1</i> > <i>s2</i> , с учётом регистра. strncmp("самолет","самокат",4) = 0

strcmp(s1,s2)	Сравнивает строку <i>s1</i> со строкой <i>s2</i> и возвращает результат типа int: 0 – если строки эквивалентны, > 0 – если <i>s1</i> > <i>s2</i> , < 0 – если <i>s1</i> < <i>s2</i> , без учёта регистра. <code>strcmp("INfOrmation","information") = 0</code>
strnicmp(s1,s2,n)	Сравнивает <i>n</i> символов строки <i>s1</i> со строкой <i>s2</i> и возвращает результат типа int: 0 – если строки эквивалентны, > 0 – если <i>s1</i> > <i>s2</i> , < 0 – если <i>s1</i> < <i>s2</i> , без учёта регистра
Обработка символов	
isalnum(c)	Возвращает значение true, если <i>c</i> является буквой или цифрой, и false в других случаях. <code>bool t = isalnum('!'); // t = false</code> <code>bool t = isalnum('2'); // t = true</code> <code>bool t = isalnum('z'); // t = true</code> <code>bool t = isalnum('Z'); // t = true</code> Корректно вычисляется для латинских символов, работа с кириллицей зависит от компилятора, при обработке букв кириллицы может быть возбуждено исключение и прерывание работы программы.
isalpha(c)	Возвращает значение true, если <i>c</i> является буквой, и false в других случаях. <code>bool t = isalpha('2'); // t = false</code> <code>bool t = isalpha('z'); // t = true</code> <code>bool t = isalpha('Z'); // t = true</code> Корректно вычисляется для латинских символов, работа с кириллицей зависит от компилятора, при обработке букв кириллицы может быть возбуждено исключение и прерывание работы программы.
isdigit(c)	Возвращает значение true, если <i>c</i> является цифрой, и false в других случаях. <code>bool t = isdigit('2'); // t = true</code> <code>bool t = isdigit('z'); // t = false</code> Работа с кириллицей зависит от компилятора, при обработке букв кириллицы может быть возбуждено исключение и прерывание работы программы.
islower(c)	Возвращает значение true, если <i>c</i> является буквой нижнего регистра, и false в других случаях. <code>bool t = islower('z'); // t = true</code> <code>bool t = islower('Z'); // t = false</code> Работа с кириллицей зависит от компилятора, при обработке букв кириллицы может быть возбуждено исключение и прерывание работы программы.

isupper(c)	<p>Возвращает значение true, если <i>c</i> является буквой верхнего регистра, и false в других случаях.</p> <pre>bool t = isupper('z'); // t = false bool t = isupper('Z'); // t = true</pre> <p>Работа с кириллицей зависит от компилятора, при обработке букв кириллицы может быть возбуждено исключение и прерывание работы программы.</p>
isspace(c)	<p>Возвращает значение true, если <i>c</i> является пробелом, и false в других случаях.</p> <pre>bool t = isspace('*'); // t = false bool t = isspace(' '); // t = true</pre>
Функции поиска	
strchr(s, c)	<p>Поиск первого вхождения символа <i>c</i> в строке <i>s</i>. В случае удачного поиска возвращает указатель на место первого вхождения символа <i>c</i>. Если символ не найден, то возвращается пустая ссылка.</p> <pre>char s[25]="Информатика"; char *p = strchr(s, 'ф'); // определение позиции символа int position = (p - s + 1); // position = 3</pre>
strcspn(s1, s2)	<p>Определяет длину начального сегмента строки <i>s1</i>, содержащего те символы, которые не входят в строку <i>s2</i>.</p> <pre>strcspn("самолет", "вертолет") = 3</pre> <p>Начальным сегментом строки "самолет", который не входит в строку "вертолет" является "сам", длина сегмента равна 3</p>
strspn(s1, s2)	<p>Возвращает длину начального сегмента строки <i>s1</i>, содержащего только те символы, которые входят в строку <i>s2</i>.</p> <pre>strspn("самолет", "самокат") = 4</pre> <p>Совпадающим сегментом является "само", длина сегмента равна 4</p>
strpbrk(s1, s2)	<p>Возвращает указатель первого вхождения любого символа строки <i>s2</i> в строке <i>s1</i>.</p> <pre>char s[25]="Информатика"; char *p = strpbrk(s, "форма"); // определение позиции символа int position = (p - s + 1); // position = 3</pre>
Функции преобразования	
atof(s)	<p>Преобразует строку <i>s</i> в число типа double.</p> <pre>double x = atof("12.345"); // d = 12.345</pre> <p>Допустимым разделителем целой и дробной части яв-</p>

	<p>ляется точка, в противном случае происходит преобразование цифр до ближайшего символа, отличного от точки:</p> <pre>double x = atof("12,345"); // d = 12</pre>
atoi(s)	<p>Преобразует строку <i>s</i> в число типа int.</p> <pre>int x = atof("12"); // x = 12</pre> <p>Если в строке встречаются символы, отличные от цифр, то преобразование происходит до ближайшего символа, отличного от цифры</p> <pre>int x = atof("12.345"); // x = 12</pre>
atol(s)	Преобразует строку <i>s</i> в число типа long int

Глава 9. СТРУКТУРЫ

9.1. Вопросы и задания с решениями

9.1.1. Условия заданий

1. Как называется тип данных, общий вид которых представлен ниже?

```
struct [имя_типа]
{
    тип_1 элемент_1;
    тип_2 элемент_2;
    . . .
    тип_n элемент_n;
};
```

2. Чем структуры отличаются от массивов?

3. Описать структуру для обработки списка студентов ВГУ, элементы которого имеют, например, следующий вид.

Исаев Влад	ПММ	2	61	1800
Иванова Олеся	ФИЗФАК	1	72а	1200

30 символов	10 символов	курс	группа	размер стипендии
-------------	-------------	------	--------	------------------

4. В каком случае целесообразно не задавать имя структурного типа, а описывать сразу переменные? В чем недостаток такого описания?

5. Описать структуру, которая содержит сведения о данных, входящих в библиографическую карточку:

- Фамилия И.О. автора книги
- Заголовок книги
- Место издания
- Издательство
- Год издания
- Количество страниц

Количество символов для первых четырех полей неизвестно, поэтому типом этих полей должен быть указатель.

6. Пусть введен структурный тип card для библиографической карточки (как в предыдущем задании):

```
struct card {
    char *author;
    char *title;
    char *city;
    char *publisher;
    int year;
    int pages;
};
```

Дана следующая последовательность операторов:

```
card dic;
```



```
dic.author = "Ускова О.Ф., Каплиева Н.А.";
dic.title = "Начала структурного программирования на языке
            С++";
dic.city = "Воронеж";
dic.publisher = "Воронежский госуниверситет";
dic.year = 2019;
dic.pages = 165;
```

Заменить эту последовательность одним оператором, который объявляет и инициализирует элементы структуры.

7. Что определяет следующий оператор?

```
struct
{
    char A[20];
    int value;
} BB, CC, EE[9], *ptr;
```

8. Каково имя у соответствующего структурного типа?

```
struct
{
    char A[20];
    int value;
} BB, CC, EE[9], *ptr;
```

9. Описана структура

```
struct
{
    char A[20];
    int value;
} BB, CC, EE[9], *ptr;
```

Что входит в каждый элемент массива EE[0], EE[1], ..., EE[8]?

10. Описана структура

```
struct
{
    char A[20];
    int value;
} BB, CC, EE[9], *ptr;
```

Написать оператор для ввода значения переменной value структуры EE[7].

11. Описана структура

```
struct
{
    char A[20];
    int value;
} BB, CC, EE[9], *ptr;
```

Каков тип переменных с именем value из структур EE[0], EE[1], ..., EE[8]?

12. Дано определение структурного типа, совмещенное с определением конкретных структур этого типа

```
struct PRIMER
```

```
{  
    char* A;  
    long summa;  
} B, C, D;
```

Найти ошибки в следующих высказываниях:

- a) структуры B, C, D имеют одинаковое внутреннее строение;
- b) тип структуры B PRIMER;
- c) PRIMER.summa = 100L;
- d) D.summa = 100L.

13. Определить общий вид указателя на структуры.

14. Что может быть значением каждого указателя на структуру?

15. Пусть определен структурный тип card

```
struct card
```

```
{  
    char *author;  
    char *title;  
    char *city;  
    char *publisher;  
    int year;  
    int pages;  
};
```

и структура rec

```
card rec;
```

Что делает следующий оператор?

```
card *ptrcard = &rec;
```

16. Написать два формата выражения для доступа к элементам структуры, с которой в данный момент связан указатель.

17. Пусть дано

```
struct card
```

```
{  
    char *author;  
    char *title;  
    char *city;  
    char *publisher;  
    int year;  
    int pages;  
};
```

```
card rec5;
```

```
card *ptrcard = &rec5;
```

Для чего предназначены следующие три выражения

```
(*ptrcard).pages  
ptrcard->pages  
rec5.pages
```

18. Написать определение общего вида ссылки для структур.

19. Дано определение структурного типа

```
struct PRIMSTR  
{  
    char *name;  
    long sum;  
} A, B, C;
```

a) верно ли введены следующие ссылки на структуры A, B?

```
PRIMSTR& refA = A;
```

```
PRIMSTR& refB(B);
```

b) заменить на эквивалентные выражения

```
A.sum
```

```
*B.name
```

```
A.name
```

20. Дана программа, в которой определена структура STR и введены значения адресов ее элементов.

```
#include <iostream>  
using std::cout;
```

```
int main()  
{  
    struct  
    {  
        long L;  
        int i1, i2;  
        char c[4];  
    } STR = {10L, 20, 30, 'a', 'b', 'c', 'd'};  
    cout << "\nsizeof(STR) = " << sizeof(STR);  
    cout << "\n&STR.i1 = " << &STR.i1;  
    cout << "\n&STR.i2 = " << &STR.i2;  
    cout << "\n&STR.c = " << &STR.c;  
    return 0;  
}
```

Каков результат выполнения программы, если в результате работы оператора

```
cout << "\n&STR.c = " << &STR.c;
```

было выведено

```
&STR.c = 0x8d8000ff4 ?
```

21. Из приведенных ниже определений структур выбрать правильные и указать причину ошибочных

a) struct cor { cor *pc; long e; };

b) struct mis { mis s; int k; };

c) struct prog { int m; char *a; } begstr;
 struct next { prog beg; float f; };

22. Могут ли поля разных структур иметь одинаковые имена?

23. Дано описание двух структур А и В

```
struct A { int a; double x; };
struct B { A a; double x; } x[5];
```

a) обнулить поле a нулевого элемента массива x структуры В;

b) присвоить полю x последнего элемента массива x его индекс.

24. Описать структуру complex из двух полей, проинициализировав двумерный массив compl[2][3], каждый элемент которого имеет тип complex. Значения элемента массива compl это номер строки и столбца.

25. Перечислить операции, которые можно выполнять над структурами целиком, а не по отдельным полям.

26. С помощью какой операции можно найти размер структуры.

27. Описана и проинициализирована структура

```
struct magazin
{
    char *title;
    int year;
    int number;
};
magazin *pm, m = { "Nature", 2018, 15 };
```

Разместить структуру в свободной динамической памяти.

28. Описана и проинициализирована структура

```
struct magazin
{
    char *title;
    int year;
    int number;
};
```

magazin *pm, m = { "Nature", 2018, 15 };

Разместить массив из 10 структур в свободной динамической памяти.

29. Дано описание структуры store

```
struct store
{
    char arr[3][2], var;
    char *ref;
    float (*ptr)[4];
};
```

store *Ref;

store Str;

- a) полю var структуры Str присвоить символ A;
- b) первому элементу первой строки массива arr присвоить 5;
- c) последнему элементу второй строки поля arr указателя ref присвоить символ D;
- d) содержимому поля var указателя ref присвоить символ X.

9.1.2. Ответы и решения

1. Структура.

2. В отличие от массивов, все элементы которых однотипны, структуры могут содержать элементы разных типов.

3.

```
struct student
{
    char name[30];
    char facult[10];
    int kurs;
    char gruppa[3];
    int stip;
};
```

4. Имя структурного типа можно не задавать, если в программе используются только те переменные, имена которых перечислены в описании структуры.

Имя структурного типа не сохранено для других переменных, имена которых не перечислены в описании структуры.

5.

```
struct card
{
    char *author; // Фамилия И.О. автора книги
    char *title;  // Заголовок книги
    char *city;   // Место издания
    char *publisher; // Издательство
    int year;     // Год издания
    int pages;    // Количество страниц
};
```

6.

```
card dic = {"Ускова О.Ф., Каплиева Н.А.", "Начала структурного  
программирования на языке C++", "Воронеж", "Воронежский  
госуниверситет", 2019, 165};
```

7. Две структуры с именами ВВ, СС, массив структур с именем ЕЕ, указатель ptr на структуру.

8. Имени нет.

9. Массив `char A[20]` и целочисленная переменная `value`.
10. `cin >> EE[7].value;`
11. `int`
12. Ошибочная конструкция `PRIMER.summa = 100L;`
13. Имя_структурного_типа *имя_указателя_на_структуру.
14. Адрес структуры того же типа, т. е. номер байта, с которого структура размещается в памяти.
15. Определяет указатель `ptrcard`, которому с помощью инициализации присваивается значение адреса структуры типа `card`.
16. имя_указателя \rightarrow имя_элемента_структуры
 (*имя_указателя).имя_элемента_структуры
17. Все они именуют один и тот же элемент `int pages` конкретной структуры `rec5`, имеющий тип `card`.
18. имя_структурного_типа
 &имя_ссылки_на_структуру_инициализатор
19.
 - a) верно;
 - b) `A.sum` эквивалентно `refA.sum`;
 *`B.name` эквивалентно *`refB.name`;
`A.name` эквивалентно `refA.name`.
20.


```
sizeof(STR) = 16
&STR.i1 = 0x8d800ff8
&STR.i2 = 0x8d800ffa
&STR.c = 0x8d800ffc
```
21.
 - a) правильно, так как элементом определяемой структуры может быть указатель на структуру определенного типа;
 - b) ошибка, так как элемент структуры не может иметь тот же самый тип, что и определяемый структурный тип;
 - c) верно, так как элементом определяемой структуры может быть структура, тип которой уже определен.
22. Поля разных структур могут иметь одинаковые имена, так как у них разная область видимости.
23.
 - a) `x[0].a.a = 0;`
 - b) `x[4].x = 4;`
24.


```
struct complex
{
    float real, im;
} compl[2][3] =
```

```

{
    { { 0,0 }, { 0,1 }, { 0,2 } },
    { { 1,0 }, { 1,1 }, { 1,2 } }
};
25. Только одна операция присваивания.
26. sizeof(имя_структуры)
27.
pm = new magazin;
*pm = m;
pm->title = "Nature";
. . .
. . .
delete pm;
28.
pm = new magazin[10];
pm[0].title = "Nature";
. . .
. . .
delete [] pm;
29.
a) Str.var = 'A';
b) Str.arr[1][1] = '5';
c) Ref->arr[2][2] = 'D';
d) Ref->var = 'X';

```

9.2. Задания с решениями на составление программ

9.2.1. Условия заданий

1. Разработать структуру базы данных, содержащую информацию о жилых домах микрорайона:

<i>Название поля</i>	<i>Формат</i>
номер микрорайона RegNum	беззнаковое короткое целое
название улицы Street	строка из 50 символов
номер дома HouseNum	беззнаковое короткое целое
количество этажей MaxFloorNum	беззнаковое короткое целое
число квартир MaxFloatNum	беззнаковое короткое целое

информация о наличии автостоянки Parking	логическое значение
---	---------------------

Заполнить объект разработанной структуры своими данными.

2. К финалу конкурса красоты мисс-ВГУ допущено три участницы. Информация о каждой из них содержит следующие данные:

- номер участницы;
- рост;
- вес;
- параметры: окружность груди, талии и бедер. Например, параметры заданы 90 60 90.

Составить программу для определения номера участницы с минимальным весом.

3. Даны координаты N точек пространства. Найти радиус наименьшего шара, внутрь которого попадают все эти точки.

4. Сформировать массив из структур, содержащих следующие сведения о студентах:

- фамилия_имя_отчество;
- курс;
- факультет;
- результаты зачетной сессии;
- результаты экзаменационной сессии.

Напечатать фамилию, имя, отчество первокурсников факультета ПММ, у которых сданы все зачеты.

5. На плоскости заданы координаты вершин N треугольников, раскрашенных в цвета red, green, blue, yellow, black. Сколько среди них равносторонних треугольников цвета green.

9.2.2. Решения

```

1.
struct HOUSE
{
    unsigned short RegNum;
    char Street[51]; // с учетом '\0'
    unsigned short MaxFloorNum, MaxFloatNum;
    bool Parking;
};

HOUSE MyHouse;
```



```

2.
struct miss
{
    int num;
    int rost;
    float ves;
    char param[9]; // с учетом '\0'
};

miss A = { 16, 173, 57.3, "90 60 90" };
miss B = { 28, 172, 55.6, "89 58 90" };
miss C = { 5, 175, 52.7, "88 62 90" };
int nomer;
float min;

if (A.ves < B.ves)
{
    min = A.ves;
    nomer = A.num;
}
else
{
    min = B.ves;
    nomer = B.num;
}
if (C.ves < min)
    nomer = C.num;
cout << "Участница с номером " << nomer << " имеет минимальный
вес";

```

```

3.
struct toch
{
    float x, y, z;
};

int N;
cout << "Введите количество точек" << endl;
cin >> N;
toch *a = new toch[N];
cout << "Введите координаты " << N << "точек" << endl;
for (int i = 0; i < N; i++)
    cin >> a[i].x >> a[i].y >> a[i].z;
float min = a[0].x*a[0].x + a[0].y*a[0].y + a[0].z*a[0].z;

```

```

for (int i = 1; i < N; i++)
{
    float r = a[i].x*a[i].x + a[i].y*a[i].y + a[i].z*a[i].z;
    if (r < min)
        min = r;
}
min = sqrt(min);
cout << "радиус наименьшего шара = " << min;

```

```

4.
struct name
{
    char fam[30],
        im[15],
        ot[20];
};
struct stud
{
    name men;
    int kurs;
    char facultet[12];
    bool zachet[10];
    int exz[5];
};
int N;
cout << "Введите количество студентов" << endl;
cin >> N;
stud *a = new stud[N];
cout << "Введите информацию о студентах " << endl;
for (int i = 0; i < N; i++)
{
    cin >> a[i].men.fam >> a[i].men.im >> a[i].men.ot;
    cin >> a[i].kurs >> a[i].facultet;
    for (int j = 0; j < 10; j++)
        cin >> a[i].zachet[j];
    for (int k = 0; k < 5; k++)
        cin >> a[i].exz[k];
}
for (int i = 0; i < N; i++)
{
    if (a[i].kurs == 1 && strcmp(a[i].facultet, "ПММ")==0)
    {
        bool t = true;
        int j = 0;

```

```

        while (t && j < 10)
        {
            t = a[i].zachet[j];
            j++;
        }
        if (t)
            cout << a[i].men.fam << ' ' << a[i].men.im << ' '
<< a[i].men.ot << endl;
    }
}

```

5.

```
enum color {red, green, blue, yellow, black};
```

```
struct ver
{
    float x, y;
};
```

```
struct tre
{
    ver A, B, C;
    color car;
};
```

```
int N;
cout << "Введите количество треугольников" << endl;
cin >> N;
tre *a = new tre[N];
float AB, AC, BC;
cout << "Ввести информацию о каждом треугольнике" << endl;
int ccolor;
for (int i = 0; i < N; i++)
{
    cin >> ccolor;
    switch (ccolor)
    {
        case 0:
            a[i].car = red;
            break;
        case 1:
            a[i].car = green;
            break;
        case 2:
            a[i].car = blue;

```

```

        break;
    case 3:
        a[i].car = yellow;
        break;
    case 4:
        a[i].car = black;
        break;
    }
    cin >> a[i].A.x >> a[i].A.y;
    cin >> a[i].B.x >> a[i].B.y;
    cin >> a[i].C.x >> a[i].C.y;
}
int kol = 0;
for (int i = 0; i < N; i++)
{
    AB = (a[i].A.x - a[i].B.x)*(a[i].A.x - a[i].B.x) +
        (a[i].A.y - a[i].B.y)*(a[i].A.y - a[i].B.y);
    AC = (a[i].A.x - a[i].C.x)*(a[i].A.x - a[i].C.x) +
        (a[i].A.y - a[i].C.y)*(a[i].A.y - a[i].C.y);
    BC = (a[i].B.x - a[i].C.x)*(a[i].B.x - a[i].C.x) +
        (a[i].B.y - a[i].C.y)*(a[i].B.y - a[i].C.y);
    if (AB == AC && AB == BC && a[i].car == green)
        kol++;
}
cout << "Равносторонних треугольников цвета green " << kol << "
штук";

```

9.3. Тренировочные задания

1. Составить программу для выполнения четырех арифметических операций (+ - * /) над двумя комплексными числами, представив каждое комплексное число как структуру с двумя полями, соответствующими действительной и коэффициенту при мнимой части комплексного числа.

2. Реализовать четыре арифметические операции (+ - * /) над комплексными числами. Применить эти операции к числам $a + bi$ и $c + di$, когда a (действительная часть) принимает значения $a + h$, где h принимает значения 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6; коэффициент при мнимой части d принимает значения d , $d - 0.1$, $d - 0.2$, $d - 0.3$, $d - 0.4$, $d - 0.5$, $d - 0.6$. Коэффициенты a , b , c , d определяются вводом.

3. К финалу конкурса красоты мисс-ВГУ допущено три участницы. Информация о каждой из них содержит следующие данные:

- номер участницы;
- рост;

- вес;
- параметры: окружность груди, талии и бедер. Например, параметры заданы 90 60 90.

Напечатать вес любой участницы конкурса с минимальным ростом.

4. К финалу конкурса красоты мисс-ВГУ допущено три участницы. Информация о каждой из них содержит следующие данные:

- номер участницы;
- рост;
- вес;
- параметры: окружность груди, талии и бедер. Например, параметры заданы 90 60 90.

Напечатать параметры участницы, вес которой минимален. Если таких участниц несколько, напечатать параметры каждой из них.

5. К финалу конкурса красоты Мисс-ВГУ допущено 30 девушек. Информация о каждой из них содержит следующие данные:

- номер участницы;
- рост;
- вес;
- параметры: окружность груди, талии и бедер.

Напечатать номера каждой из них, параметры окружности груди, талии и бедер которой 90 60 90.

6. К финалу конкурса красоты Мисс-ВГУ допущено 30 девушек. Информация о каждой из них содержит следующие данные:

- номер участницы;
- рост;
- вес;
- параметры: окружность груди, талии и бедер.

Составить одномерный массив из номеров и параметров тех участниц, рост которых превышает минимальный рост всех участниц конкурса.

7. Новогодний подарок содержит конфеты трех сортов, печенье трех сортов и игрушки. Каждый сорт сладостей определяется весом (в кг) и ценой (за 1 кг). Для игрушек известно их количество и цена каждой игрушки. Цена сладостей и игрушек в различных подарках не совпадает.

Сформировать массив из 100 подарков и найти максимальную цену подарка.

8. Новогодний подарок содержит конфеты трех сортов, печенье трех сортов и игрушки. Каждый сорт сладостей определяется весом (в кг) и ценой (за 1 кг). Для игрушек известно их количество и цена каждой игрушки. Цена сладостей и игрушек в различных подарках не совпадает.

С использованием динамического массива сформировать массив из 50 подарков и найти среднюю цену каждого подарка.

9. Новогодний подарок содержит конфеты трех сортов, печенье трех сортов и игрушки. Каждый сорт сладостей определяется весом (в кг) и ценой (за 1 кг). Для игрушек известно их количество и цена каждой игрушки. Цена сладостей и игрушек в различных подарках не совпадает.

С использованием динамического массива сформировать массив из 50 подарков и найти среднюю цену конфет во всех подарках.

