МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

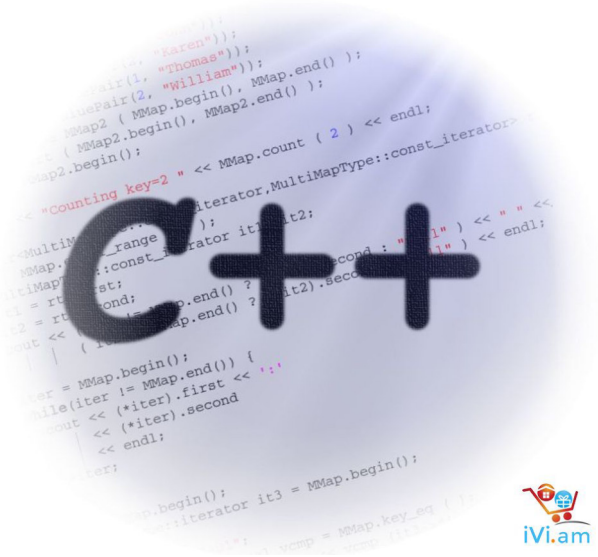
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

Кафедра автоматизированных и вычислительных систем

ПРОГРАММИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ № 1-6
по дисциплине «Основы программирования и алгоритмизации»
для студентов направления 38.03.05
«Бизнес-информатика”
(профиль «Информационные системы в бизнесе»)
очной и заочной форм обучения



Воронеж 2022

УДК 681.3.06
ББК 32.973

Составители:
канд. техн. наук Н.И. Гребенникова,
канд. техн. наук М.Ю. Сергеев,
канд. техн. наук Т.И. Сергеева

Программирование вычислительных задач: к выполнению лабораторных работ № 1-6 по дисциплине «Основы программирования и алгоритмизации» для студентов направления 38.03.05 «Бизнес-информатика” (профиль «Информационные системы в бизнесе») очной и заочной форм обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост.: Н.И. Гребенникова, М. Ю. Сергеев. Т. И. Сергеева, Воронеж: Изд-во ВГТУ, 2022. 32 с.

Цель методических указаний - получение знаний об основах алгоритмизации вычислительных задач, о базовых средствах и конструкциях языка С++, работе с массивами, матрицами, строками и структурами.

Методические указания содержат теоретические сведения и практические задания для проведения лабораторных работ.

Предназначены для проведения лабораторных работ по дисциплине «Основы программирования и алгоритмизации» для студентов 1 курса очной и заочной форм обучения.

Методические указания подготовлены в электронном виде и содержатся в файле Методичка_2022_Осн_программ_и_алгорит_БИ.pdf

Табл. 5. Библиогр.: 10 назв.

УДК 681.3.06(07)
ББК 32.973

Рецензент – П. Ю. Гусев, канд. техн. наук, доцент кафедры компьютерных интеллектуальных технологий проектирования ВГТУ

Издается по решению редакционно-издательского совета Воронежского государственного технического университета

При выполнении заданий лабораторных работ необходимо:

- составить структурную схему программы;
- написать и отладить программу в консольном режиме в соответствии с вариантом задания;
- варианты задания приведены в тексте лабораторной работы; номер варианта совпадает с порядковым номером студента в списке;
- проверить работу программы в консольном режиме.

Отчет должен содержать титульный лист, вариант задания, структурную схему алгоритма, текст программы с комментариями, скриншот результатов работы программы.

1. ОСНОВНЫЕ ФОРМЫ И СПОСОБЫ ПРЕДСТАВЛЕНИЯ АЛГОРИТМА

Алгоритм – конечная последовательность предписаний, однозначно определяющая процесс преобразования исходных и промежуточных данных в результат решения задачи.

Существуют следующие формы представления алгоритмов: словесная и графическая.

Словесная форма представления алгоритма используется для описания очень простых задач и на начальных стадиях разработки алгоритма. Словесная форма представления алгоритма имеет ряд недостатков. Для достаточно сложных алгоритмов описание становится слишком громоздким и ненаглядным.

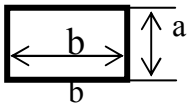
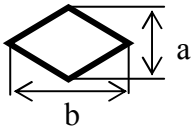
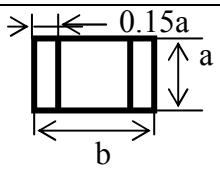
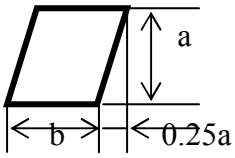
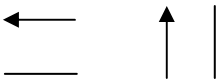
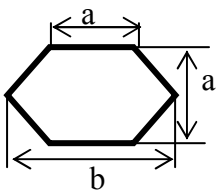
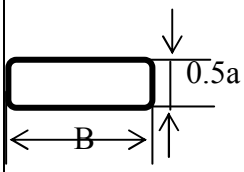
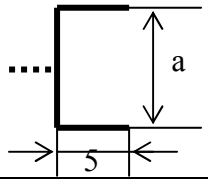
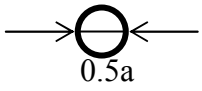
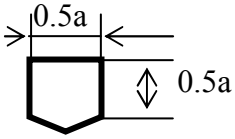
Графическая форма представления алгоритмов является более компактной и наглядной. Алгоритм изображается в виде последовательности связанных между собой блоков, каждый из которых соответствует выполнению одного или несколько операторов. Такое графическое представление называется блок-схемой алгоритма или структурной схемой алгоритма.

Условные графические обозначения символов, используемых для составления блок-схемы алгоритма, стандартизированы. Некоторые, часто используемые обозначения приведены в табл. 1.

Отдельные блоки алгоритмов соединяются между собой линиями. Линии, идущие сверху вниз и слева направо, принимаются за основные и, если не имеют изломов, стрелками не обозначаются.

Таблица 1

Блоки структурной схемы

Процесс (вычисления)		Решение (проверка условия)	
Предопределенный процесс (подпрограмма)		Ввод-вывод	
Соединительные линии		Модификация (начало цикла)	
Начало-конец		Комментарии	
Внутристраничный соединитель		Межстраничный соединитель	

Примечание. Значение a принимается из ряда чисел 10; 15; 20… мм; b = 1,5 a.

2. ЛАБОРАТОРНАЯ РАБОТА № 1. ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ

В программе, реализующей линейную вычислительную задачу, операторы выполняются в той последовательности, в которой они записаны.