10. Новогодний подарок содержит конфеты трех сортов, печенье трех сортов и игрушки. Каждый сорт сладостей определяется весом (в кг) и ценой (за 1 кг). Для игрушек известно их количество и цена каждой игрушки. Цена сладостей и игрушек в различных подарках не совпадает.

С использованием динамического массива сформировать массив из 50 подарков и уменьшить на 6 % цену конфет в тех подарках, средняя цена которых выше средней цены всех подарков.

11. Положение шара в пространстве определяется координатами его центра, радиусом и цветом (красный, синий, белый, черный, зеленый).

Сформировать массив из 50 таких шаров и найти количество шаров каждого цвета.

12. Положение шара в пространстве определяется координатами его центра, радиусом и цветом (красный, синий, белый, черный, зеленый).

Сформировать массив из 50 таких шаров и найти максимальный радиус шаров красного цвета.

13. Положение шара в пространстве определяется координатами его центра, радиусом и цветом (красный, синий, белый, черный, зеленый).

Сформировать массив из 150 таких шаров. Найти минимальный радиус шаров зеленого цвета и количество минимальных, если их несколько.

14. Положение шара в пространстве определяется координатами его центра, радиусом и цветом (красный, синий, белый, черный, зеленый).

Сформировать массив из 50 таких шаров и напечатать цвет любого шара, центр которого максимально удален от начала координат.

15. Положение шара в пространстве определяется координатами его центра, радиусом и цветом (красный, синий, белый, черный, зеленый).

Сформировать массив из 50 таких шаров. Напечатать цвет, координаты центра любого шара, центр которого расположен к началу координат ближе всех, и количество таких шаров.

16. Создать массив структур. Каждый элемент массива содержит сведения о данных, входящих в библиографическую карточку:

- Фамилия И. О. автора книги
- Заголовок книги
- Специальность
- Место издания
- Издательство

- Год издания
- Количество страниц.

Так как количество символов для первых пяти полей неизвестно, типом этих полей должен быть указатель.

Сколько книг по специальности «Информатика» издано в текущем году в г. Воронеже.

17. Создать массив структур. Каждый элемент массива содержит сведения о данных, входящих в библиографическую карточку:

- Фамилия И. О. автора книги
- Заголовок книги
- Специальность
- Место издания
- Издательство
- Год издания
- Количество страниц.

Так как количество символов для первых пяти полей неизвестно, типом этих полей должен быть указатель.

Напечатать название любой книги по информатике, изданной в издательстве «Питер», с минимальным количеством страниц.

18. Создать массив структур. Каждый элемент массива содержит сведения о данных, входящих в библиографическую карточку:

- Фамилия И. О. автора книги
- Заголовок книги
- Специальность
- Место издания
- Издательство
- Год издания
- Количество страниц.

Так как количество символов для первых пяти полей неизвестно, типом этих полей должен быть указатель.

Составить массив из заголовков книг по языку C++, изданных в текущем году.

19. Создать массив структур. Каждый элемент массива содержит сведения о данных, входящих в библиографическую карточку:

- Фамилия И. О. автора книги
- Заголовок книги
- Специальность
- Место издания
- Издательство
- Год издания

- Количество страниц.

Так как количество символов для первых пяти полей неизвестно, типом этих полей должен быть указатель.

Удалить из исходного массива структур, содержащих сведения о библиографических карточках, любую книгу по математике с минимальным количеством страниц.

20. Создать массив структур. Каждый элемент массива содержит сведения о данных, входящих в библиографическую карточку:

- Фамилия И. О. автора книги
- Заголовок книги
- Специальность
- Место издания
- Издательство
- Год издания
- Количество страниц.

Так как количество символов для первых пяти полей неизвестно, типом этих полей должен быть указатель.

Удалить из исходного массива структур, содержащих сведения о библиографических карточках, информацию о всех книгах по информатике, изданные в прошлом веке.

21. Создать массив структур. Каждый элемент массива содержит сведения о данных, входящих в библиографическую карточку:

- Фамилия И. О. автора книги
- Заголовок книги
- Специальность
- Место издания
- Издательство
- Год издания
- Количество страниц.

Так как количество символов для первых пяти полей неизвестно, типом этих полей должен быть указатель.

Если у автора с заданной фамилией есть несколько книг с одинаковым названием, но изданных в разные годы, оставить в массиве информацию только об одной книге этого автора, которая издана позже всех.

22. Создать массив структур. Каждый элемент массива содержит сведения о данных, входящих в библиографическую карточку:

- Фамилия И. О. автора книги
- Заголовок книги
- Специальность
- Место издания
- Издательство

- Год издания
- Количество страниц.

Так как количество символов для первых пяти полей неизвестно, типом этих полей должен быть указатель.

Сколько книг издано у автора, Фамилия И. О. которого вводятся.

23. Создать массив структур. Каждый элемент массива содержит сведения о данных, входящих в библиографическую карточку:

- Фамилия И. О. автора книги
- Заголовок книги
- Специальность
- Место издания
- Издательство
- Год издания
- Количество страниц.

Так как количество символов для первых пяти полей неизвестно, типом этих полей должен быть указатель.

Напечатать название любой специальности (информатика, математика, физика), по которой в текущем году было больше всего книг.

24. Пусть даны комплексное число z (пара вещественных чисел) и вещественное число $\varepsilon > 0$. Вычислить с точностью ε значение следующей комплексной функции:

$$a) e^z = 1 + \frac{z}{1!} + \frac{z^2}{2!} + \dots + \frac{z^n}{n!} + \dots;$$

$$b) \operatorname{sh} z = z + \frac{z^3}{3!} + \frac{z^5}{5!} + \dots + \frac{z^{2n+1}}{(2n+1)!} + \dots;$$

$$c) \operatorname{ch} z = z + \frac{z^2}{2!} + \frac{z^4}{4!} + \dots + \frac{z^{2n}}{(2n)!} + \dots;$$

$$d) \sin z = z - \frac{z^3}{3!} + \frac{z^5}{5!} - \dots + \frac{(-1)^n z^{2n+1}}{(2n+1)!} + \dots;$$

$$e) \cos z = z - \frac{z^2}{2!} + \frac{z^4}{4!} - \dots + \frac{(-1)^n z^{2n}}{(2n)!} + \dots;$$

$$f) \ln(1+z) = z - \frac{z^2}{2} + \frac{z^3}{3} - \dots + \frac{(-1)^{n-1} z^n}{n} + \dots \quad (|z| < 1);$$

$$g) \operatorname{arctg} z = z - \frac{z^3}{3} + \frac{z^5}{5} - \dots + \frac{(-1)^{n-1} z^{2n+1}}{2n+1} + \dots \quad (|z| < 1).$$

25. В массиве содержится информация о зимней сессии студентов 3-го курса. Сведения о каждом студенте (максимальное число студентов – 150) содержат следующие данные:

- фамилия (до 12 символов);
- номер группы (от 1 до 7);
- оценки по трем предметам (функциональный анализ, численные методы и экономика).

Напишите программу, которая вводит эту информацию и печатает следующие данные:

- а) фамилии студентов, имеющих задолженность хотя бы по одному предмету;
- б) процент студентов, сдавших все экзамены на 4 и 5;
- с) название предмета, который был сдан лучше всего;
- д) номера групп в порядке убывания средней успеваемости их студентов.

26. Сведения о студентах вуза записаны в массиве и содержат следующую информацию:

- фамилия, имя, отчество;
- пол (М или Ж);
- возраст (от 16 до 35);
- курс (от 1-го до 5-го).

Напишите программу, которая вводит эту информацию и печатает следующие данные:

- а) номер курса, на котором наибольший процент мужчин;
- б) самые распространенные мужские и женские имена;
- с) фамилии (в алфавитном порядке) и инциалы всех студенток, возраст и отчества которых являются одновременно самыми распространенными.

27. Создать массив структур, каждый элемент которого содержит информацию:

- фамилия, имя, отчество;
- адрес (индекс, город, улица, дом, квартира);
- телефон;
- хобби (музыка, спорт, компьютер).

а) напечатайте фамилии всех людей, у которых имеется компьютер IBM PC XT;

б) посчитайте количество людей, которые увлекаются музыкой;

с) напечатайте данные обо всех различных музыкальных инструментах, которые есть в массиве структур.

9.4. Задания для самостоятельной работы на компьютере

1. Сформировать массив из N структур, каждая из которых содержит информацию о сотрудниках предприятия:

- фамилия, имя, отчество;
- название отдела;
- должность;
- год рождения;
- зарплата за каждый месяц;
- пол.

Увеличить зарплату за январь месяц всех сотрудников непенсионного возраста одного из отделов, название которого вводится, на 15 %.

2. Сформировать массив из N структур, каждая из которых содержит информацию о сотрудниках предприятия:

- фамилия, имя, отчество;
- название отдела;
- должность;
- год рождения;
- зарплата за каждый месяц;
- пол.

Сформировать для каждого сотрудника поле SR – средняя зарплата сотрудника.

3. Сформировать массив из N структур, каждая из которых содержит информацию о сотрудниках предприятия:

- фамилия, имя, отчество;
- название отдела;
- должность;
- год рождения;
- зарплата за каждый месяц;
- пол.

Найти среднюю зарплату предприятия в феврале месяце и в связи с инфляцией уменьшить зарплату отдела «Связь с общественностью» на 0.6 % в марте месяце.

4. Сформировать массив из N структур, каждая из которых содержит информацию о сотрудниках предприятия:

- фамилия, имя, отчество;
- название отдела;
- должность;
- год рождения;
- зарплата за каждый месяц;
- пол;
- бонус.

Из зарплаты всех сотрудников отдела «Информационная безопасность» за второе полугодие вычесть 0.5 % от бонуса.

5. Сформировать массив из N структур, каждая из которых содержит информацию о сотрудниках предприятия:

- фамилия, имя, отчество;
- название отдела;
- должность;
- год рождения;
- зарплата за каждый месяц;
- пол.

Напечатать фамилию, имя, отчество всех сотрудников отделов «Системная администрация» и «Начисление заработной платы» с минимальной зарплатой в апреле месяце и количество таких сотрудников.

6. Сформировать массив из N структур, каждая из которых содержит информацию о сотрудниках предприятия:

- фамилия, имя, отчество;
- название отдела;
- должность;
- год рождения;
- зарплата за каждый месяц;
- пол.

Удалить из массива информацию о всех сотрудниках-пенсионерах и напечатать полученный массив.

7. Сформировать массив из N структур, каждая из которых содержит информацию о сотрудниках предприятия:

- фамилия, имя, отчество;
- название отдела;
- должность;
- год рождения;
- зарплата за каждый месяц;
- пол.

Составить новый массив из фамилий, имен, отчеств тех сотрудников отдела «Связь с общественностью», среднегодовая зарплата которых ниже среднегодовой зарплаты всех сотрудников.

8. Сформировать массив из N структур, каждая из которых содержит информацию о сотрудниках предприятия:

- фамилия, имя, отчество;
- название отдела;
- должность;
- год рождения;
- зарплата за каждый месяц;
- пол.

Составить новый массив из фамилий, имен, отчеств тех сотрудников отдела «Снабжение», средняя зарплата за первое полугодие которых выше минимальной среднегодовой зарплаты всего предприятия.

9. Сформировать массив из N структур, каждая из которых содержит информацию о сотрудниках предприятия:

- фамилия, имя, отчество;
- название отдела;
- должность;
- год рождения;
- зарплата за каждый месяц;
- пол.

Составить новый массив из фамилий, имен, отчеств, должности, пола, названия отдела тех сотрудников предприятия, у которых хотя бы в одном месяце была нулевая зарплата.

10. Сформировать массив из N структур, каждая из которых содержит информацию о температуре в различных месяцах различных городов России:

- название города;
- расстояние в км от Москвы;
- название месяца;
- три числа, характеризующих измерения температуры (утро, день, вечер) в каждый день месяца.

Напечатать название любого города с максимальной температурой в марте месяце, считая что хотя бы один такой город в сформированном массиве есть.

11. Сформировать массив из N структур, каждая из которых содержит информацию о температуре в различных месяцах различных городов России:

- название города;
- расстояние в км от Москвы;
- название месяца;
- три числа, характеризующих измерения температуры (утро, день, вечер) в каждый день месяца.

Найти максимальную среднедневную температуру в марте месяце и напечатать название всех таких городов.

12. Сформировать массив из N структур, каждая из которых содержит информацию о температуре в различных месяцах различных городов России:

- название города;
- расстояние в км от Москвы;
- название месяца;

- три числа, характеризующих измерения температуры (утро, день, вечер) в каждый день месяца.

Найти максимальную среднемесячную температуру в июне месяце в городах, удаленных от Москвы на более 600 км. Напечатать название этого города или всех таких городов, если их несколько.

13. Сформировать массив из N структур, каждая из которых содержит информацию о температуре в различных месяцах различных городов России:

- название города;
- расстояние в км от Москвы;
- название месяца;
- три числа, характеризующих измерения температуры (утро, день, вечер) в каждый день месяца.

Для г. Воронежа найти наибольший интервал увеличения средней температуры. Напечатать первую и последнюю температуры этого интервала и название месяца.

14. Сформировать массив из N структур, каждая из которых содержит информацию о температуре в различных месяцах различных городов России:

- название города;
- расстояние в км от Москвы;
- название месяца;
- три числа, характеризующих измерения температуры (утро, день, вечер) в каждый день месяца.

Составить массив из названий городов, их расстояний от Москвы и месяцев с максимальной среднемесячной температурой.

15. Сформировать массив из N структур, каждая из которых содержит информацию о температуре в различных месяцах различных городов России:

- название города;
- расстояние в км от Москвы;
- название месяца;
- три числа, характеризующих измерения температуры (утро, день, вечер) в каждый день месяца.

Для любого города, наиболее удаленного от Москвы, найти среднемесячную температуру. Напечатать название города, месяца, расстояния от Москвы и среднемесячную температуру.

16. Сформировать массив из N структур, каждая из которых содержит информацию о температуре в различных месяцах различных городов России:

- название города;

- расстояние в км от Москвы;
- название месяца;
- три числа, характеризующих измерения температуры (утро, день, вечер) в каждый день месяца.

Удалить из массива элементы, у которых совпадают названия городов, оставив только один из таких элементов.

17. Сформировать массив из N структур, каждая из которых содержит информацию о температуре в различных месяцах различных городов России:

- название города;
- расстояние в км от Москвы;
- название месяца;
- три числа, характеризующих измерения температуры (утро, день, вечер) в каждый день месяца.

Составить новый массив структур, включив в него элементы исходного массива, относящиеся к апрелю месяцу.

18. Сформировать массив из N структур, каждая из которых содержит информацию о температуре в различных месяцах различных городов России:

- название города;
- расстояние в км от Москвы;
- название месяца;
- три числа, характеризующих измерения температуры (утро, день, вечер) в каждый день месяца.

Составить новый массив структур, включив в него элементы исходного массива, относящиеся к сентябрю месяцу городов Центрального Черноземья.

19. Сформировать массив из N структур, каждая из которых содержит информацию о температуре в различных месяцах различных городов России:

- название города;
- расстояние в км от Москвы;
- название месяца;
- три числа, характеризующих измерения температуры (утро, день, вечер) в каждый день месяца.

Составить новый массив структур из элементов исходного массива, включив в него города, удаленные от Москвы дальше Воронежа, и заменив три измерения температуры (утро, день, вечер) одной средней. В новом массиве структур каждый город должен встречаться только один раз.

20. Пусть дан массив, содержащий сведения об экспортируемых товарах:

- наименование товара;
- страна, импортирующая товар;
- объем поставляемой партии в штуках.

Составить массив стран, в которые экспортируется товар (наименование задается пользователем), и общий объем его экспорта.

21. Пусть дан массив, содержащий сведения об игрушках:

- название игрушки (например, кукла, кубики, мяч, конструктор и т. д.);
- стоимость в копейках;
- возрастные границы детей, для которых игрушка предназначена (например, для детей от 2 до 5 лет).

Получите информацию:

- а) о названиях игрушек, цена которых не превышает 4 рубля и которые подходят детям до 5 лет;
- б) о цене самого дорогого конструктора;
- с) о названиях наиболее дорогих игрушек (цена которых отличается от цены самой дорогой игрушки не более чем на 1 рубль);
- д) о названиях игрушек, которые подходят как детям 4 лет, так и детям 10 лет;
- е) о ценах всех кубиков;
- ф) можно ли подобрать любую, игрушку, кроме мяча, подходящую ребенку 3 лет, и дополнительно мяч так, чтобы суммарная стоимость игрушек не превышала 5 рублей;
- г) имеется ли мяч ценой 2 рубля 50 копеек, предназначенный детям от 3 до 8 лет; если да, то вывести сведения об этой игрушке.

22. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Пусть дан массив, содержащий информацию о багаже нескольких пассажиров:

А. Найдите багаж, средний вес каждой вещи которого отличается не более чем на 0,3 кг от среднего веса вещей всех пассажиров.

В. Найдите число пассажиров, имеющих более двух вещей, и число пассажиров, количество вещей которых превосходит среднее число вещей.

С. Определите, имеются ли два пассажира, багажи которых совпадают по числу вещей и отличаются по весу не более чем на 0,5 кг.

Д. Выясните, имеется ли пассажир, багаж которого превышает багаж каждого из остальных пассажиров и по количеству вещей и по весу.

Е. Выясните, имеется ли пассажир, багаж которого состоит из одной вещи весом менее 30 кг.

Ф. Найдите сведения о багаже, число вещей в котором не меньше, чем в любом другом багаже, а вес вещей не больше, чем в любом другом багаже с этим же количеством вещей.

Г. Упорядочьте сведения о багаже по невозрастанию веса багажа.

Н. Удалите сведения о багаже, общий вес вещей в котором меньше чем 10 кг.

23. Пусть дан массив, содержащий сведения о книгах. Сведения о каждой книге:

- фамилия автора;
- название;
- год издания.

А. Найдите названия книг данного автора, изданных с 1977 года;

В. Определите, имеется ли книга с названием «информатика», если да, то выведите фамилию автора и год издания книги, если таких книг несколько, то выведите имеющиеся сведения обо всех этих книгах.

24. Пусть даны два массива $M1$ и $M2$. Массив $M1$ – это инвентарный массив, содержащий сведения о том, сколько изделий каких видов хранится на складе (вид продукции задается ее номером). Массив $M2$ – это вспомогательный массив, содержащий сведения о том, на сколько уменьшилось или увеличилось количество изделий по некоторым видам продукции одного вида. Обновите инвентарный массив на основе вспомогательного.

25. Пусть дан массив, содержащий сведения о претендентах на руку и сердце. Сведения могут содержать следующие данные:

- фамилия, имя, отчество;
- возраст;
- рост;
- цвет глаз;
- цвет волос;
- зарплата;
- наличие квартиры;
- наличие машины.

Сформулируйте несколько критериев, по которым будут отбираться претенденты. Напишите программу, предназначенную для ввода данных о претендентах и печати сведений о наиболее подходящих кандидатах в зависимости от того или иного критерия.

26. Пусть дан массив, содержащий сведения о продаваемых квартирах. Сведения могут содержать следующую информацию:

- общая площадь;
- жилая площадь;
- площадь кухни;
- наличие санузла и его характеристики (совмещенный или нет);
- наличие лоджии;
- панельный или кирпичный дом;
- общее количество этажей в доме;

- этаж, на котором расположена квартира;
- район, в котором находится квартира;
- адрес;
- стоимость.

Сформулируйте несколько критериев, по которым можно отобрать ту или иную квартиру для покупки и, основываясь на этих критериях, выведите сведения о ней. Если подходящих квартир несколько, то выведите сведения обо всех.

27. Информация о студентах вуза включена в массив структур, каждая из которых имеет поля:

- фамилия, имя, отчество;
- факультет;
- курс;
- группа;
- специальность;
- результаты сессии:
 - оценки 6 экзаменов;
 - результаты 9 зачетов.

Факультету выделили стипендиальный фонд в размере F рублей. Стипендия студентам начисляется по правилам:

- для получения стипендии должны быть сданы все зачеты и отсутствовать неудовлетворительные оценки по результатам экзаменов;
- студенты, имеющие только одну удовлетворительную оценку, а остальные оценки «отлично» и «хорошо», могут рассчитывать на стипендию $S3$, размер которой вводится;
- стипендия хорошистов $S2$ вычисляется по формуле $S2 = S3 + 0.5 * S3$;
- стипендия отличников вычисляется по формуле $S1 = S2 + 0.5 * S2$.

Каков размер стипендии у студентов каждой категории и каков остаток стипендиального фонда, идущего на материальную помощь.

28. Информация о студентах вуза включена в массив структур, каждая из которых имеет поля:

- фамилия, имя, отчество;
- факультет;
- курс;
- группа;
- специальность;
- результаты сессии:
 - оценки 6 экзаменов;
 - результаты 9 зачетов.

Составить новый массив структур из информации об отличниках факультета ПММ. Каждый элемент массива должен быть структурой, имеющей поля:

- фамилия, имя, отчество;
- курс;
- группа;
- специальность.

Элементы нового массива упорядочить по курсам.

29. Информация о студентах вуза включена в массив структур, каждая из которых имеет поля:

- фамилия, имя, отчество;
- факультет;
- курс;
- группа;
- специальность;
- результаты сессии:
 - оценки 6 экзаменов;
 - результаты 9 зачетов.

Составить новый массив структур из информации о задолжниках факультета ПММ, обучающихся по специальности «Защита информации». Каждый элемент массива должен быть структурой, имеющей поля:

- фамилия, имя, отчество;
- курс;
- группа;
- специальность.

Для каждого задолжника указать количество несданных зачетов и экзаменов.

30. Информация о студентах вуза включена в массив структур, каждая из которых имеет поля:

- фамилия, имя, отчество;
- факультет;
- курс;
- группа;
- специальность;
- результаты сессии:
 - оценки 6 экзаменов;
 - результаты 9 зачетов.

По каждому факультету (по результатам экзаменов) вычислить средний балл успеваемости. Максимальное количество факультетов 16.

31. Информация о студентах вуза включена в массив структур, каждая из которых имеет поля:

- фамилия, имя, отчество;
- факультет;
- курс;
- группа;
- специальность;
- результаты сессии:
 - оценки 6 экзаменов;
 - результаты 9 зачетов.

Для каждого студента вычислить средний балл его успеваемости (по результатам экзаменов), разместить который в новом поле исходного массива. Составить массив, содержащий следующие сведения о студентах 2 курса 4 группы факультета ПММ:

- фамилия, имя, отчество;
- средний балл успеваемости.

32. Пусть дан массив, содержащий сведения о жителях, обслуживаемых данной поликлиникой. Сведения содержат следующую информацию:

- фамилия, имя, отчество жителя;
- адрес;
- место работы;
- наличие прививки от дифтерии;
- дата прохождения последней флюорографии.

Напечатайте фамилии и адреса тех жильцов, которые не сделали прививку от дифтерии.

33. Пусть дан массив, содержащий сведения о жителях, обслуживаемых данной поликлиникой. Сведения содержат следующую информацию:

- фамилия, имя, отчество жителя;
- адрес;
- место работы;
- наличие прививки от дифтерии;
- дата прохождения последней флюорографии.

Сформируйте массив с данными о жильцах, у которых на данный момент просрочена дата флюорографии (дата просрочена, если с ее момента прошло больше года).

34. Сведения о каждой машине включают в себя следующую информацию:

- модель (строка длиной 10 символов);
- номер (код региона, цифровой код, буквенный код);
- цвет;

- сведения о владельце (фамилия, имя, отчество);
- дата последнего техосмотра.

Выберите данные обо всех машинах, не прошедших техосмотр в текущем году.

35. Сведения о каждой машине включают в себя следующую информацию:

- модель (строка длиной 10 символов);
- номер (код региона, цифровой код, буквенный код);
- цвет;
- сведения о владельце (фамилия, имя, отчество);
- дата последнего техосмотра.

Напечатайте информацию обо всех владельцах машин «Волга» черного цвета.

36. Сведения о каждой машине включают в себя следующую информацию:

- модель (строка длиной 10 символов);
- номер (код региона, цифровой код, буквенный код);
- цвет;
- сведения о владельце (фамилия, имя, отчество);
- дата последнего техосмотра.