В консольном приложении можно вывести информацию на экран и получить данные с клавиатуры несколькими способами:

- при помощи функций printf (вывод) и scanf (ввод);

- вывести информацию в поток вывода (cout), прочитать данные из потока ввода (cin).

В примере, приведенном ниже, использован потоковый ввод-вывод. Каждый элемент вывода начинается с признака вывода в поток <<.

Конструкция endl обеспечивает перевод курсора на следующую строку.

Пример. Составить программу для вычисления функции при произвольных значениях A, X.

$$y(x) = \sqrt{|a * x| + \sin^2 x} - \ln|x + a| + e^{ax}$$

Листинг программы приведен ниже

```
#include "stdafx.h"  // Добавлен в 2010 по умолчанию

#include <iostream>  //Подключение заголовочного файла
                     // с функциями ввода-вывода
using namespace std;   // Определение пространства имен
int _tmain(int argc, _TCHAR* argv[])  // По умолчанию
{
  setlocale(LC_ALL, "Russian");         //Установка //соответствующей
                                        //кодовой страницы
  float a, x, y, y1, y2;                //Объявление переменных
  cout << "Введите a" << endl;  //Вывод фразы
  cin >> a;                             // Ввод a
  cout << "Введите x" << endl;  //Вывод фразы
  cin >> x;                             // Ввод x
  y1 = sqrt(abs(a * x)) + sin(x) * sin(x);//1-я часть формулы
  y2 = -log(abs(x + a)) + exp(x * a);     //2-я часть формулы
  y = y1 + y2;                          //формула целиком
  cout << "Результат = " << y << endl; //Вывод надписи
                                         // и результата (y)
  system("pause");   //Функция ожидания
                     // нажатия клавиши
  return 0;
}
```

В C++ существуют следующие основные операции (табл. 2).

Таблица 2

Основные операции языка C++

Операция	**Краткое описание**
Унарные операции	
++	увеличение на 1
--	уменьшение на 1
!	логическое отрицание
-	арифметическое отрицание (унарный минус)
Бинарные операции	
*	умножение
/	деление
%	остаток от деления
+	сложение
-	вычитание
<	меньше
<=	меньше или равно
>	больше

Окончание табл. 2

Операция	**Краткое описание**
>-	больше или равно
==	Равно
!=	не равно
&&	логическое И
||	логическое ИЛИ
=	присваивание
*=	умножение с присваиванием
/*	деление с присваиванием
%=	остаток отделения с присваиванием
+=	сложение с присваиванием
-=	вычитание с присваиванием

В С++ существуют следующие математические функции, приведенные в табл. 3 (их описание содержится в файле math.h).

Таблица 3

Основные математические функции С++

Функция	Вид	Назначение
abs	int abs(int num)	Возвращает модуль числа
acos	double acos(double x)	Возвращает арккосинус аргумента
asin	double asin(double x)	Возвращает арксинус аргумента
atan	double atan(double x)	Возвращает арктангенс аргумента
cos	double cos(double x)	Возвращает косинус аргумента
cosh	double cosh(double x)	Возвращает гиперболический косинус аргумента
exp	double exp(double x)	Возвращает степень числа e
log	double log (double x)	Возвращает значение натурального логарифма x
log10	double log10(double x)	Возвращает значение логарифма x по основанию 10
pow	double pow(double x, double y)	Функция вычисляет значение числа x в степени y
sin	double sin(double x)	Возвращает синус аргумента
sinh	double sinh(double x)	Возвращает гиперболический синус аргумента
sqrt	double sqrt(double x)	Вычисляет квадратный корень
tan	double tan(double x)	Возвращает значение тангенса аргумента
tanh	double tanh(double x)	Возвращает значение гиперболического тангенса аргумента

Таблица 4

Варианты заданий для реализации линейных вычислений

Номер варианта	Математическое выражение
1	$y = \cos^2(x+a)/\lvert x+a\rvert + \sin(x+a)^2 \cdot (x+a)$
2	$y = (x-a)^2 \cdot \ln\lvert x-a\rvert \cdot \cos(x-a)^2$
3	$y = x \cdot \sqrt{\lvert x-a\rvert} \cdot \cos(x-a)^2 - \sin^2(x-a)$
4	$y = \cos(x-b)/\ln\lvert x-b\rvert - 1/\sin(x-b)$
5	$y = (x+a)^2/\ln\lvert x+a\rvert + 1/\sqrt{\lvert(x+a)\rvert}$
6	$y = (x^2+b)/\sqrt{\lvert x^2-b\rvert} + \sqrt{\lvert x^2-b\rvert}/\cos(x^2+b)$
7	$y = \sqrt{\lvert x-a\rvert}/(x-a) + \sqrt{\lvert x+a\rvert}/(x+a)$
8	$y = (x^2\cdot(x+a) + a^2\cdot\sin(x+a))/\sqrt{\lvert x+a\rvert}$
9	$y = (e^{x+a})/\ln\lvert x+a\rvert + (x+a)^2/\sin^2(x+a)$
10	$y = \ln\lvert x^2-b\rvert \cdot \sqrt{x^2-b} - e^{b-x}$
11	$y = (x+a)/(x-a) + e^{x-a} - (x+a)/\ln\lvert x-a\rvert$
12	$y = (x+b)^2/\sin^2(x+b) + \ln(x+b)$
13	$y = (\sqrt{p\cdot(p-a)\cdot(p-b)}\cdot\ln\lvert p-a\rvert)/(p-b)$
14	$y = ((x+b)^2 + \sin^2(x+b))/e^{x+b}$
15	$y = \sqrt{\lvert(x+a)/(x-a)\rvert}\cdot\sin^2(x-a) + \cos(x+a)^2$
16	$y = (\cos(a\cdot x) + \sqrt{\lvert\sin(a\cdot x)\rvert})/(a\cdot x) + e^{ax}$
17	$y = \sqrt{\ln\lvert x+b\rvert}/(x-b) + (x+b)/e^{x-b}$
18	$y = (\sin(x-c^2+8))/(x-c^2+8) + \ln\lvert x-c^2+8\rvert$
19	$y = (x+b)^2/\sqrt{\lvert x+b\rvert+1} + \ln\lvert x+b\rvert$
20	$y = \sqrt{(x-a)^2}\cdot\ln^2\lvert x-a\rvert + (x-a)/e^{x-a}$
21	$y = ((x^2-\sqrt{c})\cdot\sin(x^2-\sqrt{c}))/\ln\lvert x^2-\sqrt{c}\rvert$
22	$y = \cos(x^2-a^2) + \sqrt{\lvert x^2-a^2\rvert}\cdot\sin(x^2-a^2)$
23	$y = (\sin(x\cdot d) + \cos(x+d))/e^{x\cdot d} - \ln\lvert x+d\rvert$
24	$y = (a\cdot x)/(1+(a\cdot x)^2) - \sqrt{\lvert\sin(a\cdot x)\rvert}/(a\cdot x)^2$
25	$y = \lvert x-a\rvert/(x+a)\cdot\ln\lvert x-a\rvert + (x-a)\cdot e^{x+a}$

3. ЛАБОРАТОРНАЯ РАБОТА № 2. ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ

В разветвляющейся вычислительной задаче происходит разветвление выполняемой последовательности действий в зависимости от результата проверки какого-либо условия.