По номеру машины выдайте всю имеющуюся информацию о ней.

Глава 10. Объединения

10.1. Вопросы, задачи, программы

1. Верно ли, что описание объединения не распределяет память, а представляет шаблон для будущего объявления переменных?

2. Описать объединение с именем `distance`, содержащее два элемента:

- расстояние в милях;
- расстояние в метрах.

3. Объявить двумя способами три переменных объединения `japan`, `Germany`, `france`, имя которого `distance`, содержащее два элемента

- расстояние в милях;
- расстояние в метрах.

4. Чем объединение отличается от структуры?

5. Начертить участок памяти, выделенный структуре `STR`

```
struct
{
    int i;
    char ch;
    long l;
} STR;
```

6. Начертить участок памяти, выделенный объединению `UNI`

```
union
{
    int i;
    char ch;
    long l;
} UNI;
```

7. Чему соответствуют размеры элементов объединения?

8. Чему соответствует размер переменной типа структура?

9. Чему равен размер участка памяти, выделяемого для объединения?

10. Верно ли, что изменение любого поля объединения меняет значения других полей?

11. Дана программа

```
int main()
{
    union distance
    {
        int miles;
        long meters;
    } walk;

    walk.miles = 5;
```

```

        cout << "пройденное расстояние в милях: " << walk.miles <<
endl; // оператор_1
        walk.meters = 10000; // оператор_2
        cout << "пройденное расстояние в метрах: " << walk.meters
<< endl;

        return 0;
}

```

А. Сформулировать условие задачи.

В. Каков результат работы программы?

С. Можно ли поменять местами *оператор_1* и *оператор_2* ?

12. В каких случаях целесообразно использовать объединения?

13. Дано определение структуры и определение объединения:

```

struct { long l1; int i1, i2; char ch[6]; } STR;
union { long l1; int i1, i2; char ch[6]; } UNI;

```

А. Что общего у STR и UNI?

В. Существенное отличие STR и UNI?

14. Дана программа

```

int main()
{
    union
    {
        long l;
        int i1, i2;
        char ch[6];
    } UNI = { 10L };

    cout << "\nsizeof(UNI) = " << sizeof(UNI) << hex;
    cout << "\n&UNI.l = " << &UNI.l;
    cout << "\n&UNI.i2 = " << &UNI.i2; // оператор 3
    cout << "\n&UNI.ch = " << &UNI.ch;
    cout << "\nsizeof(UNI.i1) = " << sizeof(UNI.i1);
    cout << "\nsizeof(UNI.l) = " << sizeof(UNI.l);

    return 0;
}

```

Каков результат выполнения программы, если в результате выполнения *оператора 3* будет выведено

&UNI.i2 = 0x8d7d0ffc ?

15. Дана программа

```

int main()
{
    enum paytype { card, check };
    paytype ptype;

```

```

union payment
{
    char card[25];
    long check;
} info;
Оператор_4
Оператор_5
switch (ptype)
{
case card:
    cout << "оплата по карте: " << info.card;
    оператор_6
case check:
    оператор_7
    оператор_8
}

return 0;
}

```

Что должно быть на месте операторов *оператор_4*, *оператор_5*, *оператор_6*, *оператор_7*, *оператор_8* ?

16. Какой английский термин заменяет русские слова «структурный тип»?

17. Как перевести (приминительно к информатике) на английский язык сочетание слов «объединяющий тип» ?

18. Написать общий вид объединения.

19. Пусть введен тип объединения

```

union tip
{
    double a;
    long B[4];
    int A[6];
};

```

Объявить:

a) объединение tC;

b) массив объединений tD из 5 элементов;

c) указатель на объединение;

d) ссылку на объединение.

20. Написать в общем виде все способы обращения к элементу объединения.

21. Постарайтесь продолжить следующую фразу:

«Если элементы объединения имеют одинаковый тип и одинаковую длину, а отличаются только именами, то использование объединений было бы аналогично...»

22. Пусть введено объединение

```
union
{
    float a;
    unsigned long b;
} FUL;
```

Можно ли занести в участок памяти, выделенный для объединения FUL, вещественное число

FUL.a = 2106.12;

а затем рассматривать код его внутреннего представления как некоторое беззнаковое длинное целое?

23. Пусть определено объявление

```
union
{
    float Fa;
    unsigned long b;
    char C[5];
} Fbc;
```

и в участок памяти, выделенной для объединения, занесено вещественное число

Fbc.Fa = 1917.102017;

Как можно получить:

а) значение кода его внутреннего представления?

б) значение кодов, находящихся в отдельных байтах ?

24. Дана программа

```
int main()
{
    std::setlocale(LC_ALL, "Russian_Russia.1251");

    enum paytype { CARD, CHECK };
    paytype ptype;
    union payment
    {
        char card[31];
        long check;
    } info;
    оператор_4
    оператор_5
    switch (ptype)
    {
```

```

    case CARD:
        cout << "оплата по карте: " << info.card;
        break;
    case CHECK:
        cout << "оплата чеком: " << info.check;
        break;
    }
}

```

Заменить *оператор_4* и *оператор_5*, которые присваивают значения `info` и `ptype`, одним оператором присваивания, внося объединение в структуру.

25. Какое объявление называется анонимным?

26. Пусть дана программа, содержащая объявление объединения `distance` (расстояние в милях и метрах), в котором присваиваются и выводятся на экран некоторые значения полей этого объединения.

```

int main()
{
    union distance
    {
        int miles;
        long meters;
    } rasst;

    rasst.miles = 100;
    cout << "пройденное расстояние в милях: " << rasst.miles <<
endl;
    rasst.meters = 12000;
    cout << "пройденное расстояние в метрах: " << rasst.meters
<< endl;
}

```

Переделать эту программу, используя анонимное объединение, которое не использует имя объединения и не объявляет переменную объявления.

27. В каких случаях используются анонимные объединения?

28. В чем сходство и различие анонимных объявлений и обычных переменных?

10.2. Ответы и решения

1. Верно.

2.

```

union distance
{
    int miles;

```

```

    long meters;
};

```

3. Первый способ

```

union distance
{
    int miles;
    long meters;
} japan, germany, france;

```

Второй способ

```

union distance
{
    int miles;
    long meters;
};
distance japan, germany, france;

```

4. Способом организации внутренней памяти: все элементы объединения имеют один и тот же начальный адрес.

5.

байт	байт	байт	байт	байт	байт	байт
<i>i</i>		<i>ch</i>	<i>l</i>			

6.

байт	байт	байт	байт
ch			
i			
l			

7. Размеры элементов объединения соответствуют их типам.

8. Суммарной длине всех полей структуры.

9. Максимальной из длин элементов объединения.

10. Верно.

11.

а) В программе описано объединение с именем `distance`, содержащее два элемента. Сначала программа присваивает значение элементу `miles` и выводит это значение. Затем программа присваивает значение элементу `meters` и выводит это значение, которое является расстоянием в метрах.

б) Результат работы программы:

пройденное расстояние в милях: 5

пройденное расстояние в метрах: 10000

с) В отличие от структуры в каждый момент времени в переменной типа объединение хранится только одно значение. Поэтому при присваива-

нии значения элементу объединения происходит перезапись предыдущего присваивания.

Ответственность за правильное использование объединений лежит полностью на программисте. Изменение значения любого поля меняет значение других полей, так как поля объединения накладываются друг на друга, поэтому общий объем занимаемой памяти равен размеру самого большого поля.

Ответ: нельзя поменять местами *оператор_1* и *оператор_2*.

12. Объединения используют в целях экономии памяти в тех случаях, когда известно, что больше одного поля одновременно не требуется.

13.

а) Количество элементов в объединении с именем UNI и их типы совпадают с количеством и типом элементов структуре STR.

б) Все элементы объединения имеют один и тот же начальный адрес.

14.

```
sizeof(UNI) = 8
&UNI.l = 0x8d7d0ffc
&UNI.i2 = 0x8d7d0ffc
&UNI.ch = 0x8d7d0ffc
sizeof(UNI.i1) = 4
sizeof(UNI.l) = 4
```

15. На месте *оператора_4* и *оператора_5* должны быть операторы, присваивающие значения info и ptype.

На месте *оператора_6* и *оператора_7* должен стоять оператор break.

Оператор_7 может иметь вид

```
cout << "оплата чеком: " << info.check;
```

16. tag

17. union

18. union имя_объединяющего_типа {элементы объединения};

19.

а) tip tC;

б) tip tD[5];

в) tip *ptr;

г) tip& ss = tC;

20. Для обращения к элементу объединения можно использовать или уточненное имя

имя_объединения.имя_элемента

или конструкцию, включающую указатель

указатель_на_объединение -> имя_элемента

(*указатель_на_объединение).имя_элемента
или конструкцию, включающую ссылку
ссылка_на_объединение.имя_элемента

21. Применению ссылок.

22. Можно. Основное достоинство объединения – это возможность разных трактовок одного и того же содержимого участка памяти.

23.

a) с помощью уточненного имени

b) Fbc.C[0], Fbc.C[1], Fbc.C[2], Fbc.C[3], Fbc.C[4],
Fbc.C[5]

24.

```
enum paytype { CARD, CHECK };
struct
{
    paytype ptype;
    union payment
    {
        char card[31];
        long check;
    } un;
} info;
. . .
// присваивание значения info
switch (info.ptype)
{
case CARD:
    cout << "оплата по карте: " << info.un.card;
    break;
case CHECK:
    cout << "оплата чеком: " << info.un.check;
    break;
}
```

25. Анонимное объединение представляет собой объединение без имени.

26.

```
union
{
    int miles;
    long meters;
};
```

```
miles = 100;  
cout << "значение в милях: " << miles << endl;  
meters = 12000;  
cout << "значение в метрах: " << meters << endl;
```

27. Язык C++ предоставляет анонимные объединения, чтобы упростить использование элементов объединений, предназначенных для экономии памяти или создания псевдонимов для определенного значения.

28. Программа трактует элементы анонимных объединений как обычные переменные. Различие между элементами анонимных объединений и обычных переменных заключается в том, что при присваивании значения любому из элементов объединения, значение другого элемента теряется.

Глава 11. ФУНКЦИИ

11.1. Контрольные вопросы, задания, упражнения

11.1.1. Тексты заданий

1. Как называется именованное объединение группы операторов?
2. В чем преимущества оформления любой последовательности операторов, которая встречается в программе более одного раза, как функции?
3. Перечислить компоненты функции.
4. С помощью чего активируются функции?
5. В каком случае наличие прототипа не обязательно?
6. Как называется переменная, которая известна только внутри функции, в которой она определена?
7. Какой оператор в вызываемой функции используется для того, чтобы передать значение выражения обратно в вызывающую функцию?
8. Как называется часть программы, в которой идентификатор может использоваться?
9. Назвать ключевое слово, используемое в заголовке функции, чтобы указать, что функция не возвращает значения или что она не имеет параметров.
10. Как называется переменная, объявленная вне любого блока или функции?
11. Как называется функция, которая прямо или косвенно вызывает сама себя?
12. Где указывается тип значения, которое возвращает функция?
13. Когда функцией возвращается значение, тип которого `int`?
14. В программе на C++ может быть несколько функций с одним именем, каждая из которых оперирует с различными типами и (или) количеством аргументов. Как это называется?
15. Какая функция позволяет определить единственную функцию для задач со многими различными типами данных?
16. Что позволяет компилятору проверить число, тип и порядок следования аргументов, передаваемых функции?
17. В каком случае прототип функции мог бы объявлять тип параметра `double&`?

18. Какой квалификатор используется для объявления переменных только для чтения?

19. Верны или нет следующие утверждения?

- A. Все аргументы функций передаются в C++ по значению.
- B. Аргументы функций могут быть различного скалярного типа.
- C. Среди аргументов функций могут быть аргументы типа void.
- D. Оператор return должен завершать тело любой функции.
- E. Аргументом функции может быть функция.
- F. Аргументом функции может быть динамическая структура.

20. Найти и исправить ошибки в следующих фрагментах программ:

```
a) int y(void)
{
    cout << "Inside y" << endl;
    int z(void)
    {
        cout << "Inside z" << endl;
    }
}

b) int sum(int x, int y)
{
    int result;
    result = x + y;
}

c) int sum(int n)
{
    if (n == 0)
        return 0;
    else n + sum(n - 1);
}

d) void func(double a);
{
    float a;
    cout << a << endl;
}

e) void progr(void)
{
    int a, b, c, result;
    cout << "Enter three integers:" << endl;
    cin >> a >> b >> c;
    result = a*b*c;
    cout << "Result is " << result;
    return result;
}
```


21. Написать заголовки для каждой из следующих функций

А. Функция `fa` принимает два аргумента с плавающей точкой двойной точности `side1` и `side2` и возвращает результат с плавающей точкой.

В. Функция `fb`, которая принимает три целых значения `x`, `y` и `z` и возвращает целое значение.

С. Функция `fc`, которая не получает ни одного аргумента и не возвращает значения.

Д. Функция `fd`, которая принимает целый аргумент `number` и возвращает результат с плавающей точкой одинарной точности.

22. Написать прототипы для каждой функции из упражнения 21.

23. По данному фрагменту программы восстановить условие задачи.

```
const double PI = 3.14159;
double V(const double r)
{
    return 4.0 / 3.0*PI*pow(r, 3);
}

int main()
{
    double radius;
    cout << "Enter the length of the radius of your sphere:";
    cin >> radius;
    cout << "Volume of sphere with radius " << radius << " is "
<< V(radius) << endl;
    return 0;
}
```

24. Ниже выписаны несколько заголовков функции `func`, которая возвращает сумму элементов одномерного массива вещественных чисел.

- 1) `float func(float a[], int n);`
- 2) `float func(double a[], int n);`
- 3) `float func(float *a, int n);`
- 4) `float func(float a[1000], int n);`
- 5) `float func(float *begin, float *end).`

Перечислить номера заголовков, которые выписаны верно.

25. По приведенной ниже программе восстановить условие.

```
#include <iostream>
using namespace std;
void mas(int, int[]);
void main()
{
    const int n = 10;
```

```

    int a[n];
    for (int i = 0; i < n; i++)
        cin >> a[i];
    mas(n, a);
    for (int i = 0; i < n; i++)
        cout << "\ta[" << i << "]= " << a[i];
}

void mas(int k, int x[])
{
    for (int j = 1; j < k - 1; j++)
        if (x[j] > 0 && x[j] % 10 != 5)
            x[j - 1] = x[j - 1] - x[j + 1] % 10;
}

```

26. Определена функция, которая распечатывает двумерный массив `a[13][5]` в виде матрицы.

```

void print1(float a[][5], int n)
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < 5; j++)
            cout << a[i][j] << " ";
        cout << endl;
    }
}

```

1) Заменить заголовок функции, используя указатель на одномерный массив.

2) Составить функцию `print3`, рассматривая матрицу *a* как одномерный массив.

3) Написать оператор вызова функции `print3` для печати матрицы `arr`.

```

float arr[][5] = {{11,12,13,14,15},
                  {21,22,23,24,25},
                  {31,32,33,34,35},
                  {41,42,43,44,45}
                  };

```

4) Выполнить предыдущее задание, используя явное приведение типа.

27. Определена функция `swap`, которая меняет местами значения двух переменных

```

void swap(int *a, int *b)
{
    int z;
    z = *a;
    *a = *b;
}

```

```

    *b = z;
}

```

- 1) Написать оператор вызова этой функции для переменных
int x1 = 20, x2 = 11;
- 2) Верно ли, что указатели передаются функции в качестве параметров и не меняются в результате выполнения функции, но меняются те значения, на которые направлены указатели.

28. Определить, что будет напечатано в результате выполнения программ.

- 1) `void show1(int x = 1);`
`void show2(int a, int b = 10, int c = 13);`

```

int main()
{
    show1(25);
    show1();
    show2(3, 4, 5);
    show2(17, 21);
    show2(99);
    return 0;
}

void show1(int x)
{
    cout << "x=" << x << endl;
}

void show2(int a, int b, int c)
{
    cout << "a=" << a << " ";
    cout << "b=" << b << " ";
    cout << "c=" << c << " ";
}

```

- 2) Можно ли вместо прототипа
`void show2(int a, int b = 10, int c = 13);`
использовать прототип
`void show2(int a, int = 10, int = 13);`

29. Назвать два подхода, которые используются при определении количества параметров при программировании функций с переменным числом параметров.

30. Ниже приведена программа, содержащая функцию sum с переменным числом параметров.

```

int sum(int, ...); // прототип функции

void main()
{
    cout << "\n 1+9+6+9=" << sum(4, 1, 9, 6, 9);
    cout << "\n 1+2+3+4+5+6+7=" << sum(7, 1, 2, 3, 4, 5, 6, 7);
    cout << "\n Параметры отсутствуют" << sum(0);
}

int sum(int n, ...)
{
    int *pt = &n;
    int s = 0;
    for (int i = 1; i <= n; i++)
        s += *(++pt);
    return s;
}

```

Каков результат выполнения программы?

31. Ниже приведена программа, которая содержит функцию с переменным числом параметров. Для определения конца списка параметров используется параметр-индикатор.

```

double pr(double x, ...)
{
    double b = 1.0;
    double *pt = &x;
    if (*pt == 0)
        return 0.0;
    for (; *pt; pt++)
        b *= *pt;
    return b;
}

void main()
{
    cout << "\n pr(1e0, 9e0, 4e0, 5e0, 0e0)=" << pr(1e0, 9e0,
4e0, 5e0, 0e0);
    cout << "\n pr(2.0, 2.5, 0.0, 9.0, 15.0, 0.0)=" << pr(2.0,
2.5, 0.0, 9.0, 15.0, 0.0);
    cout << "\n pr(0.0)=" << pr(0.0);
}

```

Каков результат выполнения программы?

32. Для чего предназначена следующая функция?

```
void was(int *a, int n)
{
    int imi, temp;
    for (int i = 0; i < n - 1; i++)
    {
        imi = i;
        for (int j = i + 1; j < n; j++)
            if (a[imi] > a[j])
                imi = j;
        if (imi > i)
        {
            temp = a[i];
            a[i] = a[imi];
            a[imi] = temp;
        }
    }
}
```

33. Назвать библиотечную функцию, которая позволяет завершить выполнение программы и возвращает операционной системе значение своего параметра.

34. Что называется перегрузкой функции?

35. Для чего используется перегрузка функций?

36. Какой процесс называется разрешением перегрузки (от английского слова resolution)?

37. Дать определение процесса декодирования имени.

38. Сформулировать основные правила описания перегруженных функций.

11.1.2. Ответы и решения

1. Функция.

2. Сокращается размер программы, так как код функции хранится в одной области памяти.

3. Прототип, определение, вызов.

4. С помощью вызова.

5. Когда определение функции располагается до основной программы (main()), в которой функция вызывается.

6. Локальная переменная.

7. `return`.
8. Область действия идентификатора.
9. `void`.
10. Глобальная переменная.
11. Рекурсивная функция.
12. Перед именем функции.
13. Когда перед именем функции указывается `int` или ничего не указывается?
14. Перегрузка функций.
15. Шаблон.
16. Прототип.
17. Если объявлен ссылочный параметр типа «ссылка на `double`», чтобы получить доступ по ссылке к исходной переменной вызывающей функции.
18. `const`.
19. Верны b), c), e), f).
20. a) функция `z` определяется в функции `y`.
 b) ошибка: функция `sum` должна возвращать целое, но не делает этого. Исправленное тело функции `sum`:

```
{
    return x+y;
}
```


 c) ошибка: результат выражения не возвращается. Исправление:

```
else
    return n+sum(n-1);
```


 d) две ошибки: точка с запятой в заголовке функции и переопределение параметра `a` в теле функции.
 e) функция возвращает значение `result`, хотя не должна этого делать. Исправление: удалить оператор `return`.
21. a) `double fa(double side1, double side2);`
 b) `int fb(int x, int y, int z);`
 c) `void fc()` или `void fc(void);`
 d) `float fd(int number);`
22. a) `double fa(double, double);`
 b) `int fb(int, int, int);`
 c) `void fc();` или `void fc(void);`
 d) `float fd(int);`

23. Программа предлагает пользователю ввести радиус шара, после чего вычисляет и печатает его объем, используя встроенную функцию V, которая возвращает результат выражения $4.0/3.0*3.14159*pow(r,3)$

24. 1, 2, 3, 4, 5.

25. Если очередной элемент массива положительный и не оканчивается цифрой 5, то предыдущий элемент уменьшить на последнюю цифру следующего элемента массива.

```
26. 1) void print2(float (*a)[5], int n)
    2) void print3(float a[], int m, int n)
        {
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                    cout << a[i*n+j] << " ";
                cout << endl;
            }
        }
    3) print3(*arr, 4, 5);
    4) print3((float*)arr, 4, 5).
```

27. 1) swap(&x1, &x2);

2) верно

28. 1) x=25

x=1

a=3 b=4 c=5

a=17 b=21 c=13

a=99 b=10 c=13

2) Можно, функциональность программы от этого не изменится.

29. 1) один из параметров определяет число дополнительных параметров функции;

2) в списке явных параметров последним задается параметр-индикатор, который указывает на окончание списка параметров.

30. 1+9+6+9=25

1+2+3+4+5+6+7=28

Параметры отсутствуют 0

31. pr(1e0, 9e0, 4e0, 5e0, 0e0)=180

pr(2.0, 2.5, 0.0, 9.0, 15.0, 0.0)=5

pr(0.0)=0

32. Приведенная функция предназначена для сортировки массива по возрастанию.

33. `exit()`

34. Использование нескольких функций с одним и тем же именем, но с различными списками параметров.

35. Для создания нескольких функций, реализующих сходные алгоритмы, но с разными списками данных.

36. Процесс, когда компилятор определяет, какую именно функцию необходимо выбрать по типу фактических параметров, их количеству.

37. Компилятор кодирует идентификатор каждой функции в зависимости от числа и типов параметров, чтобы обеспечить безопасную по типу компоновку, которая гарантирует, что будет вызвана нужная функция и что ее аргументы будут согласованы по типу с параметрами.

38. Перегруженные функции должны находиться в одной области видимости, иначе произойдет сокрытие аналогичных переменных во вложенных блоках.

Перегруженные функции могут иметь параметры по умолчанию, но значения одного и того же параметра в разных функциях должны совпадать.

Перегруженные функции могут иметь разные по типу и количеству параметры.

В различных вариантах перегруженных функций может быть разное количество параметров по умолчанию.

Функции не могут быть перегружены, если описание их параметров отличается только модификатором `const` или использованием ссылки.

Перегруженные функции должны отличаться друг от друга либо типом хотя бы одного параметра, либо количеством параметров, либо и тем и другим.

11.2. Примеры разработанных программ

11.2.1. Условия заданий

1. Что будет напечатано в результате работы следующей программы.

```
using namespace std;
void repchar(char, int);
int main()
{
    repchar('-', 10);
    cout << "Типы данных          Диапазон" << endl;
    repchar('=', 25);
}
```



```

        cout << "char    -128 to 127" << endl << "short    -32768
to 32767" << endl << "int системно зависимый -2.147.483.648 to
2.147.483.647" << endl;
        repchar('*', 25);
        return 0;
}
void repchar(char ch, int n)
{
    for (int i = 0; i < n; i++)
        cout << ch;
    cout << endl;
}

```

2. Внести изменения в основную программу предыдущей задачи: не изменяя функцию `repchar`, основная программа должна передавать в функцию `repchar` значения переменных вместо констант.

3. Составить функцию упорядочения по возрастанию в тех же переменных пары целых чисел x , y .

4. Составить функцию, которая вычисляет площадь треугольника, если три заданных числа x , y , z могут быть длинами его сторон.