Пример. Составить программу вычисления функции

$$\begin{cases} \ln|x+a| + \sin^2(x), \ \textit{если}\ x < a; \\ a + \sqrt{x-a}, \ \textit{если}\ x \geq a. \end{cases}$$

Программа на С++

```
#include "stdafx.h"   // Ставится по умолчанию в 2010
#include <iostream> // Подключение заголовочного файла
                    // с функциями ввода-вывода
using namespace std;   // Определение пространства имен
int _tmain(int argc, _TCHAR* argv[])  //Ставится по
                                      // умолчанию в 2010
{
setlocale(LC_ALL, "Russian");  //Установка //соответствующей
                               //кодовой страницы
  float a, x, y;                //Объявление переменных
  cout << "Введите a" << endl;  //Вывод фразы
  cin >> a;                     // Ввод a
  cout << "Введите x" << endl;  //Вывод фразы
  cin >> x;                     // Ввод x

  if (x < a)                    //Оператор ветвления
        y = log(abs(x + a)) + pow(sin(x), 2);  //Ветвь 1
  else
        y = a + sqrt(x - a);                  //Ветвь 2
  cout << "Результат = " << y << endl;  //Вывод надписи
                                        // и y
  system("pause");   // Ожидание нажатия клавиши
  return 0;
}
```

Таблица 5

Варианты заданий для реализации разветвляющихся вычислений

Номер варианта	Функция
1	$y = \begin{cases} a + x^2 - e^{-ax}; \ \textit{если}\ x \leq 10; \\ b - x + c \cdot x^3; \textit{если}\ x > 10. \end{cases}$
2	$y = \begin{cases} a \cdot x + b \cdot x^3; \ \textit{если}\ x > 13.4; \\ e^{ax} + 1.3; \textit{если}\ x \leq 13.4. \end{cases}$

Продолжение табл. 5

Номер варианта	Функция
3	$y=\begin{cases} x-\ln(a\cdot x);\ \textit{если}\ x\geq 7.3; \\ x^3-e^{bx};\ \textit{если}\ x<7.3. \end{cases}$
4	$y=\begin{cases} b-a\sin(b\cdot x);\ \textit{если}\ x\geq 12; \\ x-b^x;\ \textit{если}\ x<12. \end{cases}$
5	$y=\begin{cases} x^{2/3};\ \textit{если}\ x<4.3; \\ \sqrt{a\cdot x^2+b};\ \textit{если}\ x\geq 4.3. \end{cases}$
6	$y=\begin{cases} x^3-1;\ \textit{если}\ x<0.1; \\ \cos(-x^3);\ \textit{если}\ x\geq 0.1. \end{cases}$
7	$y=\begin{cases} \sin x^2-x^3;\ \textit{если}\ x\leq 1; \\ a\cdot x+b\cdot\ln(1+x);\ \textit{если}\ x>1. \end{cases}$
8	$y=\begin{cases} x+a\cdot\cos(\ln(x));\ \textit{если}\ x>3; \\ \sqrt{b+x^2};\ \textit{если}\ x\leq 3. \end{cases}$
9	$y=\begin{cases} tg(x);\ \textit{если}\ x\geq 5; \\ a\cdot x-\ln(b\cdot x)+c;\ \textit{если}\ x<5. \end{cases}$
10	$y=\begin{cases} a\cdot x+b\cdot x\cdot\sin(c\cdot x);\ \textit{если}\ x\leq 3.4; \\ \sin(a\cdot x)+b;\ \textit{если}\ x>3.4. \end{cases}$
11	$y=\begin{cases} a-b\cdot e^{-abx};\ \textit{если}\ x<2.7; \\ \ln(a+b\cdot x);\ \textit{если}\ x\geq 2.7. \end{cases}$
12	$y=\begin{cases} a\cdot x^2;\ \textit{если}\ x>4; \\ \sqrt{a+bx^3+\ln(x)};\ \textit{если}\ x\leq 4. \end{cases}$
13	$y=\begin{cases} a/(b\cdot x^2+c);\ \textit{если}\ x\geq 7.3; \\ a\cdot x+b\cdot x^3+c;\ \textit{если}\ x<7.3. \end{cases}$
14	$y=\begin{cases} e^{ax}+b;\ \textit{если}\ x>6.4; \\ e^{-ax}+c;\ \textit{если}\ x\leq 6.4. \end{cases}$
15	$y=\begin{cases} \ln(a\cdot x)+b;\ \textit{если}\ x\geq 3.9; \\ b\cdot x+c;\ \textit{если}\ x<3.9. \end{cases}$
16	$y=\begin{cases} \sin^2(a\cdot x)+b\cdot x;\ \textit{если}\ x>3.7; \\ \ln^3(x);\ \textit{если}\ x\leq 3.7. \end{cases}$
17	$y=\begin{cases} \ln\lvert x\rvert;\ \textit{если}\ x=3; \\ x;\ \textit{если}\ x=2; \\ \sqrt{\lvert\sin(x)\rvert};\ \textit{в остальных случаях}. \end{cases}$

Окончание табл. 5

Номер варианта	Функция
18	$y = \begin{cases} e^{ax};\ если\ x = 3; \\ 0.75 + a \cdot x;\ если\ x = 2; \\ a \cdot x;\ в\ остальных\ случаях. \end{cases}$
19	$y = \begin{cases} (1 + b \cdot x^2)^2;\ если\ x = -1; \\ \sqrt{(1 + b \cdot x^3)^2};\ если\ x = 3; \\ x;\ в\ остальных\ случаях. \end{cases}$
20	$y = \begin{cases} \ln\lvert\sin(x)\rvert;\ если\ x = 1; \\ e^{\lvert x\rvert} + \sqrt{\lvert x\rvert};\ если\ x = 2; \\ 0.5 + \lvert x\rvert;\ в\ остальных\ случаях. \end{cases}$

4. ЛАБОРАТОРНАЯ РАБОТА № 3. ПРОГРАММИРОВАНИЕ ЗАДАЧ С ОДНОМЕРНЫМИ МАССИВАМИ

Конечная именованная последовательность однотипных элементов называется массивом. При описании массива необходимо указать общее число входящих в массив элементов и тип этих элементов [1]. Например:

```
float a[10];
int b[15];
```

Из описания массивов следует, что массив **a** состоит из 10 вещественных чисел, а массив **b** – из 15 целых чисел.