5. Составить функцию для определения косинуса угла между векторами x и y , в каждом из которых n координат по формуле

$$\cos \alpha = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2 * \sum_{i=1}^n y_i^2}}.$$

6. Найти результат выполнения следующего фрагмента программы.

```

void max_vect(int n, int *x, int *y, int *z)
{
    for (int i = 0; i < n; i++)
        z[i] = x[i]>y[i] ? x[i] : y[i];
}

void main()
{
    int a[] = { 1,12,3,4,16,6,7,0 };
    int b[] = { 0,7,6,15,14,3,12,1 };
    int c[8];
    max_vect(8, a, b, c);
    for (int i = 0; i < 8; i++)
        cout << "\t" << c[i];
}

```

7. Восстановить условие задачи, программа решения которой приведена ниже.

```
#include <iostream>
using namespace std;
int **single_matr(int n);
void main()
{
    int n;
    cout << "Ввести порядок матрицы:" << endl;
    cin >> n;
    int **matr;
    matr = single_matr(n);
    for (int i = 0; i < n; i++)
    {
        cout << "\n строка " << i + 1 << ":";
        for (int j = 0; j < n; j++)
            cout << "\t" << matr[i][j];
    }
    for (int i = 0; i < n; i++)
        delete[] matr[i];
    delete[] matr;
}

int **single_matr(int n)
{
    int **p = new int *[n];
    if (p == NULL)
    {
        cout << "Не создан динамический массив!";
        exit(1);
    }
    for (int i = 0; i < n; i++)
    {
        p[i] = new int[n];
        if (p[i] == NULL)
        {
            cout << "Не создан динамический массив!";
            exit(1);
        }
        for (int j = 0; j < n; j++)
        {
            if (j != i)
                p[i][j] = 0;
            else
```

```

        p[i][j] = 1;
    }
}
return p;
}

```

8. Дана программа

```

#include <iostream>
using namespace std;
const int n = 5;
int s(...); // прототип функции, аргументы которой
            // предлагается вписать студентам

int main()
{
    int mas[] = { 2018, 1917, 2045, 90, 1969 };
    cout << s(...);
    return 0;
}

int s(...)
{
    int summa = 0;
    for (int i = 0; i < n; i++)
    {
        summa += mas[i] % 10;
    }
    return summa;
}

```

8.1. Выбрать из предложенных ниже возможный вариант заголовка функции s.

```

int s(const int *mas, const int n);
int s(int mas[], int n);
int s(int mas[n], int n);

```

8.2. Что будет напечатано в случае выбора правильного заголовка функции s?

9. Каков результат работы приведенной ниже программы?

```

#include <iostream>
using namespace std;
void order(int*, int*);
int main()
{
    int a = 2018, b = 1945, c = 50, d = 100;
    order(&a, &b);
}

```

```

    order(&c, &d);
    cout << "a=" << a << endl;
    cout << "b=" << b << endl;
    cout << "c=" << c << endl;
    cout << "d=" << d << endl;
    return 0;
}

void order(int *n1, int *n2)
{
    if (*n1 > *n2)
    {
        int x = *n1;
        *n1 = *n2;
        *n2 = x;
    }
}

```

10. В приведенной ниже программе используется функция `order` из предыдущей задачи. Сформулировать условие программы, текст которой приведен ниже. Что будет напечатано в результате выполнения программы?

```

void was(int *, int);
int main()
{
    const int N = 10;
    int arr[N] = { 20,18,119,64,19,43,1961,5,4,3 };
    was(arr, N);
    for (int i = 0; i < N; i++)
        cout << arr[i] << " ";
    cout << endl;
    return 0;
}

void order(int *n1, int *n2)
{
    if (*n1 > *n2)
    {
        int x = *n1;
        *n1 = *n2;
        *n2 = x;
    }
}

```

```

void was(int *p, int n)
{
    for (int i = 0; i < n - 1; i++)
        for (int k = i + 1; k < n; k++)
            order(p + i, p + k);
}

```

11. Для чего предназначена следующая функция, текст которой приведен ниже?

```

void was(int *arr, int n)
{
    for (int i = 0; i < n; i++)
    {
        int k = i;
        for (int j = i + 1; j < n; j++)
            if (arr[k] > arr[j])
                k = j;
        int x = arr[i];
        arr[i] = arr[k];
        arr[k] = x;
    }
}

```

12. Пусть описана функция `sum`, для вычисления суммы элементов одномерного массива, указатель на первый элемент которого `*begin`, на элемент, следующий за последним, `*end`.

```

float sum(float *begin, float *end)
{
    float s = 0;
    while (begin != end)
        s += *(begin++);
    return s;
}

```

В основной программе описан и проинициализирован массив
`float arr[10] = { 20,1,18,5,80,1943 };`

Используя функцию `sum`, напечатать сумму элементов массива `arr`.

13. Описать функцию печати двумерного массива как одномерного. Параметры этой функции: имя массива `a`, количество строк `str`, столбцов `stb`.

14. Описать функцию, которая возвращает ссылку на больший из двух аргументов.

15. Каков результат выполнения программы, содержащей функции, аргументы которых имеют значения по умолчанию

```

void print1(int a = 0)
{
    cout << "a=" << a << endl;
}

void print2(int a, int b = 1, int c = 2);

int main()
{
    print1(2018);
    print1();
    print2(2, 0, 19);
    print2(100, 50);
    print2(1917);
    return 0;
}

void print2(int a, int b, int c)
{
    cout << "a=" << a << " ";
    cout << "b=" << b << " ";
    cout << "c=" << c << endl;
}

```

16. Составить программу, которая определяет для трех различных положительных целых многозначных чисел максимальную сумму цифр. Использовать функции для вычисления суммы цифр целого положительного многозначного числа и для определения максимального из трех чисел.

17. Составить программу уменьшения каждого элемента массива на его индекс с использованием соответствующей функции.

18. Каков результат выполнения программы?

```

#include <iostream>
using namespace std;
int sum(int, ...); // прототип функции с переменным числом
                  // параметров

void main()
{
    cout << sum(2, 10, 5) << endl;
    cout << sum(5, 10, 20, 1, 2, 3) << endl;
    cout << sum(0);
}

```

```

int sum(int n, ...) // n - число суммируемых параметров
{
    int *pt = &n;
    int s = 0;
    for (int i = 1; i <= n; i++)
    {
        if (n == 0)
            break;
        s += *(++pt);
    }
    return s;
}

```

19. Составить функцию поиска элемента в неупорядоченном массиве целых чисел.

20. Составить функцию для вычисления длины *s*-строки (не используя встроенную функцию `strlen`).

21. Составить функцию для раскодировки и печати текстового сообщения, закодированного как целочисленный массив. В числе аргументов функции присутствует текстовая строка (символьный массив), с помощью которой происходит раскодировка.

22. Составить программу для нахождения с заданной точностью методом половинного деления корня уравнения $f(x) = 0$ на заданном интервале (a, b) при условии, если на этом интервале есть корень, то он единственный.

Программа должна содержать функции вычисления значения $f(x) = x^2 - 9x + 14$ и функцию поиска корня.

23. Составить функции ввода массива, отображения его элементов, вычисления дисперсии. Написать программу, вызывающую все эти функции.

24. Составить функцию `gasst` для вычисления расстояния между двумя точками пространства. Точки представляются в виде экземпляров структур, которые имеют по три поля типа `double` – координаты точки.

25. Даны три точки в пространстве. Если они могут быть вершинами треугольника, вычислить его площадь. Использовать функцию `gasst` (из задания 24), функцию вычисления площади треугольника с проверкой условия его существования.

26. Составить функцию, которая определяет номер строки матрицы порядка $m \times n$, в которой находится самая длинная подпоследовательность равных элементов.

Написать программу, вызывающую эту функция для конкретной матрицы.

27. Составить рекурсивную функцию упорядочивания одномерного массива методом быстрой сортировки.

28. Составить рекурсивную функцию определения с заданной точностью eps корень уравнения $f(x)=0$ на отрезке $[a,b]$, считая, что исходные данные задаются без ошибок, т. е. $eps > 0$, $b > a$, $f(a) * f(b) < 0$ и на отрезке $[a,b]$ есть только один корень.

Применить составленную рекурсивную функцию для нахождения корня уравнения $\frac{2}{x} * \cos\left(\frac{x}{2}\right) = 0$ на отрезке $[0.1, 3.5]$ с точностью 0.00005. Определить количество обращений к тестовой функции.

29. При некоторых заданных x , N и E , определяемых вводом, вычислите

а) сумму N слагаемых заданного вида;

б) сумму тех слагаемых, которые по абсолютной величине больше E .

Вычисление второй суммы выполните для двух значений E , отличающихся на порядок, при этом определите количество слагаемых, включенных в сумму, вычисляемую для каждого значения E . Сравните результаты со значением функции, для которой данная сумма определяет приближенное значение при x , лежащем в интервале $(-R, R)$, вычисленным с помощью встроенных функций компилятора.

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (R = \infty).$$

Для решения сформулированной задачи необходимы две функции.

Первая функция (вариант а), сигнатура которой имеет вид

```
float rsin (float x, int N);
```

вычисляет значение ряда $\sin(x)$ для заданного вещественного значения аргумента x с помощью N слагаемых ряда Тейлора. Эта функция имеет два параметра и возвращает вещественное значение вычисленной суммы.

Вторая (перегруженная) функция с тем же именем

```
float rsin (float x, float e, int &count);
```

вычисляет по тому же алгоритму Тейлора значение суммы с заданной точностью eps . Она (вариант б) возвращает в основную программу не только вычисленное значение суммы, но и количество слагаемых в этой сумме (`int &count`). Количество параметров для этого варианта функции `rsin` равно 3.

30. Использовать перегрузку функций для нахождения корня уравнения $f(x)=0$ разными методами: методом деления отрезка пополам (программа рассмотрена в заданиях 22, 28), методом хорд и методом Ньютона.

Метод хорд является разновидностью метода деления отрезка пополам. Контрольная точка c выбирается как точка пересечения хорды, проводимой через концы отрезка $[a,b]$ с координатной осью:

$$c = \frac{f(b)a - f(a)b}{f(b) - f(a)}.$$

Для вычисления корня уравнения $f(x)=0$ по методу Ньютона необходимо знать начальное приближение x_0 к корню. Далее этот корень уточняется

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

где производная вычисляется по формуле

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}.$$

Шаг изменения аргумента h является малой величиной.

Использовать разработанные перегруженные функции для решения тестового уравнения $f(x) = x^2 - 9x + 14$.

Предложите другие тестовые примеры.

11.2.2. Ответы и решения

1. Результат работы предложенной программы.

```
-----
Типы данных    Диапазон
=====
char            -128 to 127
short           -32768 to 32767
int             системно зависимый -2.147.483.648 to 2.147.483.647
*****
```

2.

```
int main()
{
    char ch;
    int col;
    cout << "Введите символ: " << endl;
    cin >> ch;
    cout << "Введите число повторений символа: " << endl;
    cin >> col;
```

```

    cout << endl;
    repchar(ch, col);
    ...
    return 0;
}

```

3.

```

void order(int &x, int &y)
{
    if (x > y)
    {
        int tmp = x;
        x = y;
        y = tmp;
    }
}

```

4.

```

double square(double x, double y, double z)
{
    double s = 0;
    if (x > 0 && y > 0 && z > 0 && (x + y > z) && (x + z > y)
&& (y + z > x))
    {
        double p = (x + y + z) / 2;
        s = sqrt(p*(p - x)*(p - y)*(p - z));
    }
    return s;
}

```

5.

```

float cosinus(int n, float x[], float y[])
{
    float a = 0, b = 0, c = 0;
    for (int i = 0; i < n; i++)
    {
        a += x[i] * y[i];
        b += x[i] * x[i];
        c += y[i] * y[i];
    }
    return float(a / sqrt(double(b*c)));
}

```

6.1 12 6 15 16 6 12 1

7. В программе описана функция формирования единичной диагональной матрицы, порядка n с использованием динамических массивов.

8.

8.1. Любой из предложенных заголовков является правильным.

8.2. 29

9. a=1945

b=2018

c=50

d=100

10. 3 4 5 18 19 20 43 64 119 1961

11. Функция сортировки элементов одномерного массива по возрастанию. Параметры функции: указатель на имя массива и количество элементов в массиве.

12. `cout << sum(arr, arr + 10);`

13.

```
void print(float a[], int str, int stb)
{
    for (int i = 0; i < str; i++)
    {
        for (int j = 0; j < stb; j++)
            cout << a[i*stb + j] << " ";
        cout << endl;
    }
}
```

14.

```
int &smax(int &x, int &y)
{
    if (x >= y)
        return x;
    else
        return y;
}
```

15.

```
a=2018
a=0
a=2 b=0 c=19
a=100 b=50 c=2
a=1917 b=1 c=2
```

16.

```
#include <iostream>
using namespace std;
```

```

int sum(int x)
{
    int s = 0;
    do
    {
        s += x % 10;
        x /= 10;
    } while (x != 0);
    return s;
}

int max3(int x, int y, int z)
{
    int m = x > y ? x : y;
    if (z > m) m = z;
    return m;
}

int main()
{
    int a, b, c;
    cout << "Ввести три целых положительных числа:" << endl;
    cin >> a >> b >> c;
    cout << "Максимальная сумма цифр=" << max3(sum(a), sum(b),
sum(c));
    return 0;
}

```

17.

```

#include <iostream>
using namespace std;
void minusind(int, float[]); // прототип функции
void main()
{
    const int n = 10;
    float arr[n];
    for (int i = 0; i < n; i++)
    {
        cout << "\nВведите элемент массива: ";
        cin >> arr[i];
    }
    minusind(n, arr);
    for (int i = 0; i < n; i++)
        cout << "\tarr[" << i << "]= " << arr[i] << endl;
}

```

```

void minusind(int k, float a[])
{
    for (int i = 0; i < k; i++)
        a[i] -= i;
}

```

18.

```

15
36
0

```

19.

```

int poisk(int a[], int n, int x)
{
    bool ok = false;
    int i = 0;
    while (i < n && !ok)
        if (x == a[i])
            ok = true;
        else
            i++;
    if (ok)
        return i;
    else return -1;
}

```

20.

```

int dlina (char *str)
{
    int k = 0;
    while (str[k])
        k++;
    return k;
}

```

21.

```

void decod(char *s, int *str, int n)
{
    char *result=new char[n+1];
    for (int i = 0; i < n; i++)
        result[i] = s[str[i]];
    result[n] = '\0';
    cout << result << endl;
    delete[]result;
}

```

22.

```
// функция для вычисления полинома
double F(double x)
{
    return x*x - 9 * x + 14;
}
// функция нахождения корня
double Root(double(*f)(double), double a, double b, double eps)
{
    // a, b концы интервала, eps точность
    double c; // середина интервала (a,b)
    while ((b - a) / 2 > eps)
    {
        c = (a + b) / 2;
        if (f(a)*f(c) > 0)
            a = c;
        else
            b = c;
    }
    return c;
}

int main()
{
    double a, b, eps, x;
    cout << "Ввести концы интервала:" << endl;
    cin >> a >> b;
    // проверка корректности интервала
    if (F(a)*F(b)>0)
    {
        cout << "На заданном интервале нет корня!\n";
        return -1;
    }
    if (a > b)
    {
        x = a;
        a = b;
        b = x;
    }
    cout << "Ввести точность\n";
    cin >> eps;
    // поиск корня
    x = Root(F, a, b, eps);
    cout << "x=" << x << endl;
```

```

        return 0;
    }

    23.
#include <iostream>
#include <Windows.h>
using namespace std;
// среднее арифметическое
double sr(double *x, int n)
{
    double s=0;
    for (int i = 0; i < n; i++)
        s += x[i];
    s /= n;
    return s;
}

// дисперсия
double disp(double *x, int n)
{
    double y=sr(x,n), s = 0;
    for (int i = 0; i < n; i++)
        s += x[i] * x[i];
    s /= n;
    return s - y*y;
}

// заполнение массива
void sozdrand(double *x, int n)
{
    srand(GetTickCount());
    for (int i = 0; i < n; i++)
        *(x + i) = rand() % 10;
}

// ввод элементов массива
void input(double *x, int n)
{
    for (int i = 0; i < n; i++)
        cin >> *(x + i);
}

// печать массива

```

```

void print(double *x, int n)
{
    for (int i = 0; i < n; i++)
        cout << *(x + i) << " ";
}
// динамическое выделение памяти
bool anew(double *x, int n)
{
    x = new double[n];
    if (x == NULL)
    {
        cout << "Не создан динамический массив";
        return false;
    }
    else
        return true;
}
int main()
{
    const int N = 20;
    double* arr=NULL; // массив значений
    // выделение динамической памяти для массива arr[N]
    bool t = anew(arr, N);
    if (!t)
    {
        cout << "Выход из основной программы";
        return -1;
    }
    else
    {
        // ввод элементов массива arr
        cout << "Ввести элементы массива arr\n";
        input(arr, N);
        cout << "Исходный массив\n";
        print(arr, N);
        // дисперсия массива arr
        double d = disp(arr, N);
        cout << "Результат работы программы\n";
        cout << "Дисперсия введенного массива=" << d << endl;
        // освобождение памяти
        delete[] arr;
    }
    return 0;
}

```



```

24.
struct Point
{
    double x;
    double y;
    double z;
};

double rasst(Point A, Point B)
{
    return sqrt(pow(A.x - B.x, 2) + pow(A.y - B.y, 2) +
        pow(A.z - B.z, 2));
}

25.
struct Point
{
    double x;
    double y;
    double z;
};

double rasst(Point A, Point B)
{
    return sqrt(pow(A.x - B.x, 2) + pow(A.y - B.y, 2) + pow(A.z
- B.z, 2));
}

double S(Point A, Point B, Point C)
{
    double d1 = rasst(A, B);
    double d2 = rasst(A, C);
    double d3 = rasst(B, C);
    if (d1 > 0 && d2 > 0 && d3 > 0 && d1 + d2 > d3 &&
        d1 + d3 > d2 && d2 + d3 > d1)
    {
        double p = (d1 + d2 + d3) / 2;
        double p1 = sqrt(p(p - d1)*(p - d2)*(p - d3));
        return p1;
    }
    else
    {
        return - 1;
    }
}

```

```

int main()
{
    Point A, B, C;
    cout << "Ввести координаты трех точек\n";
    cin >> A.x >> A.y >> A.z;
    cin >> B.x >> B.y >> B.z;
    cin >> C.x >> C.y >> C.z;
    double f = S(A, B, C);
    if (f < 0)
    {
        cout << "Исходные точки не являются вершинами
треугольника\n";
        return -1;
    }
    else
    {
        cout << "Площадь треугольника=" << f << endl;
        return 0;
    }
}

```

26.

```

// функция, возвращая номер строки, в которой находится
// самая длинная подпоследовательность равных элементов
int equal(int **a, const int m, const int n)
{
    // m строк, n элементов в каждой строке
    int k, line = -1, max = 0;
    for (int i = 0; i < m; i++)
    {
        k = 0;
        for (int j = 0; j < n - 1; j++)
        {
            if (*(a + i) + j == *(a + i) + j + 1))
                k++;
            else
            {
                if (k > max)
                {
                    max = k;
                    line = i;
                }
                k = 0;
            }
        }
    }
}

```

```

        if (k > max)
        {
            max = k;
            line = i;
        }
    }
    return line;
}

int main()
{
    int m, n;
    cout << "Ввести количество строк" << endl;
    cin >> m;
    cout << "Ввести количество элементов в строке" << endl;
    cin >> n;
    // динамическое выделение памяти
    int **x = new int *[m];
    if (x == NULL)
    {
        cout << "Не создан динамический массив\n";
        return -1;
    }
    for (int i = 0; i < m; i++)
    {
        *(x + i) = new int[n];
        if (*(x + i) == NULL)
        {
            cout << "Не создан динамический массив\n";
            return -1;
        }
    }
    cout << "Ввести элементы матрицы\n";
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++)
            cin >> (*(x + i) + j);
    cout << "Исходная матрица\n";
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
            cout << (*(x + i) + j) << " ";
        cout << endl;
    }
}

```

```

    int l = equal(x, m, n);
    if (l >= 0)
        cout << "самая длинная подпоследовательность равных
элементов находится в строке" << l;
    else
        cout << "подпоследовательностей одинаковых элементов в
строках матрицы нет";
    return 0;
}

```

27.

```

void sort(float *a, int left, int right)
{
    int i = left, j = right;
    float s = a[(left + right) / 2];
    float x;
    while (i < j)
    {
        while (a[i] < s) i++;
        while (s < a[j]) j--;
        if (i <= j)
        {
            x = a[i];
            a[i] = a[j];
            a[j] = x;
            i++;
            j--;
        }
    }
    if (left < j) sort(a, left, j);
    if (i < right) sort(a, i, right);
}

```

28.

```

double RootRec(double f(double), double a, double b, double eps)
{
    double fa = f(a), fb = f(b), c, fc;
    if (fa*fb > 0)
    {
        cout << "\nНеверный интервал локализации корня\n";
        exit(-1);
    }
    c = (a + b) / 2;
    fc = f(c);
}

```

```

        if (fc == 0 || b - a < eps)
            return c;
        return (fa*fc < 0.0) ? RootRec(f, a, c, eps) : RootRec(f,
c, b, eps);
}

```

```

static int col = 0; // счетчик вызовов тестовой функции

```

```

double testf(double); // прототип тестовой функции

```

```

int main()
{
    SetConsoleOutputCP(1251);
    double root, A = 0.1, B = 32, EPS = 5e-5;
    root = RootRec(testf, A, B, EPS);
    cout << "\nЧисло обращений к тестовой функции = " << col;
    cout << "\nКорень = " << root << endl;
    return 0;
}

```

```

// определение тестовой функции
double testf(double x)
{
    extern int col;
    col++;
    return (2.0 / x*cos(x / 20));
}

```

29. В основной программе происходит ввод значений параметров для каждой перегруженной функции, вызов этих функций и вывод полученных результатов.

// функция для вычисления $\sin(x)$ с помощью N слагаемых ряда Тейлора

```

float rsin (float x, int N)
// функция имеет два параметра: x, N
{
    float s = x, a = -x*x, y = 1;
    // y - очередное слагаемое суммы s
    for (int i = 1; i < N; i++)
    {
        y *= a/(2*i*(2*i+1));
        s += y;
    }
    return s;
}

```

```

// функция для вычисления sin(x) с помощью ряда Тейлора
// с точностью e и определения количества слагаемых суммы
float rsin (float x, float e, int &count)
// функция имеет три параметра: x, e, count
{
    float s = x, a = -x*x, y = 1;
    int i = 1;
    while (abs(y)>e)
    {
        y *= a/(2*i*(2*i+1));
        s += y;
        i++;
    }
    count = i;
    return s;
}
void main ()
{
    cout << "Ввести x" << endl;
    cin >> x;
    cout << "Ввести количество слагаемых" << endl;
    cin >> N;
    if (N > 0)
    {
        cout << "x = " << x << "N = " << N <<
            "сумма ряда = " << rsin(x,N);
        cout << "точное значение суммы =" << sin(x) << endl;
        cout << "Ввести точность" << endl;
        cin >> e;
        int count;
        cout << "x=" << x << "e=" << e <<
            "сумма ряда = " << rsin(x,e,count) <<
            "количество слагаемых = " << count << endl;
        cout << "точное значение суммы =" << sin(x) << endl;
        e*=0.1;
        cout << "x=" << x << "e=" << e <<
            "сумма ряда = " << rsin(x,e,count) <<
            "количество слагаемых = " << count << endl;
        cout << "точное значение суммы =" << sin(x) << endl;
    }
    else
        cout << "неверный ввод количества слагаемых" << endl;
}

```

30.

```
// функция для теста
double f(double x)
{
    return x*x - 9 * x + 14;
}

double RootFind(double(*f)(double), double a, double b, double
eps)
{
    double c, fa = f(a), fb=f(b);
    while (abs(fb - fa) > eps)
    {
        c = (fa*a - fb*b) / (fb - fa);
        if (fa*fb > 0)
        {
            a = c;
            fa = f(a);
        }
        else
        {
            b = c;
            fb = f(b);
        }
    }
    return c;
}

const int N = 20;

// Функция поиска корня методом Ньютона (два аргумента)
double RootFind(double(*f)(double), double x0)
// x0 - начальное приближение
{
    double x = x0, df, h = 0.000001;
    df = (f(x + h) - f(x)) / h;
    for (int i = 1; i < N; i++)
        x = x - f(x) / df;
    return x;
}

// другой вариант функции поиска корня методом хорд
// функция имеет три аргумента
```

```

double RootFind(double(*f)(double), double a, double b)
{
    double c, fa, fb;
    for (int i = 1; i < N; i++)
    {
        fa = f(a);
        fb = f(b);
        c = (fb*a - fa*b) / (fb - fa);
        if (fa*f(c) > 0)
            a = c;
        else
            b = c;
    }
    return c;
}

int main()
{
    double x0, a, b, xh, xNt, x;
    cout << "x0=";
    cin >> x0;
    xh = RootFind(f, x0);
    cout << xh << endl;
    cout << "a=";
    cin >> a;
    cout << "b=";
    cin >> b;
    if (b < a)
    {
        x = a;
        a = b;
        b = x;
    }
    if (f(a)*f(b) < 0)
    {
        xNt = RootFind(f, a, b);
        cout << xNt << endl;
    }
    else
    {
        cout << "Неверный интервал!\n";
        return 222;
    }
}

```


11.3. Задания для самостоятельной работы

1. В трех автопарках работают по 20 машин с заданными госномерами и указанными марками: ГАЗ, ЗИЛ, ЛИАЗ и т. д. Каждый водитель сдает сводку о том, сколько километров прошла его машина в каждый день недели, и о размере выручки. Подсчитайте общий километраж в каждом автопарке по каждому дню отдельно. В каком из автопарков самый большой автопробег? Какой день недели соответствует самой большой прибыли, полученной автохозяйством? Автомобили какой марки приносят наибольшую прибыль?