Элементы массива нумеруются с нуля.

Инициализирующие значения для массивов записываются в фигурных скобках. Например:

```
int b[5] = {3, 2, 1, 4, 2};
```

Элементы массива получают следующие значения:

b[0] = 3; b[1] = 2; b[2] = 1; b[3] = 4; b[4] = 2;

Пример ввода массива a[10] приведен ниже.

```
float a[10];   // Описание массива
cout << "Введите массив a размерностью 10 элементов"  <<endl;
// Вывод фразы и перевод курсора
                        // на следующую строку
for (int i = 0; i < 10; i++)
{
  cout << "Введите a[" << i <<"]="; // Вывод фразы и i
  cin >> a[i];                      // Ввод a[i]
}
```

Пример вывода массива a[10] приведен ниже.

```
cout << "Массив a[10]" << endl;  // Вывод заголовка
                                 // и переход на следующую строку
```

```
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";  // Вывод элемента массива и пробела
    cout << endl;
```

При работе с массивами выполняются циклические действия. **Циклические** алгоритмы – это алгоритмы, в которых происходит повторение некоторой последовательности действий.

Пример. В массиве **a**(10) найти количество отрицательных элементов и сформировать новый массив **b**, каждый элемент которого получается делением соответствующего элемента массива **a** на найденное количество.

Программа на C++

```
#include "stdafx.h"  // Ставится в версии 2010
#include <iostream>    // Подключение заголовочного файла
using namespace std;   // Определение пространства имен
int _tmain(int argc, _TCHAR* argv[])  // Ставится в 2010
{
    const int n = 10;
    setlocale(LC_ALL, "Russian"); // Кодовая страница
    float a[n];                   //Объявление массива a
    float b[n];                   //Объявление массива b
cout << "Не забудьте отрицательные числа" << endl;
cout << "Введите массив a[" << n << "]" << endl;
for (int i = 0; i < n; i++)    //Цикл ввода элементов массива a
    {
        cout << "Введите a[" << i << "]=";
        cin >> a[i];
    }
    int k = 0;
    for (int i = 0; i < n; i++)  // Цикл подсчета количества
                                 // отрицательных элементов
                                // в массиве a
        if (a[i] < 0)
            k++;
    for (int i = 0; i < n; i++) //Цикл формирования массива b
        b[i] = a[i] / k;

    cout << "Массив a[" << n << "]\n";
    for (int i = 0; i < n; i++)  //Вывод на экран массива a
        cout << a[i]<<" ";
    cout << endl;
    cout << "Число отрицательных элементов " <<k << endl;
    cout << "Массив b[" << n << "]\n";
    for (int i = 0; i < n; i++)  //Вывод на экран массива b
```

```
        cout << b[i] << " ";
    cout << endl;
    system("pause");    // Ожидание нажатия клавиши
    return 0;
}
```

Варианты заданий по обработке одномерных массивов

1. Дан вещественный массив A(10). Найти количество положительных элементов в данном массиве. Вывести количество положительных элементов или фразу «положительных элементов нет». Вывести массив A.
2. Дан вещественный массив B(12). Найти среднее значение (сумма элементов, деленная на их количество) элементов массива. Вывести среднее значение и массив A.
3. Дан массив целых чисел D(12), найти и вывести на печать четные числа (делятся на два без остатка) из этого массива и массив D. Если четных элементов нет, то сообщить об этом.
4. Найти сумму максимального и минимального элементов целочисленного массива B(10), вывести сумму, найденные элементы и массив B.
5. Дан вещественный массив D(12). Ввести число K. Переписать в массив A элементы, меньшие числа K, а в B – элементы, большие числа K. Массивы D,A и B вывести. Если какой-то новый массив пуст, то сообщить об этом.
6. Дан массив целых чисел P(15), определить в нем количество и сумму четных чисел (делятся на два без остатка). Вывести сумму и количество, либо фразу, что четных элементов нет. Вывести массив P.
7. Дан целочисленный массив A(10), найти максимальный элемент среди отрицательных элементов, вывести его значение и номер. Если отрицательных элементов нет, то сообщить об этом. Вывести массив A.
8. Дан целочисленный массив F(12), найти минимальный элемент среди положительных элементов, вывести его значение и номер. Если положительных элементов нет, то сообщить об этом. Вывести массив F.
9. В вещественном массиве Z(10) найти минимальный элемент и сформировать новый массив, каждый элемент которого получится умножением элемента массива Z на найденный минимальный элемент. Вывести оба массива.
10. В вещественном массиве P(10) найти произведение положительных элементов. Вывести полученное произведение. Если положительных элементов нет, то сообщить об этом. Вывести массив P.
11. Дан целочисленный массив M(15), сформировать из него новый массив, переписав положительные элементы без изменения, а вместо отрицательных элементов - их модули. Вывести массив M и новый массив N.
12. Дан целочисленный массив L(15), определить количество и сумму чисел, кратных трем (делятся на три без остатка). Вывести найденные сумму и количество. Если кратных трем элементов нет, то сообщить об этом. Вывести массив L.

13. Дан целочисленный массив M(12), определить количество и сумму нечетных чисел (делятся на два с остатком) в массиве. Вывести сумму и количество. Если в массиве нет нечетных чисел, то сообщить об этом. Вывести массив M.

14. В вещественном массиве B(12) найти сумму элементов с четными номерами (номера 0, 2, 4, 6 и т.д.), а затем – с нечетными номерами (номера 1, 3, 5 и т.д.). Среди полученных сумм найти максимальную сумму. Вывести обе суммы, максимальную сумму и массив B.

15. В вещественном массиве Y(10) найти количество отрицательных элементов и сформировать новый массив, у которого элемент с четным номером (0, 2, 4 и т.д.) останется без изменений, а из элемента с нечетным номером (1, 3, 5 и т.д.) будет вычтено найденное количество. Вывести количество и оба массива. Если отрицательных элементов нет, то сообщить об этом и новый массив не формировать.

16. Сформировать новый массив из элементов вещественного массива Z(10), которые удовлетворяют условиям C1 <= Zi <= C2. Числа C1 и C2 ввести. Вывести оба массива. Если все элементы исходного массива Z не попадают в заданный интервал, то сообщить об этом и новый массив не формировать.

17. Дан вещественный массив Q(10). Сформировать из него новый массив A(10), в котором каждый отрицательный элемент массива Q будет возведен в квадрат, а из каждого положительного будет извлечен квадратный корень. Вывести оба массива.

18. Даны два целочисленных массива Z(10) и B(10). Создать из них новый массив, в котором каждый элемент с четным номером (номера 0, 2, 4 и т.д.) будет представлять сумму элементов массивов Z и B(с идентичными номерами(Z0 + B0, Z2 + B2, Z4 + B4, ...), а элемент с нечетным номером - их разность(Z1 - B1, Z3 - B3, ...). Вывести все массивы.

19. В целочисленном массиве Z(10) определить число соседств двух элементов с разными знаками и число соседств двух элементов с положительными знаками (использовать проверку знака у произведения соседей). Вывести массив Z и найденные количества.

20. В вещественном массиве B(10) найти сумму всех элементов с нечетными номерами (номера 1, 3, 5 и т.д.) и сумму всех элементов с четными номерами (номера 0, 2, 4 и т.д.). Вывести суммы и массив B.

5. ЛАБОРАТОРНАЯ РАБОТА № 4. ПРОГРАММИРОВАНИЕ ЗАДАЧ С МАТРИЦАМИ

Многомерные массивы задаются указанием каждого измерения в квадратных скобках [1]. Например:

```
float a[3][5];
float b[2][2];
```

В данном примере матрица **a** состоит из трех строк и пяти столбцов, все элементы матрицы являются вещественными числами. Матрица **b** состоит из

двух строк и двух столбцов. Элементы второй матрицы являются целыми числами.

В памяти такие массивы располагаются в последовательных ячейках построчно. Например, элементы матрицы b будут располагаться так:

b[0][0] b[0][1] b[1][0] b[1][1]

Примеры инициализации многомерного массива:

```
int mas2 [][]={ {1, 1}, {0, 2}, {1, 0}, {1, 2}  };
int mas2 [3][2]={1, 1, 0, 2, 1, 0};
```

Пример ввода вещественной матрицы размером 3 на 4.

В данной матрице будет три строки, номера которых изменяются от 0 до 2.

В матрице будет четыре столбца, номера которых меняются от 0 до 3.

```
const int n = 3;
const int m = 4;
float x[n][m];
cout <<"Введите матрицу x размерностью 3 на 4"
<< endl;
for (int i = 0; i < n; i++)
  for (int j = 0; j < m; j++)
  {
        cout << "Введите x[" << i << "][" << j              << "]=";
        cin >> x[i][j];
  }
```

Пример вывода вещественной матрицы приведен ниже.

```
const int n = 3;
const int m = 4;
float x[n][m];
cout << "\nМатрица x[" << n << "][" << m << "]\n";
for (int i = 0; i < n; i++)
{
  for (int j = 0; j < m; j++)
        printf("%5.2f ", x[i][j]);
  printf("\n");
}
```

С помощью оператора printf каждый элемент матрицы выводится как вещественное число, занимающее 5 позиций, из которых две последние отводятся под дробную часть. Матрица выводится по строкам как таблица. Оператор printf("\n") обеспечивает переход курсора на следующую строку при окончании вывода строки матрицы.

Пример ввода целочисленной матрицы y[4][4] приведен ниже. Матрица имеет 4 строки и 4 столбца.

```
const int n = 4;
int y[n][n];
cout << "Введите матрицу y размерностью" << n <<" на " << n << endl;
```

```
for (int i = 0; i < n; i++)
  for (int j = 0; j < n; j++)
  {
        cout << "Введите y[" << i << "][" << j << "]=";
        cin >> y[i][j];
  }
```

Пример вывода целочисленной матрицы приведен ниже.

```
const int n = 4;
int y[n][n];
cout << "\nМатрица y[" << n << "][" << n <<"]\n";
for (int i = 0; i < n; i++)
{
  for (int j = 0; j < n; j++)
        printf("%2d ", y[i][j]);
  printf("\n");
}
```

Каждый элемент матрицы выводится как целое число и занимает две позиции.

Реализован вывод матрицы как таблицы. Во вложенном цикле фиксируется параметр внешнего цикла, затем меняется параметр внутреннего цикла. После того, как проработает внутренний цикл с параметром j и выведутся элементы i-ой строки, с помощью оператора printf("\n") произойдет переход на следующую строку.

Таким образом, каждая строка матрицы будет выводиться с новой строки.

Пример. Составить программу, которая в матрице **a**(4,5) определит сумму элементов каждой строки, выберет среди них наибольшую и выведет на экран дисплея матрицу **a**, суммы строк, наибольшую сумму и номер строки с наибольшей суммой.

Программа на C++

```
#include "stdafx.h"     // По умолчанию в 2010
#include <iostream>   // Подключение заголовочного файла
                      // с функциями ввода-вывода
#include <iomanip>   // Подключение заголовочного файла
                      // для управления отображения данных на экране
using namespace std;  // Определение пространства имен
int _tmain(int argc, _TCHAR* argv[])  // По умолчанию в 2010
{
setlocale(LC_ALL, "Russian");  // Кодовая страница
      const int m = 4;
      const int n = 5;
      float a[m][n];              // Объявление матрицы a
      float c[m];                 // Объявление массива c
      cout << "Введите матрицу A размерностью " << m << " на " << n << endl;
```

```
        for (int i = 0; i < m; i++) // Ввод матрицы a в цикле
                for (int j = 0; j < n; j++)
                {
                cout << "Введите a[" << i << "][" << j << "]=";
                        cin >> a[i][j];
                }
        for (int i = 0; i < m; i++)  // Расчет суммы элементов
                                //  каждой  строки и запись
                               // полученных значений в массив c
        {
                c[i] = 0;
                for (int j = 0; j < n; j++)
                        c[i]  += a[i][j];
        }
        float s = c[1];
        int max_s = 1;
        for (int i = 1; i < m; i++)    // Нахождение максимальной
                                    // суммы элементов строки
                if (s < c[i])
                {
                        s = c[i];
                        max_s = i;
                }
        cout << "Матрица a[" << m << "][" << n << "]" << endl;
// установка формата вывода вещественных чисел
// фиксированный формат с двумя знаками после запятой
        cout << setiosflags(ios::fixed) << setprecision(2);
       for (int i = 0; i < m; i++) //Вывод матрицы a на экран
        {
                for (int j = 0; j < n; j++)
                        cout << setw(6)<< a[i][j] << " ";
                 // setw() задает ширину отображаемой переменной
                cout << endl;
        }
                cout << "Суммы элементов строк" << endl;
        for (int i = 0; i < m; i++)
        cout << "Строка " << i <<" - сумма " << c[i] << endl;
        cout << endl;
        cout << "Максимальная сумма - " << s << ", номер строки - " << max_s <<
endl;
        system("pause");   // Ожидание нажатия клавиши
        return 0;
      }
```