2. На заправочной станции для каждого автомата известно количество отпущенного топлива каждого вида (А76, А93, А95, ДИЗ ...) и количество обслуженных машин в каждый из дней недели. Вычислите общий объем топлива, проданный в указанный период недели. В какой день недели было обслужено больше всего машин? Из какого автомата было продано наибольшее количество топлива? Какой вид топлива пользуется наибольшим спросом?

3. Дана последовательность вещественных чисел a_1, a_2, \dots, a_n . Для каждого элемента последовательности a_i подсчитайте количество элементов слева, которые по модулю меньше a_i , и количество элементов справа, которые по модулю больше a_i . Используйте функцию, производящую вычисления для одного элемента последовательности.

4. Заданный массив целых чисел делится на три части двумя элементами: максимальным и минимальным. Определите сумму элементов в каждой части массива. Используйте функции для нахождения индексов минимального и максимального элементов и подсчета суммы элементов в указанной части массива.

5. Пусть дана матрица A размером $n \times m$, состоящая из вещественных чисел. Необходимо получить матрицу B размером $n \times m$, каждый элемент b_{ij} которой равен true, если сумма соседних с a_{ij} элементов меньше указанной величины S , false – в противном случае. Воспользуйтесь функцией для нахождения суммы соседних элементов.

6. Пусть даны две матрицы $A(n \times m)$, $B(n \times m)$, состоящие из вещественных чисел. Необходимо получить матрицу C той же размерности, где элемент C_{ij} равен сумме элементов i -й строки матрицы A , которые отсутствуют j -м столбце матрицы B . Напишите функцию вычисления очередного элемента матрицы C , использующую функцию проверки наличия числа в j -м столбце матрицы B .

7. Пусть на плоскости задано N параллелограммов координатами своих вершин. Определите номер параллелограмма, у которого площадь максимальна. Напишите функцию для определения площади параллелограмма по его заданным координатам вершин.

8. Пусть даны две квадратные вещественные матрицы порядка n . Получите новую матрицу следующим способом (для нахождения минимального элемента и произведения элементов в указанной строке используйте функции):

а) умножением минимального элемента каждой строки первой матрицы на наибольший элемент соответствующего столбца второй матрицы;

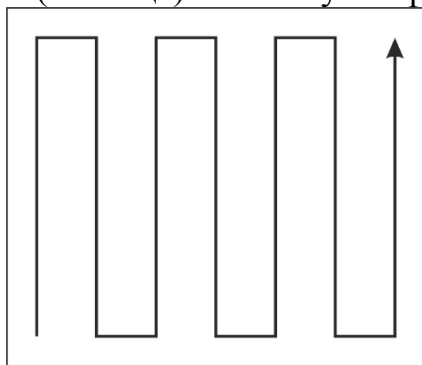
б) прибавлением к элементам каждого столбца первой матрицы произведения элементов соответствующих строк второй матрицы.

9. Допустимым преобразованием матрицы назовем перестановку, двух соседних строк или двух соседних столбцов. Пусть дана вещественная квадратная матрица порядка N . С помощью допустимых преобразований получите матрицу, в которой один из элементов с наибольшим значением располагался бы в левом верхнем углу. Напишите функции, осуществляющие соответствующие преобразования строк и столбцов.

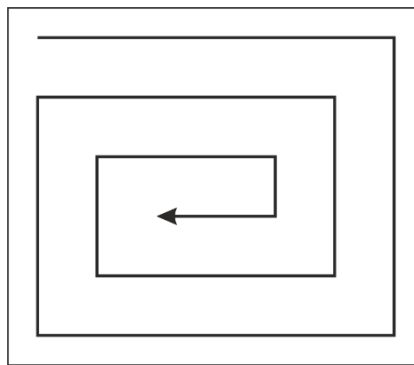
10. Пусть дана вещественная матрица A из n строк, в каждой из которых m элементов. В каждой строке переместите в конец строки все элементы, у которых соседние элементы в строке разные по знаку. Используйте функцию для перемещения указанного элемента строки.

11. Пусть дана матрица A из n строк, в каждой из которых m элементов, состоящая из натуральных чисел. В каждом столбце обнулите минимальное количество элементов так, чтобы сумма элементов столбца не превышала заданную.

12. Для заданной вещественной матрицы определите, образуют ли ее элементы упорядоченную последовательность при их переборе по схеме, представленной на рисунке. Для определения факта упорядоченности части строки (столбца) используйте функции.



a)



b)

Рис. 11.1. Схема расположения последовательности

13. Пусть дана вещественная квадратная матрица порядка $2n$. Получите новую матрицу, переставляя ее блоки размером n так, как показано на рис. Для обмена четырех заданных фрагментов матрицы напишите функцию.

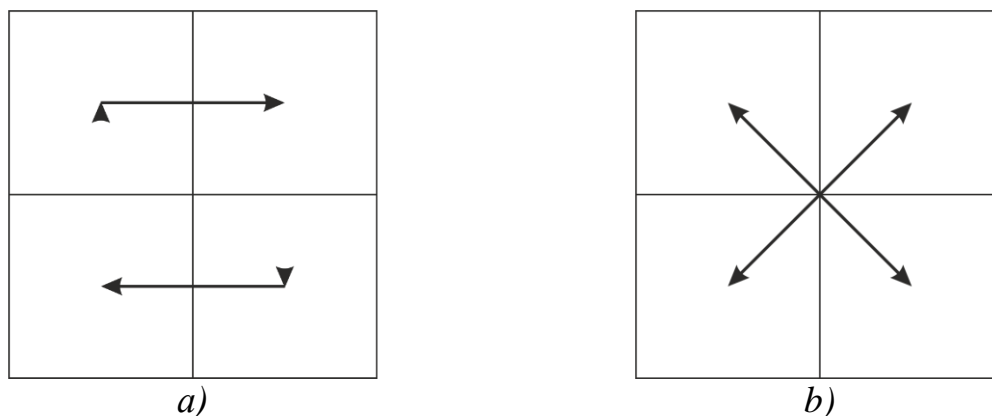


Рис. 11.2. Переставляемые блоки матрицы

14. Пусть дана прямоугольная матрица A из n строк, в каждой из которых m целых чисел. Определите, в какой строке матрицы находится наибольшее количество симметричных чисел. Составьте функцию, проверяющую симметричность числа.

15. Пусть дана прямоугольная матрица A из n строк, в каждой из которых m вещественных чисел. Определите для каждой строки, все ли ее элементы являются симметричными. Составьте функцию, проверяющую симметричность числа.

16. Пусть дана прямоугольная матрица A из n строк, в каждой из которых m целых чисел. Определите для каждого элемента матрицы позиции элементов матрицы с такой же суммой цифр, какую имеет максимальный элемент матрицы.

17. Пусть дана прямоугольная матрица A из n строк, в каждой из которых m целых чисел. Замените все положительные четные числа заданной матрицы на числа, являющиеся их «перевертышами». Составьте функцию, получающую для заданного числа его «перевертыш» (число a будем считать «перевертышем» числа b , если, читая число a справа налево, получаем число b).

18. Задан текст, состоящий из слов, разделенных пробелами. Удалите все слова, которые начинаются с нечетной позиции. Для удаления слова используйте функцию.

19. Задан текст, состоящий из слов, разделенных пробелами. По заданной строке текста напечатайте строку, в которой все гласные буквы удваиваются, а двойные согласные заменяются одиночными.

20. Напишите функцию, которая в заданной строке определяет количество вхождений в нее заданной подстроки. Для заданных строки и слова определите количество вхождений слова в строку.

21. Пусть даны N шестизначных номеров троллейбусных билетов – XXXXXX. Определите те билеты, номера которых являются «счастливыми», то есть сумма первых трех цифр совпадает с суммой последних трех цифр. Используйте функцию для определения, является ли номер «счастливым».

22. В квадратной матрице определите количество строк, отвечающих следующему условию (для определения типа каждой строки используйте функцию):

- a) упорядоченных по возрастанию;
- b) упорядоченных по убыванию;
- c) состоящих из равных элементов;
- d) неупорядоченных.

23. Пусть дан список учащихся и для каждого из них N оценок по M предметам. Переместите в конец списка тех учащихся, у которых средний балл меньше, чем средний балл в классе. Напишите функции для перемещения одной строки в конец списка и для вычисления среднего балла.

24. Пусть даны два предложения, которые оканчиваются точкой, содержат одинаковое количество слов, разделенных любым количеством пробелов. Определите для первых слов каждого предложения множество одинаковых букв, а для последних – те буквы, которые встречаются только в одном из них. Для определения буквенного состава слова используйте функцию.

25. Пусть задан текст, состоящий из нескольких предложений, разделенных точкой. Выполните следующие преобразования: первые слова в каждом предложении преобразуйте к виду, в котором первая буква прописная, а остальные строчные; в словах, которые содержат более чем одну прописную букву, все буквы замените прописными; в остальных словах все буквы должны быть строчные. Используйте функцию для преобразования одного слова в предложении.

26. Пусть дан текст в виде символьной строки, оканчивающейся точкой. Необходимо определить количество слов, которые являются «перевертышами» (например: казак, шалаш), то есть читаются одинаково с начала и

с конца слова. Слова в предложении отделяются одним либо несколькими пробелами. В начале предложения может быть любое количество пробелов.

27. Даны две таблицы A и B одинаковой размерности $m \times n$. Каждая строка таблицы A содержит количество различных изделий, выпущенных определенным цехом завода. Каждый цех выпускает n различных изделий. Структура таблицы B идентична таблице A , но содержит количество изделий, которое надо выпустить по плану. Номер строки соответствует номеру цеха завода.

Составить одномерный массив из номеров цехов, которые перевыполнили план по общему количеству выпущенных изделий.

28. Даны две таблицы A и B одинаковой размерности $m \times n$. Каждая строка таблицы A содержит количество различных изделий, выпущенных определенным цехом завода. Каждый цех выпускает n различных изделий. Структура таблицы B идентична таблице A , но содержит количество изделий, которое надо выпустить по плану. Номер строки соответствует номеру цеха завода.

Составить одномерный массив из номеров цехов, которые недовыполнили план более чем на 10 % по общему количеству выпущенных изделий.

29. Даны две таблицы A и B одинаковой размерности $m \times n$. Каждая строка таблицы A содержит количество различных изделий, выпущенных определенным цехом завода. Каждый цех выпускает n различных изделий. Структура таблицы B идентична таблице A , но содержит количество изделий, которое надо выпустить по плану. Номер строки соответствует номеру цеха завода.

Составить одномерный массив из номеров изделий, которые не выпустил хотя бы один цех.

30. Составьте подпрограмму, которая в строке текста определяет позицию самого правого вхождения указанной подстроки. Для указанной строки определите позицию подстроки *END*.

31. Напишите функцию, которая для заданного целого числа получает его представление в системе счисления с основанием n в виде строки символов.

32. Составьте функцию вычисления значения целого числа по заданной строке символов, являющейся записью этого числа в системе счисления с основанием n ($n \leq 16$). Найдите сумму чисел, записанных в виде строки, для указанного основания системы счисления.

33. Пусть дано n треугольников. Посчитайте количество треугольников, лежащих в каждой координатной четверти и не пересекающих оси ко-

ординат. Используйте функцию для определения координатной четверти, в которой лежит треугольник.

34. Пусть дано n вещественных чисел ($n=100$). Упорядочить их по неубыванию методом фон Неймана:

завести два массива A и B и занести все элементы в A ;

упорядочить пары соседних элементов (A_1 и A_2 , A_3 и A_4 и т. д.) и записать их в B ;

взять из массива B упорядоченные пары и, слив их в упорядоченные четверки, снова записать в A ;

затем взять из массива B четверки элементов, восьмерки и т. д.

35. Составьте функцию «сжатия» исходной последовательности символов, которая заменяет последовательность, состоящую из одинаковых символов, текстом вида $x(k)$, где x – символ последовательности, k – число вхождений. Определите для указанной последовательности коэффициент сжатия (отношение исходной длины последовательности к полученной).

36. Пусть дано n прямоугольников, заданных координатами левой верхней и правой нижней вершины. Стороны прямоугольников параллельны осям координат. Определите пару прямоугольников с максимальной площадью пересечения. Напишите функцию для определения площади пересечения двух прямоугольников.

37. Пусть дано n треугольников. Определите координаты центра и минимальный радиус круга, который содержит все треугольники.

38. Пусть дано n треугольников. Определите количество треугольников, которые пересекают границы заданного квадрата. Используйте функцию для определения, пересекает ли треугольник границы квадрата.

39. Пусть дано n отрезков на интервале $[A, B]$. Определите часть интервала, который покрывается наибольшим количеством отрезков. Напишите функцию для определения количества отрезков, покрывающих заданный интервал.

40. Определите минимальное количество отрезков из числа заданных, которые покрывают заданную часть прямой. Напишите функцию, определяющую, покрывает ли заданный набор отрезков необходимую часть прямой.

41. Напишите функцию сложения двух многочленов. Представьте каждый многочлен в виде массива коэффициентов, где A_i – коэффициент перед x^i .

42. Найдите многочлен, полученный в результате умножения $(a_0 + a_1x + a_2x^2 + \dots + a_nx^n) \times (b_0 + b_1x + b_2x^2 + \dots + b_nx^n)$. Напишите функцию для умножения многочлена на число и x^i .

43. Напишите функцию, которая для заданного полинома определяет полином, являющийся производной исходного полинома по указанной переменной. Вычислите:

$$\left(\frac{a_0 + a_1x + a_2x^2 + \dots + a_nx^n}{(b_0 + b_1x + b_2x^2 + \dots + b_mx^m)(b_m + b_{m-1}x + b_{m-2}x^2 + \dots + b_0x^m)} \right)'.$$

44. Вычислите интегралы:

$$\int_a^b \sin^2 x \frac{1}{1+x^2} dx \text{ и } \int_a^b \operatorname{ctg}(x^2 + 4) dx.$$

Для этого напишите функцию, вычисляющую интеграл по формуле наименьших прямоугольников для заданного интервала, количества разбиений и функции, передаваемой как параметр. Формула наименьших прямоугольников:

$$\int_a^b f(x) dx = \min(f(a=a_0), f(a_1)) + \min(f(a_1), f(a_2)) + \dots + \\ + \min(f(a_{n-1}), f(b=a_n)).$$

45. Вычислите интеграл:

$$\int_2^{10} 3 \sin(x^2 + 1) dx.$$

Для этого напишите функцию, которая вычисляет интеграл по формуле трапеций для заданного интервала, количества разбиений и функции, передаваемой как параметр. Формула трапеций:

$$\int_a^b f(x) dx = \frac{h}{2} (f(a_0) + 2f(a_1) + 2f(a_2) + 2f(a_3) + \dots + 2f(a_{n-1}) + f(a_n)).$$

46. Напишите функцию, которая вычисляет интеграл по формуле Симпсона для заданного интервала, количества разбиений и функции, передаваемой как параметр.

$$\int_a^b f(x) dx = \frac{h}{3} (f(a_0) + 4f(a_1) + 2f(a_2) + 4f(a_3) + 2f(a_4) + \dots + 4f(a_{n-1}) + f(a_n)).$$

47. В квадратной матрице A найдите строки, представляющие вектор с максимальной нормой в каждой из следующих метрик (напишите функцию для подсчета нормы вектора в указанной метрике):

$$\|a\| = \sqrt{a_1^2 + \dots + a_n^2}$$

$$\|a\| = |a_1| + \dots + |a_n|$$

$$\|a\| = \max |a_i|, \quad i = 1, \dots, n.$$

48. По вещественным числам $E > 0$ и t вычислите с точностью E величину:

$$\sqrt[4]{1 - \frac{\cos^4 t}{4}} + \sqrt[5]{1 + \frac{\operatorname{arctg} t}{4}} \cdot \sqrt[9]{\frac{1}{3 + t^2}}.$$

Для вычисления корней используйте следующий ряд Тейлора:

$$(1+x)^a = 1 + \frac{a}{1!}x + \frac{a(a-1)}{2!}x^2 + \frac{a(a-1)(a-2)}{3!}x^3 + \dots, \quad |x| \leq 1, \quad a > 0.$$

49. Пусть дана квадратная матрица A порядка n . Постройте логический вектор, каждый элемент которого принимает значение true, если среди элементов i -й строки матрицы A есть хотя бы два равных, и значение false – в противном случае. Воспользуйтесь логической функцией, которая для i -й строки выполняет указанную проверку.

50. Пусть дана матрица A . Постройте вектор, каждый элемент которого равен наибольшему количеству равных элементов в соответствующей строке матрицы A . Воспользуйтесь функцией, которая определяет это количество в i -й строке матрицы A .

51. Пусть дана матрица A . Постройте логический вектор, каждый элемент которого равен true, если в строке существует элемент, делящий весь массив на две части с одинаковой суммой элементов в каждой.

52. Проверьте, верно ли, что количество строк матрицы A , в которой все числа нечетные; кратно заданному числу x .

53. Пусть дана матрица A . Постройте логический вектор, каждый элемент которого равен true, если среди элементов соответствующей строки матрицы A есть хотя бы один элемент, принадлежащий отрезку $[0,5; 1]$, и false в противном случае. Воспользуйтесь логической функцией, делающей соответствующую проверку в i -й строке.

54. Пусть дана матрица A . Постройте вектор, каждый элемент которого равен количеству элементов в i -й строке матрицы A , не принадлежащих отрезку $[0, 10]$. Воспользуйтесь функцией, делающей соответствующую проверку в i -й строке.

55. Пусть дана матрица A . Постройте вектор, каждый элемент которого равен сумме элементов i -й строки матрицы A , больших, чем значение

минимального элемента в этой строке. Воспользуйтесь функцией, определяющей соответствующую операцию в каждой строке матрицы A .

56. Пусть дана матрица A . Упорядочьте строки по неубыванию сумм цифр элементов этой строки. Воспользуйтесь функцией, определяющей для каждого числа сумму его цифр.

57. Напечатайте строки и столбцы квадратной матрицы W порядка n , на пересечении которых находятся максимальные и минимальные элементы, если элементы матрицы вычисляются по формулам:

$$W_{ij} = \begin{cases} i^2 + j, & i < j, \\ 1/2, & i = j, \\ i + j^2, & i > j. \end{cases}$$

58. Элементы квадратной матрицы A порядка n вычисляются по формуле:

$$A_{ij} = i * \sin(j) + \sin(i), \quad i, j = 1, 2, 3, \dots, n.$$

Требуется:

а) сформировать матрицу B :

$$B_{ij} = \frac{A_{ij}}{\sqrt{A_{ij}^2 + i^2 + j^2}}, \quad i, j = 1, 2, 3, \dots, n;$$

б) вывести на печать сформированные матрицы A , B и их произведение $A * B$.

59. Выведите на печать квадратную матрицу K порядка n . Вычислите и выведите на печать скалярное произведение i -й строки на i -й столбец (всего n произведений). Элементы матрицы вычисляются по формуле:

$$K_{ij} = \begin{cases} 1, & i > j, \\ 0, & i = j, \\ j, & i < j. \end{cases}$$

60. Элементы квадратной матрицы A порядка n вычисляются по формуле:

$$A_{ij} = \sin(i * j), \quad i, j = 1, 2, 3, \dots, n.$$

Требуется:

а) построить матрицу B , каждый элемент которой вычисляется по формуле:

$$B_{ij} = \frac{1}{j} \sum_{k=1}^j A_{ik};$$

б) вывести на печать матрицы A и B .

61. Подсчитайте количество строк квадратной матрицы A порядка n , элементы которых образуют монотонную последовательность. Для определения факта монотонности используйте функцию.

62. Уплотните квадратную матрицу A порядка n влево и вверх, удалив строки и столбцы, состоящие из нулей. Для выявления нулевых строк и столбцов используйте функцию.

11.4. Библиотечные математические функции

Математические функции C++ определены в заголовочном файле `<cmath>`

Функция	Описание	Пример
abs(a)	модуль или абсолютное значение a	<code>abs(-3)=3</code> <code>abs(-5.0)=5.0</code>
fabs(a)	модуль или абсолютное значение a , где a типа <code>double</code>	<code>fabs(-3.0)=3.0</code> <code>fabs(5.0)=5.0</code>
pow(a, b)	возведение a в степень b	<code>pow(2, 3)=8</code>
sqrt(a)	корень квадратный из a , причём a не отрицательно	<code>sqrt(9.0)=3.0</code>
floor(a)	округление a до наибольшего целого, но не больше чем a (округление вниз)	<code>floor(12.4)=12</code> <code>floor(-2.9)=-3</code>
ceil(a)	округление a до наименьшего целого, но не меньше чем a (округление вверх)	<code>ceil(2.3)=3.0</code> <code>ceil(-2.3)=-2.0</code>
round(a)	округление a по правилам математики	<code>round(7.8)=8</code> <code>round(7.2)=7</code>
modf(a, b)	отделяет дробную часть от целой. Целая часть числа a сохраняется в объект, на который указывает b , а дробная часть возвращается функцией.	<code>a=3.14159</code> <code>modf(a, b)=0.14159</code> <code>b=3.0</code>
fmod(a, b)	вычисление остатка от a/b	<code>fmod(4.4, 7.5)=4.4</code> <code>fmod(7.5, 4.4)=3.1</code>
exp(a)	вычисление экспоненты e^a	<code>exp(0)=1</code>
log(a)	натуральный логарифм a	<code>log(1.0)=0.0</code>

Функция	Описание	Пример
	(основанием является экспонента)	
log10(a)	десятичный логарифм a	$\log_{10}(10)=1$
sin(a)	синус a , где a задаётся в радианах	
cos(a)	косинус a , где a задаётся в радианах	
tan(a)	тангенс a , где a задаётся в радианах	
asin(a)	арксинус a , где $-1.0 < a < 1.0$	$\text{asin}(1)=1.5708$
acos(a)	арккосинус a , где $-1.0 < a < 1.0$	
atan(a)	арктангенс a	
atan2(a,b)	арктангенс значения a/b	
sinh(a)	гиперболический синус a	
cosh(a)	гиперболический косинус a	
tanh(a)	гиперболический тангенс a	

Округление по правилам математики: $\text{floor}(x + 0.5)$ или $\text{int}(x + 0.5)$.

Глава 12. ФАЙЛОВЫЙ ВВОД/ВЫВОД ДАННЫХ

12.1. Работа с файлами

12.1.1. Некоторые справочные сведения

После объявления потока (`ifstream` – имя класса входных файловых потоков, `ofstream` – имя класса выходных файловых потоков, `fstream` – имя класса программно направляемых файловых потоков) можно указать режим открытия файла, задаваемый вторым параметром функции-метода `open()`.