Варианты заданий по обработке матриц

1. Дана вещественная матрица F(3,4). Сформировать массив E(3), элементы которого представляют суммы положительных элементов строк матрицы F. Если в строке нет положительных элементов, то сумме присвоить 0. Вывести E и F.

2. Дана вещественная матрица A(3,3). Найти и вывести сумму максимального и минимального элементов матрицы A(3,3). Вывести матрицу A.

3. Найти максимальный элемент в каждой строке целочисленной матрицы H(4,4). Вывести на печать эти элементы, номера столбцов и строк, в которых они расположены. Вывести матрицу H.

4. Дана вещественная матрица C(3,5), сформировать массив K(3), элементы которого представляют произведение элементов строк матрицы C. Вывести массив K и матрицу C.

5. Дана вещественная матрица R(4,4), найти минимальный элемент среди положительных элементов матрицы, вывести его значение, номер строки и столбца, где он расположен. Если в матрице нет положительных элементов, то сообщить об этом. Вывести матрицу R.

6. Дана вещественная матрица D(4,4), найти максимальный элемент среди отрицательных элементов матрицы, вывести его значение, номер строки и столбца, где он расположен. Если в матрице нет отрицательных элементов, сообщить об этом. Вывести матрицу D.

7. Найти минимальный элемент в каждом столбце целочисленной матрицы W(3,4). Вывести на печать эти элементы, номера столбцов и строк, в которых они расположены. Вывести матрицу W.

8. Найти максимальный элемент на главной диагонали целочисленной матрицы F(4,4) и сформировать матрицу P(4,4), элементы которой получены из элементов F путем умножения на максимальный элемент. Вывести матрицы P и F.

9. Даны вещественные числа X_1, X_2, X_3. Получить вещественную матрицу K(3,3), для которой $K_{ij} = X_i - 3X_j$. Вывести полученную матрицу K и массив X.

10. Дана вещественная матрица C(3,3). Получить новую вещественную матрицу D(3,3) путем деления всех элементов матрицы C на ее наибольший по модулю элемент. Вывести наибольший по модулю элемент. Вывести матрицы C и D.

11. Дана вещественная матрица Q(3,5), найти среднее арифметическое каждого столбца (сумма элементов столбца, деленная на три) и вывести их. Вывести матрицу Q.

12. Дана вещественная матрица X(3,4), найти максимальный элемент в последнем столбце и сформировать матрицу Y(3,4) из соответствующих элементов X(3,4) путем умножения на найденный максимальный элемент. Вывести матрицы X и Y.

13. Найти первый отрицательный элемент в целочисленной матрице R(3,3) и сформировать матрицу P(3,3), элементы которой получаются из соответствую-

щих элементов матрицы R путем умножения на первый отрицательный. Если отрицательных элементов в матрице R нет, то сообщить об этом и матрицу P не формировать. Вывести матрицы R и P (если она сформирована).

14. Получить вещественную матрицу D(4,4):

$$D_{ij} = \begin{cases} \sin(1+j), & \text{если } i < j; \\ 1, & \text{если } i = j; \\ \cos\left(\dfrac{i+j}{2 \cdot i + 3 \cdot j}\right), & \text{если } i > j. \end{cases}$$

Вычислить сумму элементов матрицы D. Вывести матрицу D и сумму.

15. Дана вещественная матрица P(4,6), найти среднее арифметическое каждого столбца (сумма элементов столбца, деленная на 4), имеющего четный номер (номера 0, 2, 4). Вывести средние значения четных столбцов и матрицу P.

16. Даны вещественные массивы A(5), B(10). Получить вещественную матрицу C(5,10), для которой $C_{ij} = A_i / (1 + |B_j|)$. Вывести матрицу C. Вывести массивы A и B.

17. Определить, сколько положительных элементов содержит вещественная матрица T(6,6). Элементы матрицы вычисляют по формуле: Tij = sin((i2 - j2) / 6. Вывести матрицу T. Вывести количество положительных элементов. Если положительных элементов нет, то сообщить об этом.

18. Дана целочисленная матрица B(3,4). Все отрицательные элементы заменить на -1, все положительные - на 1, нулевые оставить без изменения. Вывести преобразованную и исходную матрицы.

19. Дана целочисленная матрица Z(3,3). Заменить на нули все ее элементы, расположенные на главной диагонали и выше ее. Вывести преобразованную и исходную матрицы.

20. Дана вещественная матрица Y(3,4). Все элементы матрицы с наибольшими значениями (если их несколько) заменить нулями. Вывести преобразованную и исходную матрицы.

6. ЛАБОРАТОРНАЯ РАБОТА № 5. ПРОГРАММИРОВАНИЕ ЗАДАЧ ПО ОБРАБОТКЕ СТРОК

Строка представляет собой массив символов, заканчивающийся нуль-символом.

Нуль-символ – это символ с кодом, равным 0, что записывается в виде управляющей последовательности ‘\0’.

По расположению нуль-символа определяют фактическую длину строки.

Пример 1.

char str[10] = “My text”;

В этом примере под строку выделяется 10 байт, 7 байт занято под символы строки, а восьмой байт – под нуль-символ.

Если строка при определении инициализируется, ее размерность можно опускать (компилятор сам выделит соответствующее количество байт).

Пример 2.

char str[] = “Program”;

В примере 2 под строку выделено 8 байт.

Одной строке нельзя присвоить другую строку (поскольку строка является массивом). Присвоение может выполняться с помощью цикла или функций стандартной библиотеки **<string>**.

Для работы со строками используют специальные функции. В табл. 11 приведены основные функции для работы со строками. Функция gets находится в модуле stdio.h, остальные в модуле – string.

Таблица 6

Основные функции для работы со строками

Функция	Назначение
gets_s(str)	Считывает строку, введенную пользователем с клавиатуры, и записывает ее в переменную str.
strcat_s(s1, s2)	Функция добавляет строку s2 к s1 и возвращает s1.
strcpy_s(s1, s2)	Функция копирует строку s2 в s1 и возвращает s1.
strlen(s)	Функция возвращает длину строки s (при этом символ конца строки не учитывается).
strncat_s(s1, s2, n)	Функция добавляет не более n символов из строки s2 к s1 и возвращает s1.
strncpy_s(s1, s2, n)	Функция копирует не более n символов из строки s2 в s1 и возвращает s1.
strpbrk(s1, s2)	Функция ищет символ одной строки в другой. Функция возвращает указатель на символ, являющийся первым вхождением любого из символов из s2 в строку s1, если его нет, возвращается NULL (пустое значение указателя).
strrchr(s, ch)	функция ищет символ в строке. Функция возвращает указатель на первое вхождение символа ch в строку s, если его нет - возвращает NULL.
strstr(s1, s2)	Функция ищет первое вхождение подстроки s2 в строке s1. Функция возвращает указатель на элемент из s1, с которого начинается s2 или NULL в случае неудачи
isalnum(c)	возвращает значение true, если с является буквой или цифрой, и false в других случаях
isspace(c)	возвращает значение true, если с является пробелом, и false в других случаях
toupper(c)	если символ с, является символом нижнего регистра, то функция возвращает преобразованный символ с в верхнем регистре, иначе символ возвращается без изменений.
strchr(s,c)	поиск первого вхождения символа с в строке s. В случае удачного поиска возвращает указатель на место первого вхождения символа с. Если символ не найден, то возвращается ноль.
strcspn(s1,s2)	определяет длину начального сегмента строки s1, содержащего те символы, которые не входят в строку s2

Окончание табл.6

Функция	Назначение
strspn(s1,s2)	возвращает длину начального сегмента строки s1, содержащего только те символы, которые входят в строку s2
strprbk(s1,s2)	Возвращает указатель первого вхождения любого символа строки s2 в строке s1
atof(s1)	преобразует строку s1 в тип double
atoi(s1)	преобразует строку s1 в тип int
atol(s1)	преобразует строку s1 в тип long int

Примеры использования строковых функций.

Пример 1. Применение функции **strcat_s(s1, s2), которая** добавляет **s2** к **s1** и возвращает **s1**.

```
char s1[10]="New";
char s2[10]="Text";
strcat(s1,s2);
```

Переменной **s1** будет присвоено значение NewText.

Пример 2. Применение функции **strcpy_s(s1, s2)**, которая копирует **s2** в **s1** и возвращает **s1.**

```
char s1[10]="New";
char s2[10]="Text";
strcpy(s1,s2);
```

В переменной **s1** окажется значение Text.

Пример 3. В строке размером до 30 символов определить, сколько раз встречается определенный символ. Строка и символ вводятся пользователем. Вывести строку, символ и найденное количество.

Программа на C++

```
#include "stdafx.h"        // Установлен в 2010 по умолчанию
#include <Windows.h>  // Добавляется для ввода русских букв
#include <iostream>     // Подключение заголовочного файла
                        // с функциями ввода-вывода
using namespace std;   // Определение пространства имен
int _tmain(int argc, _TCHAR* argv[])  // По умолчанию в  2010

{

SetConsoleCP(1251);    // Возможность вводить и отображать
SetConsoleOutputCP(1251);      // русские буквы
        char str[30]; // Строка, задаваемая пользователем
        char s;                 // Символ, задаваемый пользователем
cout << "Введите строку" << endl;
        cin.getline(str,30); // Ввод строки в переменную str
        cout << "Введите символ" << endl;
```

```
        cin >> s;               // Ввод символа
        int k = 0;      // Инициализация счетчика для расчета
                        // количества символов s в строке str
// Цикл подсчета количества символов s в строке str
// strlen – функция определения длины строки
        for (int i = 0; i < strlen(str); i++)
                if (str[i] == s)
                        k++;
        // Вывод результата на экран
        cout << "Количество вхождений символа " << s << " в строку \n" << str <<
"\n равно " << k << endl;
        system("pause");                // Ожидание нажатия клавиши
    }
```

Для отображения русских букв необходимо на заглавии окна при вводе данных нажать правой кнопкой мыши, выбрать пункт «Свойства», выбрать шрифт Lucida Console.

Варианты заданий по обработке строк

1. В текстовой строке проверить, какая буква встречается чаще – 'a' или ‘c’. В качестве результата вывести соответствующее текстовое сообщение.
2. В текстовой строке проверить правильность расстановки круглых скобок (т.е. соответствует ли количество открывающих скобок количеству закрывающих). Вывести сообщение о результате.
3. В текстовой строке подсчитать число вхождений тройки символов 'abc'. Вывести количество и текстовую строку.
4. В текстовой строке определить количество слов, начинающихся с буквы 'a'. Вывести количество и текстовую строку.
5. В текстовой строке определить количество слов, заканчивающихся буквой 'w'. Вывести количество и текстовую строку.
6. В текстовой строке определить количество слов, начинающихся и заканчивающихся одной и той же буквой. Вывести количество и текстовую строку.
7. В текстовой строке определить количество слов, в которых есть хотя бы одна буква 'd'. Вывести количество и текстовую строку.
8. В текстовой строке определить количество слов, в которых содержится ровно две буквы ‘e’. Вывести количество и текстовую строку.
9. В текстовой строке найти самое длинное слово и вывести его на экран вместе с его длиной. Вывести количество и текстовую строку.
10. В текстовой строке заменить все пробелы знаком '_'. Вывести старую и новую строки.
11. В текстовой строке поменять все символы '!' на '?'. Вывести старую и новую строки.

12. На основе текстовой строки, введенной пользователем, создать новую строку, из которой будут удалены все лишние пробелы (т.е. из нескольких подряд идущих пробелов оставить один). Вывести старую и новую строки.

13. На основе текстовой строки, введенной пользователем, создать новую строку, в которой все пары символов 'ph' будут заменены на символ 'f'. Вывести старую и новую строки.

14. На основе текстовой строки, введенной пользователем, создать новую, из которой будут удалены все знаки ‘-’ и ‘+’. Вывести старую и новую строки.