Режимы открытия файла	Описание
<code>ios::in</code>	Открыть файл для чтения. Является режимом по умолчанию для потоков класса <code>ifstream</code> .
<code>ios::out</code>	Открыть файл для записи. Является режимом по умолчанию для потоков класса <code>ofstream</code> .
<code>ios::ate</code>	При открытии переместить указатель в конец файла
<code>ios::app</code>	Открыть файл для записи в конец файла
<code>ios::trunc</code>	Удалить содержимое файла, если он существует
<code>ios::binary</code>	Открытие файла в двоичном режиме
<code>ios::nocreate</code>	Если указанный файл не существует, не создавать файл и вернуть ошибку.
<code>ios::noreplace</code>	Если файл существует, прервать операцию и вернуть ошибку.

Наиболее часто используемые операции при работе с файлами.

Операции и функции	Описание
<code><<</code> <code>>></code>	Операции перенаправления ввода/вывода
<code>getline()</code> , <code>get()</code> , <code>put()</code>	Функции (методы) чтения и записи символов и строк. Если данные считываются в массив <code>char[]</code> , то применяются функции (методы) класса <code>istream</code> : <code>file.getline()</code> и <code>file.get()</code> , если в строку <code>string</code> , то функции (методы) класса <code>string</code> : <code>getline()</code> и <code>get()</code> <i>Примеры:</i> 1) ... <code>char* buffer = new char[n+1];</code>

Операции и функции	Описание
	<pre> buffer[n]=0; file.get(buffer,n); file.getline(buffer,n,' '); ... 2) ... string s; getline(file,s); ... </pre>
write(), read()	<p>Функции (методы) блочных записи и чтения и обмена данными с бинарным файлом. Так как текстовый файл является частным случаем бинарного, эти методы применимы и к текстовому файлу. <i>Примеры:</i> 1) ... char* buffer=new char[n+1]; buffer[n]=0; file.read(buffer,n); ... 2) ... string buf[2]; cin>>buf[0]>>buf[1]; file.write((char *)&buf, sizeof(buf)); ...</p>
fill(), width(), precision()	<p>Функции (методы) настройки форматированного вывода для операции >> <i>Пример:</i> ... // устанавливает символ заполнитель cout.fill('*'); // задает ширину поля cout.width(20); // задает количество знаков после десятичной // точки cout.precision(5); ...</p>
eof()	Функция (метод) проверки, достигнут ли конец файла.
is_open()	Функция (метод) проверки, открыт ли файл. Возвращает 1, если файл найден и успешно открыт, 0 – в противном случае.

Операции и функции	Описание
tellg(), tellp(), seekg(), seekp()	<p>Функции (методы) позиционирования в файле.</p> <p><i>Примеры:</i></p> <p>1) ... //переместить указатель в конец файла file.seekg(0,ios_base::end); //переместить указатель на 10-й байт с конца file.seekg(10,ios_base::end); //переместить указатель на 31-й байт file.seekg(30,ios_base::beg); //переместить указатель через 3 байта от текущей // позиции file.seekg(3,ios_base::cur); //переместить указатель через 3 байта от текущей // позиции file.seekg(3); ... </p> <p>2) ... file.seekg(0, ios_base::end); cout << "Размер файла (в байтах): " << file.tellg(); ... </p> <p>3) ... string a[4]; file.read((char*)&a, sizeof(a)); ... </p>
fail()	<p>Функция (метод) fail()</p> <p>Возвращает ненулевое значение, если при выполнении операции с файлом произошла ошибка, и 0 – в противном случае.</p> <p><i>Пример:</i></p> <p>... if (file.fail()) cout<<"Ошибка!"<<endl; ... </p>
peek()	<p>Функция (метод) peek()</p> <p>Возвращает значение очередного символа потока или –1 (eof), но не перемещает указатель на следующий байт.</p>
c_str()	<p>Функция (метод) класса string</p> <p>Преобразует строку класса string в массив символов ти-</p>

Операции и функции	Описание
	па char* и возвращает указатель на него. <i>Пример:</i> ... ifstream f (name.c_str()); ...

12.1.2. Вопросы для самопроверки

1. Что такое файл данных?
2. Как определяется длина (размер) файла?
3. Какие типы файлов данных используются в программе на языке C++ ?
4. В чем состоит последовательный характер файла данных?
5. Какой объект в программе на языке C++ служит для выполнения операций с файлами?
6. Какое действие с потоком совершается при выполнении оператора ввода?
7. Какое действие с потоком совершается при выполнении оператора вывода?
8. Какой поток считается входным? Выходным?
9. В каких файлах содержатся определения классов потоков?
10. Как в программе поток привязывается к конкретному файлу на носителе данных?
11. В чем состоит операция открытия файла данных?
12. Какого типа данные могут быть элементами файла? Какого типа данные не могут быть элементами файла?
13. Какого типа файлы и потоки можно использовать для вывода на внешний носитель последовательности значений целочисленных переменных?
14. Какого типа файлы и потоки можно использовать для вывода на внешний носитель последовательности значений символьных переменных?
15. Какие основные операции применяются к файлам данных?
16. Какие инструментальные средства для выполнения операций над файлом данных предоставляет язык программирования C++?

17. Можно ли применять операцию ввода к потоку, объявленному как выходной?

18. Какими инструментальными средствами языка программирования можно вводить данные из текстового файла?

19. Какими инструментальными средствами можно вводить данные из бинарного файла?

20. Какие действия происходят при выполнении операции закрытия файла?

21. Может ли файл данных быть пустым?

22. Каков результат выполнения следующей программы

```
#include <fstream>
#include <iostream>
using namespace std;
int main()
{
    ifstream f;
    int i;
    f.open(primer.txt);
    while (!f.eof())
    {
        f >> i; cout << i;
    }
    f.close();
    return 0;
}
```

Какие действия вызовет программа в случае, если файл `primer.txt` пуст?

Как изменится результат работы программы, если оператор `while` заменить на оператор

```
do
{
    f >> i; cout << i;
}
while (!f.eof());
```

23. Какие инструментальные средства предоставляет среда программирования на языке C++ для управления форматированием данных при выводе в файл?

24. К какого типа файлам можно применять инструменты форматирования?

25. Являются ли манипуляторы встроенными в компилятор функциями?
26. Какие функции, привязанные к потоку, служат для ввода символа? Какие – для ввода строки символов?
27. Как разделяются строки в текстовом файле?
28. Какое значение принимает выражение, составленное из имени потока, в случае, если доступа к файлу нет?
29. Назовите причины, по которым невозможен доступ к файлу.
30. Каким в программе устанавливается тип файла по умолчанию – текстовым или бинарным?
31. Какими способами в программе можно задать характер потока – входной или выходной?
32. К каким файлам можно применять функции прямого доступа?

12.1.3. Упражнения и задачи с решениями

Во всех последующих примерах и заданиях формулировка «**требуется**» означает необходимость создать и отладить программу на языке C++ , реализующую условие задания.

В условиях заданий, если это не оговорено, предполагается использование текстовых файлов.

Для корректного вывода в окно результата кириллического текста можно использовать один из следующих вариантов:

- 1) включить в текст программы настройку `setlocale (0, "");`
- 2) включить в текст программы настройку `setlocale(LC_ALL, "Russian");`
- 3) подключить к программе заголовочный файл `windows.h`, включить в текст программы настройки `SetConsoleCP(1251);` `SetConsoleOutputCP(1251);` и установить в окне вывода результатов шрифт `Lucida Console`.

1. Требуется создать файл, поместив в него одно целое число, одно дробное число и текст "Мой первый файл.". Числа вводятся с клавиатуры.

```
#include<iostream>
#include<fstream>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
```

```

int i; float x;
//Создаем файл с именем test.txt и открываем его
// для вывода данных
ofstream out ("D://test.txt");
if(!out)
{
    cout << "Не удастся создать файл.\n";
    return 1;
}
cout << "Введите целое число \n"; cin>>i;
cout << "Введите дробное число\n"; cin >> x;
// Вывод данных в файл
out << i << " " << x << '\n';
out <<"Мой первый файл.";
out.close(); //Закрываем файл.
return 0;
}

```

2. Требуется вывести на экран дисплея содержимое файла test.txt, созданного по предыдущему заданию.

```

#include<iostream>
#include<fstream>
#include<string>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int i;
    float x;
    string a;
    // Открываем файл с именем test.txt для ввода данных
    ifstream in ("D://test.txt");
    if(!in)
    {
        cout << "Не удастся открыть файл.\n";
        return 1;
    }
    in >> i; cout << "Первое число - " << i;
    in >> x; cout << "Второе число - " << x;
    in.getline() >>a; cout << "Текст - " << a;
    in.close(); //Закрываем файл.
    return 0;
}

```

3. Требуется записать содержимое первой строки файла `test.txt`, содержащего несколько строк, в другой файл, имя которого задает пользователь.

```
#include <iostream>
#include <fstream>
#include <string>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    ofstream fout;
    ifstream fin;
    string a;
    char c[100];
    fin.open ("D://temp/1.txt");
    getline(fin, a);
    fin.close();
    cout << "Введите имя файла: " << endl;
    cin >> c;
    fout.open (c);
    fout << a;
    fout.close();
    system("PAUSE");
    return 0;
}
```

4. Требуется вывести на экран содержимое файла, состоящего из нескольких строк, имя которого задает пользователь.

```
#include <iostream>
#include <fstream>
#include <string>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    ifstream fin;
    string a;
    char c[100];
    cout << "Введите полное имя файла "; cin >>c;
    fin.open (c);
```

```

while (getline (fin, a))
    // при чтении из файла символ перехода к новой строке
    // файла не вводится, поэтому при выводе в cout нужно
    // добавить endl или '\n'
    cout << a<<endl;
fin.close();
system("PAUSE");
return 0;
}

```

5. Требуется создать файл out.dat, являющийся копией файла in.dat. Файл in.dat содержит строки символов, разделенных парами символов '\r' и '\n', длина файловой строки не превосходит 128 символов.

```

#include <iostream>
#include <fstream>
#include <string>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    char buf [128];
    ifstream fin;
    ofstream fout;
    fin.open ("in.dat");
    if (!fin)
    {
        cout << "Не удалось открыть файл in.dat";
        return 1;
    }
    fout.open ("out.dat");
    if (fout.fail())
    {
        cout << "Не удалось создать файл out.dat";
        return 1;
    }
    while (fin.getline (buf, sizeof(buf))
        fout<<buf<<endl;
    fout.close();
    fin.close();
    system("PAUSE");
    return 0;
}

```

6. Имеется текстовый файл. Требуется вывести на экран строки файла, длина которых меньше заданной.

```
#include <iostream>
#include <fstream>
#include <string>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    ifstream f("D://temp/in.dat");
    int n;
    cout << "Введите длину строки\n";
    cin >> n;
    string s;
    if (f.is_open())
    {
        while (getline(f,s))
            if (s.size()<n)
                cout << s << endl;
    }
    else
        cout << "Файл \"D://temp/in.dat\" не открылся";// символ'\''
        - экранирующий
    f.close();
    system("PAUSE");
    return 0;
}
```

7. Имеется текстовый файл in.txt. Требуется переписать его содержимое в новый файл out.txt, удалив пустые строки.

```
#include <iostream>
#include <fstream>
#include <string>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    string str;
    ifstream fin;
    ofstream fout;
```

```

fin.open ("in.txt");
if (!fin)
{
    cout << "Не удалось открыть файл in.dat";
    return 1;
}
fout.open ("out.txt");
if (fout.fail())
{
    cout << "Не удалось создать файл out.dat";
    return 1;
}
while (!fin.eof())
{
    getline (fin, str);
    if (str.size()>0)
        fout<<str<<endl;
}
fout.close();
fin.close();
system("PAUSE");
return 0;
}

```

8. Имеется файл, в котором записан код программы на языке C++. Требуется определить, соблюдается ли баланс открывающих и закрывающих операторных скобок. В выходной файл D://temp/out.txt следует записать одну из текстовых строк: «Баланс скобок соблюдается» либо «Баланс скобок не соблюдается». Имя входного файла задается вводом с клавиатуры.

```

#include <iostream>
#include <fstream>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    char nazv[20]; // имя файла
    cout << "Введите название файла:";
    cin >> nazv;
    ifstream infile(nazv); // открываем файл для чтения «»
    if(!infile.is_open())
        cout<<"Файл не может быть открыт";
}

```

```

else
{
    int otkr=0;//счетчик количества скобок «{»
    int zakr=0;//счетчик количества скобок «}»
    bool flag=true;
    char symv;
    infile >> symv; // считываем один символ из входного файла
    while(!infile.eof() && flag)
    {
        if (symv=='{') otkr++;
        if (symv=='}')
        {
            zakr++;
            if(zakr > otkr)
                flag=false;
        }
        infile>>symv;//считываем следующий символ
    }
    infile.close();
    //открываем выходной файл для записи
    ofstream outfile("D://temp/out.txt");
    if (!outfile.is_open())
        cout<<"Не удалось открыть выходной файл";
    if (otkr==0 && zakr ==0)
    {
        cout<<"В исходном тексте открывающих и закрывающих
                скобок нет";
        outfile<<"Открывающих и закрывающих скобок нет";
    }
    else
    {
        if(otkr ==zakr)
        {
            cout << "Скобки сбалансированы";
            outfile<<"Баланс скобок соблюдается ";
        }
        else
        {
            cout<<" Скобки не сбалансированы ";
            outfile<<"Баланс скобок не соблюдается ";
        }
    }
}
system ("PAUSE");

```

```

    return 0;
}

```

9. Имеется текстовый файл in.dat. Слова в тексте разделены одним или несколькими пробелами. Требуется вывести на экран строку файла, содержащую наибольшее количество слов. Если таких строк несколько, вывести любые две из них.

```

#include <iostream>
#include <fstream>
#include <string>
#include <windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    ifstream f("D://temp/in.txt");
    int count, count1_max=0, count2_max=0;
    int slovo;//признак того, что символ является началом слова
    string s, s1_max, s2_max;
    if (!f.is_open())
    {
        cout << "Файл \"D://temp/in.dat\" не открылся";
        //символ \ экранирующий
        return 1;
    }
    while (getline(f,s))
    {
        int i = 0; count=0;
        while (s[i] == ' ' && s[i] != '\0')//пропуск пробелов
            i++;
        slovo = 0;//состояние символа – не слово
        while (s[i] != '\0')
        {
            if (s[i] != ' ' && slovo == 0)
            {
                slovo = 1; //состояние символа – начало слова
                count++;
            }
            else
                if (s[i] == ' ')
                    slovo = 0;//состояние символа – не слово
            i++;
        }
    }
}

```



```

    if (count>count1_max)
    {
        count2_max=count1_max;
        count1_max=count;
        s2_max=s1_max;
        s1_max=s;
    }
    else
        if (count>count2_max)
        {
            count2_max=count;
            s2_max=s;
        }
    }
    if (count2_max==count1_max)
        cout<<"Строки с наибольшим числом слов ->" << s1_max <<
            "->"<<s2_max<<endl;
    else
        cout<<"Строка с наибольшим числом слов ->" << s1_max <<
            endl;
    f.close();
    system("PAUSE");
    return 0;
}

```

10. Имеется файл целых чисел `in.dat`. Требуется создать два новых файла: первый – из положительных, второй – из отрицательных значений исходного файла. Если какой-то из создаваемых файлов окажется пустым, вывести на экран сообщение об этом.

```

#include <iostream>
#include <fstream>
#include <string>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    char buf [128];
    ifstream fin;
    ofstream fout1;
    ofstream fout2;
    bool flag1=true, flag2=true; //файлы fout1 и fout2 пусты
    int x;

```

```

fin.open ("in.dat");
if (!fin)
{
    cout << "Не удалось открыть файл in.dat";
    return 1;
}
fout1.open ("out1.dat");
if (fout1.fail())
{
    cout << "Не удалось создать файл out1.dat";
    return 1;
}
fout2.open ("out2.dat");
if (fout2.fail())
{
    cout << "Не удалось создать файл out2.dat";
    return 1;
}
while (fin>>x)
    if(x>0)
    {
        fout1<<x<<' ';flag1=false;
    }
    else
        if(x<0)
        {
            fout2<<x<<' ';flag2=false;
        }
fout1.close(); fout2.close();
fin.close();
if (flag1)
    cout<<"Положительных чисел в исходном файле нет";
if (flag2)
    cout<<"Отрицательных чисел в исходном файле нет";
system("PAUSE");
return 0;
}

```

11. Имеется файл целых чисел, содержащий информацию об успеваемости студентов: количество оценок студента (<10) и сами оценки. Например: 34526554434355545234 – первый студент имеет 3 оценки 4, 5, 2, второй студент имеет 6 оценок 5, 5, 4, 4, 3, 4, третий студент имеет 4 оценки 3, 5, 5, 5, четвертый студент имеет 4 оценки 5,2,3,4. Требуется определить порядко-

вый номер студента, имеющего наибольшую среднюю оценку. Если таких студентов несколько, требуется вывести порядковый номер любого из них.

```
#include <iostream>
#include <fstream>
#include <string>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    float sred; // средняя оценка студента
    int size; // количество оценок студента
    int sum, tmp;
    int k=0; // текущий порядковый номер подстроки оценок
    int k_max; // порядковый номер подстроки с максимальным
                // средним баллом
    char ch;
    ifstream in("D://temp/stud.txt");
    while ( in.get(ch) )
    {
        k++; sum=0;
        size=ch-'0';
        for(int i = 0; i < size; i++)
        {
            in >> ch;
            tmp=ch-'0';
            sum+=tmp;
        }
        sred=(float)sum/size;
        if (sred>sr_max)
        {
            sr_max=sred; k_max=k;
        }
    }
    cout<<"Наибольшая средняя оценка:"<<sr_max<<"\n";
    cout<<"Порядковый номер студента:"<<k_max<<endl;
    system("PAUSE");
    return 0;
}
```

12. Имеется файл целых чисел `matr.dat`, образованный из строк квадратной целочисленной матрицы размером 100×100 . Требуется определить, содержит ли матрица нулевые строки.

```

#include <iostream>
#include <fstream>
#include<windows.h>
using namespace std;
int main( )
{SetConsoleCP(1251);
 SetConsoleOutputCP(1251);
 ifstream fin;
 int c;
 bool flag=true;// наличие нулевой строки матрицы
 fin.open ("D://temp/matr.dat");
 flag = false;
 while (!fin.eof() && !flag)
 {flag=true;
  for (int i=0; i<100 && flag; i++ )
  {fin>>c;
   flag=c==0;
  }
 }
 if (!flag)
  cout<<"Нулевых строк в матрице нет\n";
 else
  cout<<"Есть нулевые строки\n";
 fin.close();
 system("PAUSE");
 return 0;
}

```

13. Имеются шесть файлов целых чисел одинакового размера. Требуется создать новый файл, в котором чередовались бы элементы исходных файлов с одним и тем же порядковым номером в каждом файле. Например:

```

исходный файл_1:    20 10 4 -1 -2 0 0
исходный файл_2:    4 10 30 -6 -8 0 3
исходный файл_3:    40 15 8 1 2 0 0
исходный файл_4:    2 10 46 12 21 7 7
исходный файл_5:    0 0 43 -15 -20 0 6
исходный файл_6:    52 21 4 -21 2 0 15
файл-результат: 20 4 40 2 0 52 10 10 15 10 0 21 4 30 8 46 43 4 1
-6 1 12 -15 -21 -2 -8 2 21 -20 2 0 0 0 7 0 0 0 3 0 7 6 15

```

```

#include <iostream>
#include <fstream>
#include<windows.h>
using namespace std;

```

```

int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int x; char s[23];
    // массив из семи потоков: шесть входных и один выходной
    fstream f[7];
    for (int i = 0; i < 7; ++i)
    {
        cout<<"Введите имя "<<i+1<<"-го файла"; cin >> s;
        if (i < 6)
        {
            f[i].open(s, ios::in);
            if (!f[i])
            {
                cout << "Не удалось открыть файл_"<<i<<endl;;
                return 1;
            }
        }
        else
            f[i].open(s, ios::out);
    }
    while (!f[0].eof())
        for (int i = 0; i < 6; ++i)
        {
            f[i]>>x; f[6]<<x<<" ";
        }
    for (int i = 0; i < 7; ++i)
        f[i].close();
    system("PAUSE");
    return 0;
}

```

14. Имеется бинарный файл вещественных чисел. Имя файла задается вводом с клавиатуры. Требуется заменить в нем все числа, большие минимального, на их квадраты.

```

#include <cstdlib>
#include <iostream>
#include <fstream>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);

```

```

SetConsoleOutputCP(1251);
string s;
cin >> s;
// создание потока, связанного
// с бинарным файлом, который будет использоваться
// и для ввода и для вывода;
// функция c_str() преобразует объект типа string
// в массив символов
fstream f(s.c_str(), ios::binary | ios::in | ios::out);
// перемещение файлового указателя на конец файла
f.seekg(0, ios::end);
int len = f.tellg(), // номер позиции последнего байта файла
size = sizeof(double); // размер памяти для одного числа
f.seekg(0); // возвращение файлового указателя на начало файла
f.read((char *)&x_min, sizeof(x));
for (int i = 1; i < len / size; ++i)
// len / size - количество чисел в файле
{
    f.read((char *)&x, sizeof(x));
    if (x < x_min) x_min = x;
}
f.seekg(0);
for (int i = 0; i < len / size; ++i)
{
    f.read((char *)&x, sizeof(x));
    if (x > x_min) x = x * x;
    f.seekg(size * i); // возвращение указателя файла
                        // на очередную позицию
    f.write((char *)&x, sizeof(x));
}
f.close();
system("PAUSE");
return 0;
}

```

15. Студентам 4 курса университета была предложена анкета, в которой требовалось указать, знание каких из изученных в университете курсов могут потребоваться в будущей работе. Результаты анкетирования требуется занести в файл D://temp/anketa.dat. Список изученных курсов:

- математический анализ;
- алгебра;
- информатика;
- численные методы;
- базы данных;

- теория вероятностей;
- web-программирование.

```
#include <iostream>
#include <fstream>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    bool answers [7];// экземпляр анкеты
    string name[7];// названия предметов
    char ch, res;
    ofstream fout;
    bool flag=true,
    fout.open ("D://temp/anketa.dat");
    if (fout.fail())
    {
        cout << "Не удалось создать файл out1.dat";
        return 1;
    }
    name[0]="математический анализ";
    name[1]="алгебра";
    name[2]="информатика";
    name[3]="численные методы";
    name[4]="базы данных";
    name[5]="теория вероятностей";
    name[6]=" web-программирование";
    cout<<"Добавить анкету в файл?";
    cout<<"(Д – да или любой другой символ - нет";
    cin>>ch;
    while (ch=='Д')
    {
        cout <<"Введите ответы:";
        for (int i=0; i<7; i++)
        {
            cout<<names[i]<<"-";
            cin>>res;
            answers[i]=res=='Д';
            fout<<name[i]<<' '<<answers[i];
        }
        cout<<"Добавить запись в файл?";
        cout<<"Д – да или любой другой символ - нет";
    }
}
```

```

        cin>>ch;
    }
    fout.close();
    system("PAUSE");
    return 0;
}

```

16. Выполнить задание 15, используя блочные операции вывода данных.

```

#include <iostream>
#include <fstream>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    string name[7];// названия предметов
    char ch, res;
    string s;
    cout << "Введите имя файла:";
    cin >> s;
    //создание бинарного файла для записи анкет-структур
    fstream fout (s.c_str(), ios::binary | ios::out);
    if (fout.fail())
    {
        cout << "Не удалось создать файл";
        return 1;
    }
    struct
    {
        string subj;
        bool voice;
    } ank;
    name[0]="мат.анализ";
    name[1]="алгебра";
    name[2]="информатика";
    name[3]="численные методы";
    name[4]="базы данных";
    name[5]="теория вероятностей";
    name[6]=" web-программирование";
    cout << "Добавить анкету в файл?";
    cout << "(Д – да или любой другой символ - нет";
    cin >> ch;
    while (ch=='Д')

```



```

{
    cout << "Введите ответы:";
    for (int i=0; i<7; i++)
    {
        cout << name[i]<<"->";
        cin >> res;
        ank.subj=name[i];
        ank.voice= res=='Д';
        fout.write((char *)&ank, sizeof(ank));
    }
    cout << "Добавить запись в файл?";
    cout << "(Д – да или любой другой символ - нет";
    cin >> ch;
}
fout.close();
system("PAUSE");
return 0;
}

```

17. Требуется создать бинарный файл `result.dat`, содержащий итоги экзаменационной сессии. На каждом из четырех курсов шесть студенческих групп, в каждой группе не более 30 человек. В файл заносится следующая информация по каждой студенческой группе:

- количество студентов в группе;
- количество студентов, допущенных к сессии;
- количество студентов, сдавших экзамены на «отлично»;
- количество студентов, сдавших экзамены на «отлично» и «хорошо»;
- количество студентов, получивших хотя бы одну оценку «неудовлетворительно».

```

#include <iostream>
#include <fstream>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int ngr=6;//число групп на курсе
    const int nc=5;// число курсов
    const int num=30;//максимальное число студентов в группе
    struct data //данные о группе
    {
        int num_stud;

```

```

        int num_admit;
        int num_ex;
        int num_ex_good;
        int num_not;
    } gr_data;
    string s;
    cout << "Введите имя файла:"; cin>> s;
    //создание бинарного файла для записи данных
    fstream fout (s.c_str(), ios::binary | ios::out);
    if (fout.fail())
    {
        cout << "Не удалось создать файл";
        return 1;
    }
    for (int ic=1; ic<=nc; ic++)
    {
        cout<<"Курс -"<<ic;
        for(int ig=1; ig<=ngr; ig++)
        {
            cout << "Группа->" << ig << endl;
            cout << "Количество студентов = ";
            cin >> gr_data.num_stud;
            cout << "\nДопущено к сессии -";
            cin >> gr_data.num_admit;
            cout << "\nСдали на 'отлично' - ";
            cin >> gr_data.num_ex;
            cout << "\nСдали на 'хорошо' и 'отлично'";
            cin >> gr_data.num_ex_good;
            cout << "\nНеуспевающих -";
            cin>>gr_data.num_not;
            fout.write((char *)&gr_data, sizeof(data));
        }
    }
    fout.close();
    cout<<"Файл создан\n";
    system("PAUSE");
    return 0;
}

```

18. Используя инструменты прямого доступа к данным файла *result.dat*, созданного при выполнении задания 17, вывести на экран результаты экзаменационной сессии в студенческой группе, номер которой и номер курса задаются вводом с клавиатуры.

```

#include <iostream>
#include <fstream>
#include<windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int ngr=6;//число групп на курсе
    const int nc=5;// число курсов
    const int num=30;//максимальное число студентов в группе
    struct data //данные о группе
    {
        int num_stud;
        int num_admit;
        int num_ex;
        int num_ex_good;
        int num_not;
    } gr_data;;
    string s;
    cout<<"Введите имя файла:";
    cin>> s;
    //открытие бинарного файла для ввода данных
    fstream fin (s.c_str(), ios::binary | ios::in);
    if (fin.fail())
    {
        cout << "Не удалось создать файл";
        return 1;
    }
    int ic, igr;
    cout<<"Введите номер курса и группы "; cin>>ic>>igr;
    int size=sizeof(data);
    //установка указателя файла на позицию группы
    fin.seekg(size * ((ic-1)*ngr+igr-1));
    fin.read((char *)&gr_data, sizeof(data))
    cout<<"Группа->"<<igr<<endl;
    cout<<"Количество студентов = "; cout<<gr_data.num_stud;
    cout<<"\nДопущено к сессии -"; cout<<gr_data.num_admit;
    cout<<"\nСдали на 'отлично' - "; cout<<gr_data.num_ex;
    cout<<"\nСдали на 'хорошо' и 'отлично' - ";
    cout<<gr_data.num_ex_good;
    cout<<"\nНеуспевающих -"; cout<<gr_data.num_not;
    fout.close();
    system("PAUSE");
}

```

```

    return 0;
}

```

19. Используя инструменты прямого доступа к данным файла `result.dat`, созданного при выполнении задания 17, найти и вывести на экран информацию о количестве отличников на заданном курсе.

```

#include <iostream>
#include <fstream>
#include <windows.h>
using namespace std;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    const int ngr=6; //число групп на курсе
    const int nc=5; // число курсов
    const int num=30; //максимальное число студентов в группе
    struct data //данные о группе
    {
        int num_stud;
        int num_admit;
        int num_ex;
        int num_ex_good;
        int num_not;
    } gr_data;;
    string s;
    cout<<"Введите имя файла:"; cin>> s;
    //создание бинарного файла для записи данных
    fstream fin (s.c_str(), ios::binary | ios::in);
    if (fin.fail())
    {
        cout << "Не удалось создать файл";
        return 1;
    }
    int ic, igr;
    int size=sizeof(data);
    cout<<"Введите номер курса ";
    cin>>ic;
    //установка указателя файла на позицию курса
    fin.seekg(size *(ic-1)*ngr);
    int kol_otl=0;
    for(int igr=1; igr<ngr;igr++)
    {
        fin.read((char *)&gr_data, sizeof(data));

```

```

        kol_otl+=gr_data.num_ex;
    }
    cout<<"\nСдали на курсе на отлично- "; cout<<kol_otl;
    fout.close();
    system("PAUSE");
    return 0;
}

```

20. Имеется файл D://temp/guests.txt, содержащий имена приглашенных гостей. Требуется вывести на экран список приглашенных, затем добавить в файл данные о новых гостях (в любом порядке) и вывести на экран содержимое файла с обновленным списком гостей.

```

#include <iostream>
#include <fstream>
#include<windows.h>
using namespace std;
const char* f_name = "D://temp/guests.txt";
const int len=40;
int main( )
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    char ch;
    ifstream fin;
    fin.open(file_name);
    if (fin.is_open())
    {
        cout<<"Содержимое файла"<<f_name<<" до обработки:\n";
        while(fin.get(ch))
            cout<<ch;
    }
    fin.close();
    ofstream fout(f_name, ios::out | ios::app);
    if (!fout.is_open())
    {
        cerr << "Не удалось открыть файл  " << f_name <<
            "для записи" << endl;
        return 1;
    }
    cout<<"Введите имена гостей (или пустую строку, чтобы
        завершить ввод\n";
    char name [len];
    cin.get(name, len);
    while (name[0]!='\0')

```

```

{
    while (cin.get() != '\n')
        continue; //исключить \n и длинные строки
    fout<<name<<"\n";
    cin.get(name, len);
}
fout.close();
// вывод измененного файла
fin.clear();
fin.open(f_name);
if (fin.is_open())
{
    cout<<"Обновленный файл "<<f_name<<' \n';
    while (fin.get(ch))
        cout<<ch;
}
fin.close();
system("PAUSE");
return 0;
}

```

21. Имеется файл D:/temp/phone.txt, содержащий сведения о владельцах телефонов. Требуется по фамилии владельца определить номер телефона. Для завершения работы сразу после запроса фамилии нажмите <Enter>. Рекомендуемый вид диалога с пользователем:

```

ПОИСК В ТЕЛЕФОННОМ СПРАВОЧНИКЕ
Введите фамилию -> Петров
В справочнике данных о Петров нет.
Введите фамилию -> Иванов
Иванов Василий Васильевич  578-12-45
Иванов Иван Иванович  200-99-00
Введите фамилию ->

```

```

#include<fstream>
#include<iostream>
#include<windows.h>
#include<string>
#define fname "D:/temp/phone.txt"//имяфайла
using namespace std;
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    char f_name[20]=fname;

```

```

ifstream in (f_name);
string obr;//фамилия-образец для поиска
string fam;//фамилия
string name;//имя
string otc;// отчество
string tel;// номер телефона
int n=0;// количество записей, удовлетворяющих запросу
puts("\nПОИСК В ТЕЛЕФОННОМ СПРАВОЧНИКЕ\n");
if(!in.is_open())
{
    cout<<"Ошибка открытия файла ", f_name;
    return 1;
}
// получение данных от пользователя
cout << "Фамилия ->" ;
cin >> obr;
while(!in.eof())
{
    in >> fam >> name >> otc >> tel;
    if (fam==obr)
    {
        cout<<fam<< ' '<<name<<' '<<tel<<'\n';
        n++;
    }
}
if( n)
    cout<<"Найдено записей: ",n) ;
else
    cout<<"Данных об абоненте "<<obr<<" нет";
fclose(in); //закрытьфайл
cout<<"\nДля завершения работы нажмите <Enter>";
system("PAUSE");
return 0;
}

```

22. Имеется текстовый файл, имя которого задается вводом с клавиатуры. Требуется вывести на экран количество содержащихся в нем символов и строк

В первом варианте решения будем считывать файловые данные построчно:

```

#include<fstream>
#include<iostream>
#include<windows.h>
#include<string>
using namespace std;

```

```

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    string name;
    cin >> name;
    ifstream f (name.c_str());
    int nc = 0, ns = 0;
    while (f.peek() != -1)
    {
        string s;
        getline(f, s);
        nc += s.length();
        ++ns;
    }
    f.close();
    cout << nc << ns;
    system("PAUSE");
    return 0;
}

```

Второй вариант решения будет правильно анализировать текстовый файл только в том случае, если каждая строка этого файла оканчивается маркером конца строки.

```

#include<fstream>
#include<iostream>
#include<windows.h>
#include<string>
using namespace std;
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    string name;
    cin >> name;
    ifstream f(name.c_str());
    int nc = 0, ns = 0;
    while (f.peek() != -1)
    {
        char c = f.get();
        if (c == '\n')
            ++ns;
        else

```



```

        ++nc;
    }
    f.close();
    cout<< nc <<" "<< ns;
    system("PAUSE");
    return 0;
}

```

23. Имеется текстовый файл и две символьные строки. Все вхождения первой строки в файл заменить второй строкой. Каждая символьная строка является отдельной строкой файла. Имя файла и обе строки задаются как аргументы командной строки ОС *Windows*.

```

#include <iostream>
#include <fstream>
#include <string>
#include<windows.h>
using namespace std;
int main(int argc, char * argv[])
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    string f_name=argv[1];
    string str1= argv[1];
    string str2= argv[2];
    ifstream f (f_name);
    ofstream temp ("D://temp/temp.txt");
    if (!f.is_open())
    {
        cout<<"Не удалось открыть файл\"D://temp/temp.txt\"";
        return 1;
    }
    while (getline(f,s))
    {
        if (s == str1)
            temp<< str2;
        else
            temp<<s;
        temp<<'\n';
    }
    remove(f_name.c_str());
    rename ("D://temp/temp.txt", f_name.c_str());
    f.close();
    temp.close();
}

```

```

system("PAUSE");
return 0;
}

```

12.2. Тренировочные задания

1. Требуется создать файл целых чисел, записав в него числа Фибоначчи, не превышающие заданного числа N . Число N вводится с клавиатуры.

2. Имеются два файла целых чисел. Требуется определить, в каком из них больше а) положительных; б) отрицательных; в) нулевых значений.

3. Имеется файл целых чисел. Требуется определить: а) упорядочены ли значения в файле по возрастанию; б) образуют ли значения монотонную последовательность.

4. Имеется файл целых чисел. Требуется определить, является ли последовательность чисел в файле знакопеременной.

5. Имеется файл целых чисел. Требуется найти наибольшее из чисел в файле, принадлежащее интервалу (a, b) . Значения a и b вводятся с клавиатуры.

6. Имеется файл целых чисел. Требуется создать новый файл целых чисел по следующему правилу: сумма подряд идущих значений одного знака из первого файла становится очередным значением второго файла, вместо группы нулей записывается один нуль. Например:

– исходный файл: 20 10 4 -1 -2 0 0 0 -10 41 62

– файл-результат: 34 -3 0 -10 103

7. Имеется файл целых чисел. Требуется сформировать два файла целых чисел, записывая поочередно подряд идущие числа исходного файла, образующие возрастающую последовательность, то в один, то в другой файл. Например:

– исходный файл: 2 4 3 8 15 -1 -3 10 1

– первый файл-результат: 2 4 -1 1

– второй файл-результат: 3 8 15 -3 10

8. По файлам, полученным в результате выполнения задания 9, требуется восстановить исходный файл целых чисел.

9. Имеется файл целых чисел. Требуется создать файл вещественных чисел по следующему правилу: очередное число нового файла есть среднее арифметическое каждой пятерки подряд идущих чисел исходного файла. Последняя группа чисел исходного файла может содержать меньше пяти значений.

10. Имеются два файла целых чисел, в каждом из которых числа упорядочены по возрастанию. Требуется выполнить слияние этих файлов, то есть создать третий файл, объединив в нем значения из исходных файлов и сохранив упорядоченность значений.

11. Имеется файл целых чисел. Требуется определить длину наибольшего интервала возрастания подряд идущих чисел файла и среднее арифметическое чисел этого интервала.

12. Имеется файл целых чисел. Требуется определить, образуют ли числа файла арифметическую прогрессию.

13. Имеются два файла целых чисел. Требуется проверить, совпадают ли последовательности чисел, расположенных в этих файлах.

14. Имеется файл целых чисел, принадлежащих отрезку $[a, b]$. Требуется определить, сколько раз каждое из этих чисел встречается в файле.

15. Имеется файл целых чисел, образованный из строк квадратной целочисленной матрицы размером 100×100 . Требуется определить, является ли матрица единичной. (В единичной матрице каждый элемент главной диагонали равен 1, остальные элементы равны нулю.)

16. Пусть файл содержит целые числа, образующие строки квадратной целочисленной матрицы размером 100×100 . Требуется определить значение первого (при просмотре матрицы по строкам сверху вниз) максимального элемента матрицы, а также номер строки и номер столбца, на пересечении которых расположен этот элемент.

17. Имеется файл целых чисел, образованный из строк квадратной целочисленной матрицы размером 100×100 . Требуется определить, упорядочены ли элементы m -го столбца матрицы по возрастанию. Число m задается вводом с клавиатуры.

18. Пусть файл содержит целые числа, образующие строки квадратной целочисленной матрицы размером 100×100 . Требуется создать новый файл целых чисел из элементов m -го столбца исходной матрицы. Число m задается вводом с клавиатуры.

19. Студентам 3 курса университета была предложена анкета, в которой им предлагалось указать, знание каких из изученных в университете курсов могут потребоваться в будущей работе. Результаты анкетирования были занесены в файл (см. примеры 15, 16 из предыдущего раздела). Список изученных курсов:

- *математический анализ*
- *алгебра*
- *информатика*
- *численные методы*
- *базы данных*
- *теория вероятностей*
- *web-программирование.*

По задаваемому вводом с клавиатуры предмету требуется определить, сколько студентов назвали его полезным.

20. Студентам 4 курса университета была предложена анкета, в которой требовалось указать, знание каких из изученных в университете кур-

сов могут потребоваться в будущей работе. Результаты анкетирования были занесены в файл (см. примеры 15, 16 из предыдущего раздела). Требуется определить и вывести на экран монитора название предмета, получившего наибольшую оценку. Если таких предметов несколько – вывести названия всех.

21. Студентам 4 курса университета была предложена анкета, в которой требовалось указать, знание каких из изученных в университете курсов могут потребоваться в будущей работе. Результаты анкетирования были занесены в файл (см. примеры 15, 16 из предыдущего раздела). Требуется определить:

а) сколько студентов не оценили положительно ни один из предметов из списка;

б) сколько студентов назвали полезными все предметы из списка;

в) сколько студентов назвали полезными k и более предметов (k задается вводом с клавиатуры).

22. Студентам 3 курса университета была предложена анкета, в которой требовалось указать, знание каких из изученных в университете курсов могут потребоваться в будущей работе. Результаты анкетирования были занесены в файл (см. примеры 15, 16 из предыдущего раздела). Требуется создать файл с названиями предметов, записав их в файл в порядке убывания их рейтинга.

23. Имеется файл, содержащий сведения об абонентах телефонной сети. Каждая строка файла содержит: фамилию, имя и отчество абонента (ФИО), номер телефона, состоящий из двух частей – номера АТС и собственно номера телефона. Строки в файле упорядочены по номеру телефона.

а) требуется добавить в файл новую запись об абоненте, сохранив упорядоченность по возрастанию номеров телефонов;

б) требуется отсортировать записи в файле по ключевому полю ФИО. Реализуйте алгоритм внешней сортировки простым слиянием, используя один входной и два выходных файла: фаза разбиения – серии длиной L из входного файла поочередно записываются в выходные файлы, фаза слияния – выходные файлы становятся входными, входной – выходным, серии длиной L сливаются в серии длиной $2L$. Процесс продолжается до тех пор, пока в результате фазы слияния не получится файл, состоящий из одной серии. (Серия – подпоследовательность упорядоченных, подряд идущих элементов файла).

24. Требуется создать файл *wages*, каждая строка которого будет содержать сведения о сотруднике предприятия: номер отдела, фамилия, имя, отчество (ФИО), размер зарплаты. Строки следует упорядочить по полю номер отдела.

25. Имеется файл *wages*, каждая строка которого содержит сведения о сотруднике предприятия: номер отдела, фамилия, имя, отчество (ФИО),

размер зарплаты. Строки упорядочены по полю номер отдела. Требуется распечатать ведомости для выдачи зарплаты по отделам.

26. Имеется файл *wages*, каждая строка которого содержит сведения о сотруднике предприятия: номер отдела, фамилия, имя, отчество (ФИО), размер зарплаты. Строки упорядочены по полю номер отдела. Требуется вывести на экран номер отдела с самой высокой средней зарплатой. Если таких отделов несколько, вывести все их номера.

27. Имеется текстовый файл *in.txt*. Требуется создать новый файл, перенеся в него те строки исходного файла, в которых содержится заданное слово. Слово и имя нового файла вводятся с клавиатуры.

28. Имеется текстовый файл *in.txt*. Требуется создать новый файл, перенеся в него те строки исходного файла, которые не содержат заданного слова. Слово и имя нового файла вводятся с клавиатуры.

29. Имеется текстовый файл *in.txt*, не содержащий пустых строк. Требуется каждое слово из этого файла записать в отдельную строку нового файла, имя которого задается вводом с клавиатуры. Символ новой строки в исходном файле соответствует пустой строке в новом файле.

30. По файлу, полученному по условию задания 29, требуется создать первоначальный файл, в котором слова разделены одним пробелом.

31. Имеется текстовый файл *in.txt*. Требуется вывести на экран слово наибольшей длины из каждой строки файла. Если таких слов в строке несколько, выведите любые два из них.

32. Имеется текстовый файл *in.txt*. Требуется вывести на экран строку с заданным порядковым номером, который вводится с клавиатуры.

33. Имеется текстовый файл *in.txt*. Требуется вывести на экран строки файла, начинающиеся с заданного слова. Слово задается вводом с клавиатуры.

34. Имеется текстовый файл *in.txt*. Требуется удалить из файла строку с заданным вводом с клавиатуры номером. Если строки с указанным номером в файле нет, то вывести сообщение об этом на экран.

35. Имеется текстовый файл *in.txt*. Если в нем есть подряд идущие строки, отличающиеся между собой только суммарным числом пробелов, то следует оставить в файле только первую из строк, содержащих минимальное число пробелов.

36. Имеется текстовый файл *in.txt*. Требуется создать два новых файла: в первый записать первые p слов каждой строки исходного файла, во второй – оставшиеся слова. Если в строке меньше p слов, то все слова строки переносятся в первый файл, а во втором файле соответствующая строка должна быть пустой.

37. Имеется текстовый файл *in.txt*. Требуется вывести на экран строки файла, которые непосредственно предшествуют строкам, начинающимся с заданного слова, задаваемого вводом с клавиатуры.

38. Имеется текстовый файл *in.txt*. Требуется вывести на экран первое из слов файла, которое содержит наибольшее число различных букв.

39. Имеется текстовый файл *in.txt*. Требуется вывести на экран первое из слов каждой строки файла, которое содержит наибольшее число различных букв.

40. Имеется текстовый файл *in.txt*. Требуется для каждой строки файла вывести на экран слово с порядковым номером в строке, равным целой части выражения $(n + 1) / 2$, где n – количество слов в строке.

41. Имеется текстовый файл *in.txt*. Требуется создать файл целых чисел, в каждой строке которого будет содержаться количество слов в соответствующей строке исходного файла. Пустой строке исходного файла или строке, состоящей из одних пробелов, соответствует число нуль в новом файле.

42. Имеется текстовый файл *in.txt*. Известно, что максимальная длина слова в этом файле равна пяти. Требуется создать пять новых файлов. В каждую строку первого файла заносятся слова соответствующей строки исходного файла, состоящие из одного символа, в каждую строку второго файла – слова соответствующей строки исходного файла, состоящие из двух символов, и т. д. до слов, содержащих пять символов. Если слов длины k в строке исходного файла нет, то соответствующая строка k -го файла должна быть пустой.

43. Имеются два текстовых файла *in1.txt*, *in2.txt*. Требуется проверить их на равенство. Равенство понимается в смысле совпадения соответствующих символов и одинакового разделения на строки.

44. Имеется файл *in.txt*, содержащий сведения о владельцах телефонов. Требуется дописать в файл данные еще об одном владельце. Если файла на диске нет, то необходимо создать его. В файле каждый элемент данных (фамилия, имя, отчество, номер телефона) должен находиться в отдельной строке. Рекомендуемый вид диалога с пользователем:

ДОБАВЛЕНИЕ В ТЕЛЕФОННЫЙ СПРАВОЧНИК

Фамилия - > Иванов

Имя - > Иван

Отчество - > Иванович

Телефон - >200-99-00

Информация добавлена.

Для завершения работы нажмите <Enter >

45. Имеется файл, содержащий сведения о владельцах телефонов. Требуется по фамилии владельца определить номер телефона. Для завершения работы сразу после запроса фамилии нажмите <Enter>. Рекомендуемый вид диалога с пользователем:

ПОИСК В ТЕЛЕФОННОМ СПРАВОЧНИКЕ

Введите фамилию -> Петров

В справочнике данных о Петров нет.
Введите фамилию -> Иванов
Иванов Василий Васильевич 578-12-45
Иванов Иван Иванович 200-99-00
Введите фамилию ->

46. Имеется файл, содержащий сведения о владельцах телефонов. Требуется создать программу, объединяющую возможности, описанные в заданиях 44 и 45. Рекомендуемый вид начального диалога с пользователем:

**** *ТЕЛЕФОННЫЙ СПРАВОЧНИК*****

1. Добавление
2. Поиск
3. Завершение работы

Введите номер пункта меню - >

47. Требуется создать файл с возможностью прямого доступа, содержащий результаты экзаменационной сессии на трех факультетах: ПММ, физическом и математическом. Каждый элемент файла представлен структурой, описанной в примере 17 предыдущего раздела. Названия факультетов, номера групп и курсов в структуру не включайте. Количество курсов и групп на курсах фиксировано.

48. Имеется файл с возможностью прямого доступа, содержащий результаты экзаменационной сессии на трех факультетах: ПММ, физическом и математическом. Каждый элемент файла представлен структурой, описанной в примере 17 предыдущего раздела. Требуется, используя механизмы прямого доступа, по заданному номеру факультета (ПММ – 1, физический – 2, математический – 3), номеру курса и номеру группы вывести на экран информацию о результатах экзаменационной сессии в выбранной группе. Номера факультета, курса и группы вводятся с клавиатуры.

49. Имеется файл с возможностью прямого доступа, содержащий результаты экзаменационной сессии на трех факультетах: ПММ, физическом и математическом. Каждый элемент файла представлен структурой, описанной в примере 17 предыдущего раздела. Требуется, используя механизмы прямого доступа, по заданному номеру факультета вывести на экран номер группы с наименьшим числом неуспевающих. Если таких групп несколько, вывести номера всех их.

50. Имеется файл с возможностью прямого доступа, содержащий результаты экзаменационной сессии на трех факультетах: ПММ, физическом и математическом. Каждый элемент файла представлен структурой, описанной в примере 17 предыдущего раздела. Требуется, используя механизмы прямого доступа, по заданным номерам факультетов и номеру курса вывести на экран сведения о количестве отличников и неуспевающих на курсе.

51. Имеются три файла. Первый (*f_fam*) содержит фамилии студентов, каждая фамилия – строка файла. Вторым (*f_mark*) – оценки (в цифровом виде) студентов по итогам сессии, строка файла – набор оценок одного студента через пробел. Третий файл (*f_change*) содержит корректировку оценок, в каждой строке файла указаны номер студента, совпадающий с порядковым номером фамилии в первом файле, порядковый номер изменяемой оценки (одной!) и новая оценка. Строки упорядочены по номерам студентов. Требуется вывести на экран список фамилий студентов, сдавших сессию на «хорошо» (4) и «отлично» (5). Чтение из файлов осуществляется один раз.

Пример.

Файл *f_fam*: Иванов
Петров
Сидоров

Файл *f_mark*: 4 5 4 5 5
3 3 3 2 4
4 4 3 4 5

Файл *f_change*: 3 3 4

Вывод: Иванов
Сидоров

52. Имеются три файла. Первый (*f_fam*) содержит фамилии студентов, каждая фамилия – строка файла. Вторым (*f_mark*) – оценки (в цифровом виде) студентов по итогам сессии, строка файла – набор оценок одного студента через пробел. Третий файл (*f_change*) содержит корректировку оценок, в каждой строке файла указаны номер студента, совпадающий с порядковым номером фамилии в первом файле, порядковый номер изменяемой оценки и новая оценка. Строки упорядочены по номерам студентов. Требуется по заданной фамилии студента вывести на экран его оценки с учетом возможной корректировки.

53. Требуется создать текстовый файл, содержащий сведения о сотрудниках предприятия:

фамилия, имя, отчество;

дата рождения: число, месяц, год;

домашний адрес: название улицы, номер дома, номер квартиры.

Требуется после создания файла вывести на экран список сотрудников, у которых день рождения в текущем месяце. Предусмотреть два варианта задания месяца: а) название месяца задается вводом; б) месяц определяется через системную дату.

54. Имеется файл, созданный в результате выполнения задания 53. Требуется вывести на экран список сотрудников, проживающих на заданной улице. Название улицы вводится с клавиатуры.

55. Имеется файл, созданный в результате выполнения задания 53. Требуется вывести на экран список улиц, на которых проживают сотрудники, причем в списке не должно быть повторяющихся названий.

56. Имеется файл, созданный в результате выполнения задания 53. Требуется вывести на экран фамилии самого старшего и самого молодого сотрудника. Если таких сотрудников несколько, то вывести фамилии всех их.

57. Имеется текстовый файл. Требуется удалить из каждой строки файла все лишние пробелы: пробелы в начале каждой строки, пробелы в конце строки и оставить между словами по одному пробелу.

58. Имеется текстовый файл, первая строка которого содержит целое число N , последующие строки содержат некоторый текст. Требуется записанный в файле текст разбить на строки, длина которых не превосходит N , и записать их в новый файл. В каждой строке не должно быть пробелов в начале строки, пробелов в конце строки, и слова в строке должны быть разделены одним пробелом. При этом каждая строка должна быть максимально длинной, то есть строки формируются по принципу: добавляем в строку слова из исходного файла до тех пор, пока длина полученной строки не превышает N , после этого ставим маркер новой строки. Известно, что во входном файле нет слов, содержащих более N символов.

59. Имеется файл целых чисел *in.txt*. Требуется создать новый файл из тех троек чисел исходного файла, которые могут быть длинами сторон прямоугольного треугольника. Имя нового файла задается вводом с клавиатуры.

60. Имеется файл целых чисел *in.txt*, в котором записаны координаты N точек плоскости $x_1, y_1, x_2, y_2, \dots, x_N, y_N$. Требуется создать новый файл из координат точек плоскости, попадающих внутрь кольца с центром в начале координат. Внутренний радиус кольца равен R_1 , внешний R_2 . Имя нового файла, значения R_1 и R_2 задаются вводом с клавиатуры.

61. Пусть строки целочисленной квадратной матрицы размером 20×20 записаны в файл *in.txt*. Требуется найти скалярное произведение элементов первой строки матрицы, в которой расположен максимальный элемент матрицы, и столбца, в котором расположен последний из минимальных элементов этой матрицы.

62. Имеется три файла вещественных чисел *in1.txt*, *in2.txt* и *in3.txt*, в каждом из которых расположены элементы вещественных матриц. Каждая матрица содержит m строк, в каждой из которых n чисел. Перенести в новый файл *out.txt* элементы любой из тех матриц, норма которых минимальна. Норма матрицы вычисляется по формуле

$$\max_i \sum_j |a_{ij}|$$

(максимальное значение из суммы модулей каждой строки).

63. Имеется файл *in.txt* из структур, каждая из которых содержит координаты точки трехмерного пространства. Найти количество пар точек, расстояние между которыми максимально. Расстояние между точками $A(x_1, y_1, z_1)$ и $B(x_2, y_2, z_2)$ вычисляется по формуле

$$|AB| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}.$$

64. Пусть имеются десять файлов целых чисел, в каждом из которых расположены элементы матриц. Каждая матрица содержит m строк, в каждой из которых n чисел. Найти произведение любых двух матриц, одна из которых имеет максимальную норму (см. задание 62), другая – минимальную. Элементы полученной матрицы требуется поместить в новый файл. Имена всех файлов задаются вводом с клавиатуры.

12.3. Задания для самостоятельной работы

12.3.1. Описание структур данных (элементов файлов)

Указанное в описании структур число знаков является рекомендуемым, но не обязательным. Выполнение каждого задания должно начинаться с этапа создания файла в двух вариантах: текстовом и бинарном. Поэтому решение сформулированной в условии задачи также должно быть выполнено в двух вариантах.

Файл *student.dat*

- фамилия, имя, отчество (40 символов);
- номер курса (1 символ);
- номер группы (1 символ);
- итоги сессий: 10 наборов из 5 оценок каждый;
- форма обучения: очная, очно-заочная (1 символ);
- экзамен: наименование предмета (10 символов) и оценка (1 символ);

Данные в файле упорядочены по номеру курса, внутри курса – по номеру группы, в группе – по фамилии.

Файл *telephone.dat*

- фамилия, имя, отчество (40 символов);
- домашний адрес: улица (20 символов), номер дома (3 символа), номер квартиры (3 символа);
- номер телефона (7 символов).

Данные в файле упорядочены по фамилии.

Файл *catalog.dat*

- имя файла (8 символов);
- расширение имени (3 символа);

- дата создания файла: день (2 символа), месяц (2 символа), год (4 символа);

- размер файла (9 символов).

Данные в файле упорядочены по именам файлов.

Файл *books.dat*

- автор книги (40 символов);
- название книги (80 символов);
- год издания (4 символа);
- издательство (40 символов).

Данные в файле упорядочены по фамилиям авторов.

Файл *airplane.dat*

- номер рейса (4 символа);
- пункт назначения (40 символов);
- дата вылета (6 символов);
- время вылета (4 символа);
- время прибытия (4 символа);
- количество свободных мест в салоне (2 символа).

Данные в файле упорядочены по дате вылета.

Файл *movie.dat*

- название (80 символов);
- режиссер (40 символов);
- страна (20 символов);
- количество просмотров (9 символов);
- год выпуска (4 символа).

Данные в файле упорядочены по названию фильма.

12.3.2. Варианты заданий

1. По данным, имеющимся в файле *student.dat*, требуется определить и вывести на экран количество студентов, обучающихся на каждом курсе и в каждой группе. Результат представить в виде таблицы.

2. Требуется занести в отдельный файл данные из файла *student.dat*, добавив оценки за последнюю сессию в группе, номер которой и номер курса задаются вводом с клавиатуры.

3. Требуется занести в отдельный файл данные из файла *student.dat*, добавив сведения о новых студентах. Новые данные задаются вводом с клавиатуры.

4. Требуется занести в отдельный файл данные из файла *student.dat*, исключив сведения об отчисленных студентах. Студент считается отчисленным, если у него имеется хотя бы одна неудовлетворительная (2) оценка.

5. Требуется занести в отдельный файл данные об отличниках из файла *student.dat*.

6. По данным, имеющимся в файле *student.dat*, требуется определить и вывести на экран средний балл по заданному предмету на заданном курсе. Данные вводятся с клавиатуры. Результат представить в виде таблицы.

7. По данным, имеющимся в файле *student.dat*, требуется определить и вывести на экран номера групп на каждом курсе, в которых больше, чем в других группах, отличников. Результат представить в виде таблицы.

8. По данным, имеющимся в файле *student.dat*, требуется определить и вывести на экран средний балл в каждой группе на заданном курсе и средний балл по курсу. Номер курса вводится с клавиатуры. Результат представить в виде таблицы.

9. Требуется занести в отдельный файл данные из файла *student.dat* о студентах, имеющих не более одной четверки за последнюю экзаменационную сессию.

10. Требуется занести в отдельный файл данные из файла *student.dat* о студентах 4 курса, упорядочив их по убыванию сумм баллов за все время обучения.

11. Требуется занести в отдельный файл данные из файла *student.dat*, выполнив перевод каждого студента на следующий курс, удалив данные о студентах 4 курса.

12. Требуется занести в отдельный файл данные из файла *books.dat* данные о произведениях заданного автора. Данные вводятся с клавиатуры.

13. По данным, имеющимся в файле *books.dat*, требуется занести в отдельный файл сведения о произведениях, изданных в заданном году. Данные вводятся с клавиатуры.

14. По данным, имеющимся в файле *books.dat*, требуется определить название издательства, в котором издано наибольшее количество книг. Число издательств не более 100.

15. По данным, имеющимся в файле *catalog.dat*, требуется занести в отдельный файл сведения о файлах, размер которых превышает заданное число байтов. Данные вводятся с клавиатуры.

16. Данные, имеющиеся в файле *catalog.dat*, требуется переписать в новый файл, упорядочив данные по размеру файла (в порядке убывания).

17. По данным, имеющимся в файле *catalog.dat*, требуется занести в отдельный файл сведения о файлах с заданным расширением имени. Данные вводятся с клавиатуры.

18. По данным, имеющимся в файле *catalog.dat*, требуется занести в отдельный файл сведения о файлах, созданных ранее указанной даты. Данные вводятся с клавиатуры.

19. По данным, имеющимся в файле *catalog.dat*, требуется вывести на экран сведения о файлах, имеющих расширения имен *.cpp*, *.dat*, *.exe*. Данные вводятся с клавиатуры. Результат представить в виде таблицы.

20. По данным, имеющимся в файле *telephone.dat*, требуется занести в отдельный файл сведения об абонентах, проживающих по заданному адресу (улица, дом, квартира). Данные вводятся с клавиатуры.

21. По данным, имеющимся в файле *telephone.dat*, требуется занести в отдельный файл сведения об абонентах, номера телефонов которых начинаются с заданных трех цифр. Данные вводятся с клавиатуры.

22. По данным, имеющимся в файле *telephone.dat*, требуется по заданным фамилии, имени и отчеству вывести на экран соответствующие номера телефонов и адреса абонентов. Данные вводятся с клавиатуры.

23. По данным, имеющимся в файле *telephone.dat*, требуется по заданному телефонному номеру определить фамилию, имя, отчество и адрес абонента и вывести эти данные на экран. Номер телефона вводится с клавиатуры.

24. По данным, имеющимся в файле *telephone.dat*, требуется найти и вывести на экран номера двух АТС (первые три цифры телефонного номера), имеющих наибольшее число абонентов.

25. По данным, имеющимся в файле *airplane.dat* определить и вывести на экран время вылета самолетов в город X по заданной дате. Название города и дата вводятся с клавиатуры.

26. По данным, имеющимся в файле *airplane.dat* определить и вывести на экран число свободных мест на рейс в город X с временем отправления Y и датой Z. Название города, дата и время вылета вводятся с клавиатуры.

27. По данным, имеющимся в файле *airplane.dat* определить и вывести на экран номер рейса, даты и время вылета самолетов в город X, в которые на рейс имеются свободные места. Название города вводится с клавиатуры.

28. По данным, имеющимся в файле *airplane.dat* требуется создать новый файл, занеся в него данные о вылетах в город X в заданную дату. Название города и дата вылета вводятся с клавиатуры.

29. По данным, имеющимся в файле *airplane.dat* требуется создать новый файл, занеся в него данные о всех вылетах в заданную дату до указанного времени вылета. Дата и время вылета вводятся с клавиатуры.

30. По данным, имеющимся в файле *airplane.dat*, определить и вывести на экран номер первого рейса с датой Y в город X, имеющего свободные места. Название города и дата вылета вводятся с клавиатуры.

31. По данным, имеющимся в файле *movie.dat*, определить и вывести на экран название фильма, который был выпущен в заданном году и имел наибольшее число просмотров. Год выпуска вводится с клавиатуры.

32. По данным, имеющимся в файле *movie.dat*, определить и записать в новый файл данные о фильмах режиссера, выпустившего наибольшее число фильмов.

33. По данным, имеющимся в файле *movie.dat*, определить и записать в новый файл данные о фильмах, выпущенных режиссером, имя которого вводится с клавиатуры, за время, предшествующее заданному году. Год также вводится с клавиатуры.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ускова О.Ф. Задачник-практикум по алгоритмическому языку / О.Ф.Ускова, О.Д.Горбенко. – Воронеж : Издательство Воронежского государственного университета, 1990. – 196 с.
2. Программирование на языке Паскаль : задачник; под ред. Усковой О.Ф. / О.Ф. Ускова, М.В. Бакланов, И.Е. Воронина, О.Д. Горбенко, Г.Э. Вощинская, Н.В. Огаркова, В.М. Мельников. – СПб. : Питер, 2002 (2003, 2005). – 366 с.
3. Ускова О.Ф. Информатика. Начальный курс : учебное пособие для поступающих на факультет ПММ ВГУ / О.Ф. Ускова, О.Д. Горбенко, Н.А. Каплиева. – Воронеж: Альбион, 2006.– 183 с.
4. Гуденко Д.А. Сборник задач по программированию / Д.А. Гуденко, Д.В. Петроченко. – СПб. : Питер, 2003. – 475 с.
5. Юркин А.В. Задачник по программированию / А.В. Юркин. – СПб. : Питер, 2010.–192 с.
6. Ускова О.Ф. Основы программирования : учебное пособие / О.Ф. Ускова, Н.А. Каплиева. – Воронеж: Издательско-полиграфический центр ВГУ, 2010. – 266 с. (Учебник Воронежского государственного университета).
7. Павловская Т.А. С/С++. Программирование на языке высокого уровня. Структурное программирование. Практикум / Т.А. Павловская, Ю.А. Щупак. – СПб. : Питер, 2008.– 240 с.
8. Ускова О.Ф. Перегрузка функций в языке С++ : электронный ресурс / О.Ф. Ускова, Н.А. Каплиева, Н.Б. Ускова // Информатика : проблемы, методология, технологии : сборник материалов 17-й международной научно-методической конференции, Воронеж 9-10 февраля 2017 г. : в 5 т. – Воронеж, 2017. – Т. 5. – С. 117–121.
9. Шилдт Г. С++: базовый курс, 3-е издание: пер. с англ. / Г. Шилдт. – М. : Издательский дом «Вильямс», 2010. – 624 с.
10. Языки программирования. Версия 1.0 [Электронный ресурс] : метод. указания по лабораторным работам / сост. Ю.А. Шитов, Е.А. Новиков, С.Н. Титовский и др. – Электрон. дан. (1 Мб). – Красноярск : ИПК СФУ, 2008.
11. Прата С. Язык программирования С++. Лекции и упражнения. Учебник: пер. с англ. / С. Прата. – СПб. : ООО «ДиаСофтЮП», 2005. – 1104 с.
12. Культин Н.Б. С/С++ в задачах и примерах / Н.Б. Культин. – СПб. : БХВ-Петербург, 2005. – 288 с.
13. Франка П. С++: учебный курс / П. Франка. – СПб. : Питер, 2003. – 521с.

14. Динман М.И. С++. Освой на примерах / М.И. Динман. – СПб. : БХВ-Петербург, 2006. – 384 с.
15. Хомоненко А.Д. Программирование на С++: Учебное пособие / А.Д. Хомоненко. – Альтекс, 2003. – 512 с.
16. <http://docplayer.ru/31371111-Dvoichnye-i-tekstovye-fayly-reshenie-zadach.html> 17.07.2018
17. <http://cppstudio.com/post/319/> 28.07.2018
18. Ускова О.Ф. Начала структурного программирования на языке С++. Часть 1. Линейные, разветвляющиеся и циклические алгоритмы : задачник-практикум / О.Ф. Ускова, О.Д. Горбенко. – Воронеж : Издательско-полиграфический центр ВГУ, 2013. – 32 с.
19. Ускова О.Ф. Начала структурного программирования на языке С++. Часть 2. Одномерные массивы : задачник-практикум / О.Ф. Ускова, О.Д. Горбенко. – Воронеж : ООО ИПЦ «Научная книга», 2015. – 32 с.
20. Ускова О.Ф. Начала структурного программирования на языке С++. Часть 3. Двумерные массивы : задачник-практикум / О.Ф. Ускова, Н.А. Каплиева. – Воронеж : ООО ИПЦ «Научная книга», 2015. – 37 с.
21. Ускова О.Ф. Начала структурного программирования на языке С++. Часть 4. Символы и символьные строки : задачник-практикум / О.Ф. Ускова, Н.А. Каплиева. – Воронеж : Издательский дом ВГУ, 2016. – 44 с.
22. Ускова О.Ф. Начала структурного программирования на языке С++ : задачник-практикум. Часть 5. Структуры и объединения / О.Ф. Ускова, Н.А. Каплиева. – Воронеж : Издательский дом ВГУ, 2017. – 44 с.
23. Ускова О.Ф. Начала структурного программирования на языке С++. Часть 6. Функции : задачник-практикум / О.Ф. Ускова, Н.А. Каплиева. – Воронеж : ООО ИПЦ «Научная книга», 2018. – 51 с.
24. Ускова О.Ф. Начала структурного программирования на языке С++. Часть 7. Файловый ввод/вывод : задачник-практикум / О.Ф. Ускова, О.Д. Горбенко. – Воронеж : Издательский дом ВГУ, 2019. – 69 с.
25. Павловская Т.А. С/С++. Программирование на языке высокого уровня. Структурное программирование. Практикум / Т.А. Павловская, Ю.А. Щупак. – СПб. : Питер, 2008. – 240 с.
26. Подбельский В.В. Язык С++ / В.В. Подбельский. – М. : Финансы и статистика, 1999. – 559 с.
27. Ускова О.Ф. Методическая поддержка практических занятий по программированию на языке С++ / О.Ф. Ускова, Н.А. Каплиева, О.Д. Горбенко // Актуальные проблемы прикладной математики, информатики и механики [Электронный ресурс] : сборник трудов Международной научной конференции, Воронеж, 17–18 декабря 2018 г. – Воронеж, 2019. – С. 1419–1422.

28. Ускова О.Ф. Расширение возможностей функций в языке программирования С++ по сравнению с языком программирования Паскаль / О.Ф. Ускова, Н.А. Каплиева // Информатика : проблемы, методология, технологии : материалы 19-й международной научно-методической конференции, Воронеж, 14–15 февраля 2019 г. – Воронеж, 2019. – С. 1827–1829.

ОГЛАВЛЕНИЕ

Глава 1. ЛИНЕЙНЫЕ АЛГОРИТМЫ.....	3
1.1. Задачи с решениями.....	3
1.2. Варианты заданий для самостоятельной работы.....	3
1.3. Решения.....	5
1.4. Указания к решению задач	6
Глава 2. АЛГОРИТМЫ С ВЕТВЛЕНИЯМИ	8
2.1. Задачи с решениями.....	8
2.2. Варианты заданий для самостоятельной работы.....	9
2.2.1. Задание 1	9
2.2.2. Задание 2	11
2.2.3. Задание 3	13
2.3. Решения.....	15
2.4. Указания к решению задач	19
Глава 3. ЦИКЛИЧЕСКИЕ АЛГОРИТМЫ.....	20
3.1. Задачи с решениями.....	20
3.2. Варианты заданий для самостоятельной работы.....	20
3.2.1 Задание 1	20
3.2.2 Задание 2	27
3.3. Решения.....	28
3.4. Указания к решению задач	31
Глава 4. СТАТИЧЕСКИЕ ОДНОМЕРНЫЕ МАССИВЫ	32
4.1. Условия заданий с решениями	32
4.2. Решения.....	32
Глава 5. ИСПОЛЬЗОВАНИЕ АДРЕСНОЙ АРИФМЕТИКИ ПРИ ОБРАБОТКЕ ОДНОМЕРНЫХ МАССИВОВ	40
5.1. Условия заданий с решениями	40
5.2. Решения.....	41
Глава 6. ДИНАМИЧЕСКИЕ ОДНОМЕРНЫЕ МАССИВЫ.....	45
6.1. Условия задач с решениями.....	45
6.2. Решения.....	45
6.3. Тренировочные задания	48
6.4. Задания для самостоятельной работы.....	53
6.4.1. Задание № 1	53
6.4.2. Задание № 2	57
Глава 7. ДВУМЕРНЫЕ МАССИВЫ.....	63
7.1. Задания с решениями.....	63
7.1.1. Статические двумерные массивы	63
7.1.2. Двумерные массивы и указатели	68

7.1.3. Динамические двумерные массивы	71
7.2. Тренировочные задания	74
7.3. Задания для самостоятельной работы.....	82
7.3.1. Задание 1	82
7.3.2 Задание 2.....	87
7.3.3. Задание 3	90
Глава 8. СИМВОЛЫ И СИМВОЛЬНЫЕ СТРОКИ.....	93
8.1. Условия заданий с решениями	93
8.2. Решения.....	95
8.3. Тренировочные задания	109
8.4. Задания для самостоятельной работы.....	116
8.5. Библиотечные функции для работы с символами и символьными строками.....	124
Глава 9. СТРУКТУРЫ	128
9.1. Вопросы и задания с решениями.....	128
9.1.1. Условия заданий	128
9.1.2. Ответы и решения.....	133
9.2. Задания с решениями на составление программ	135
9.2.1. Условия заданий	135
9.2.2. Решения.....	136
9.3. Тренировочные задания	140
9.4. Задания для самостоятельной работы на компьютере.....	146
Глава 10 ОБЪЕДИНЕНИЯ	158
10.1. Вопросы, задачи, программы.....	158
10.2. Ответы и решения	162
Глава 11. ФУНКЦИИ.....	167
11.1. Контрольные вопросы, задания, упражнения	167
11.1.1. Тексты заданий	167
11.1.2. Ответы и решения.....	173
11.2. Примеры разработанных программ	176
11.2.1. Условия заданий	176
11.2.2. Ответы и решения.....	185
11.3. Задания для самостоятельной работы.....	201
11.4. Библиотечные математические функции	210
Глава 12. ФАЙЛОВЫЙ ВВОД / ВЫВОД ДАННЫХ.....	212
12.1. Работа с файлами	212
12.1.1. Некоторые справочные сведения.....	212
12.1.2. Вопросы для самопроверки	215
12.1.3. Упражнения и задпчи с решениями.....	217
12.2. Тренировочные задания	242

12.3. Задания для самостоятельной работы.....	250
12.3.1. Описание структур данных (элементов файлов).....	250
12.3.2. Варианты заданий	251
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	255

У ч е б н о е и з д а н и е

Ускова Ольга Федоровна,
Каплиева Наталья Алексеевна,
Горбенко Олег Данилович

НАЧАЛА СТРУКТУРНОГО ПРОГРАММИРОВАНИЯ
НА ЯЗЫКЕ C++

Задачник-практикум

Издано в авторской редакции

Подписано в печать 03.07.2019. Формат 60×84/16
Уч.-изд. л. 16,0. Усл. печ. л. 15,2. Тираж 30 экз. Заказ 450

Издательский дом ВГУ
394018 Воронеж, пл. им. Ленина, 10

Отпечатано с готового оригинал-макета
в типографии Издательского дома ВГУ
394018 Воронеж, ул. Пушкинская, 3