15. В текстовой строке найти количество символов арифметических операций (т.е. знаков ‘+’, ‘-‘, ‘/’ и ‘*’). Вывести количество и строку.

16. На основе текстовой строки, введенной пользователем, создать новую строку, где каждый третий символ заменен знаком ‘#’. Вывести старую и новую строки.

17. Подсчитать, сколько раз пара 'th' входит в заданный текст. Вывести количество и строку.

18. В текстовой строке найти самое короткое слово и вывести его на экран вместе с его длиной. Вывести строку.

19. В строке заданы фамилия, имя и отчество студента (именно в таком порядке). Напечатайте его фамилию и инициалы, и введенную строку.

20. Дана текстовая строка, слова в которой разделены пробелами.

7. ЛАБОРАТОРНАЯ РАБОТА № 6. ПРОГРАММИРОВАНИЕ ЗАДАЧ СО СТРУКТУРАМИ

Структура может содержать элементы разных типов.

Структура задается следующим образом:

```
struct [ имя_типа ] {
  тип_1 элемент_1;
  тип_2 элемент_2;
  …
  тип_n элемент_n;
} [ список_переменных ];
```

Элементы структуры называются полями структуры и могут иметь любой тип, кроме типа этой же структуры. В описании полей можно применять указатели на структуру.

Описание структуры определяет новый тип, имя которого можно использовать в дальнейшем наряду со стандартными типами.

Пример задания структуры.

```
struct Worker{      // описание нового типа Worker
  char fio[30];
  int age, code;
  double zarpl;
}; // описание заканчивается точкой с запятой
```

Определение массива типа Worker записывают следующим образом:

Worker mas_struct[100];

Для **инициализации структуры** значения ее элементов перечисляют в фигурных скобках в порядке их описания:

```
struct {
  char fio[30];
  int age, code;
  double zarpl;
} worker = {"Миронов", 31, 215, 18400.50};
```

Доступ к полям структуры выполняется с помощью указания имени переменной типа структуры и через **.** (точку) имени поля. Примеры обращения к полям структуры для переменной (www) и элементов массива (mas_struct) приведены ниже:

```
Worker www, mas_struct[100]; // Worker – имя структуры
www.fio = "Миронов";
www.age = 25;
www.code = 220;
www.zarpl = 22000;
mas_struct[8].fio = “Петрова”;
mas_struct[8].code = 215;
mas_struct[8].age = 35;
mas_struct[8].code = 245;
mas_struct[8].zarpl = 35000;
```

Пример. Создать массив, элементы которого имеют следующую структуру:

– ФИО студента;
– курс;
– группа;
– оценка по математике;
– оценка по программированию;
– оценка по физике.

Вывести на экран массив записей в виде таблицы следующего вида.

Студенты

ФИО	Курс	Группа	Математика	Программирование	Физика	Ср. балл

Средний балл рассчитать по всем трем оценкам.

Программа на С++

```
#include "stdafx.h"        // в 2010 добавляется по умолчанию
#include <Windows.h>  // Добавляется для ввода русских букв
#include <iostream>       // Подключение заголовочного файла
                          // для ввода-вывода
```

```
#include <iomanip>            // Подключение заголовочного файла
                             //   для управления отображением
                             // данных на экране
#include <string>            // Подключения файла с функциями
                             // работы со строками
using namespace std;    // Определение пространства имен
int _tmain(int argc, _TCHAR* argv[]) // в 2010 по умолчанию

{
SetConsoleCP(1251);    // Возможность вводить и отображать
SetConsoleOutputCP(1251);      // русские буквы
        const int n = 3;  // Размерность массива со структурами
        const int m = 77;    // Ширина таблицы в символах
                        // Понадобится для рисования таблицы
        // Описание структуры
        struct stud {
                string fio;
                int kurs;
                string gruppa;
                int math, prog, fiz;
        };
        stud mas[n];           // Массив из структур
        cout << "Введите " << n << " записей" << endl;
        for (int i = 0; i < n; i++)
        {
                cout << "Введите ФИО студента" << endl;
// очистка буфера ввода для корректной работы
// функции getline
                if (i != 0)
                   while (cin.get() != '\n');
                getline(cin,mas[i].fio);  // Ввод строки (фамилии)
                cout << "Введите курс" << endl;
                cin >> mas[i].kurs;
                cout << "Введите группу" << endl;
                while (cin.get() != '\n'); // считываем символ, пока
                                           // строка не закончится
                getline(cin,mas[i].gruppa); //Ввод строки (группы)
                cout << "Введите оценку по математике" << endl;
                cin >> mas[i].math;
                cout << "Введите оценку по программированию" << endl;
                cin >> mas[i].prog;
                cout << "Введите оценку по физике" << endl;
                cin >> mas[i].fiz;
```

```
        }
        // Вывод таблицы со структурами на экран
        cout << "                              Студенты" << endl;

        // Прорисовка верхней границы таблицы
        for (int j = 0; j < m; j++)
                cout << "-";
        cout << endl;

        // Прорисовка шапки таблицы
        cout << "|        ФИО          | Курс |  Группа  | Матем. | Прог. | Физ. | Ср. балл
|"<< endl;
        // Прорисовка нижней границы шапки таблицы
        for (int j = 0; j < m; j++)
                cout << "-";
        cout << endl;
        float sr_ball;          // Переменная для расчета среднего балла
        // Прорисовка основной части таблицы
        // В ходе итерации цикла на экран выводится строка
        // таблицы, содержащая все поля текущей структуры
        for (int i = 0; i < n; i++)
        {
                // Расчет среднего балла для текущей структуры
                sr_ball = (mas[i].math + mas[i].prog + mas[i].fiz) / 3.0;
                cout << "| " << setw(20) << mas[i].fio; // Вывод
                                        // фамилии в 20 позициях
                cout << " | " << setw(4) << mas[i].kurs;
                cout << " | " << setw(8) << mas[i].gruppa;
                cout << " | " << setw(6) << mas[i].math;
                cout << " | " << setw(5) << mas[i].prog;
                cout << " | " << setw(4) << mas[i].fiz;
                cout << " | " << setiosflags(ios::fixed) <<
                        setprecision(2) << setw(8) <<  sr_ball << " | " << endl;
        }
                // Прорисовка нижней границы таблицы
        for (int j = 0; j < m; j++)
                cout << "-";
        cout << endl;
   system("pause");   // Ожидание нажатия клавиши
}
```

Варианты заданий по обработке массивов структур

1. Создать массив из 5 структур, каждая из которых содержит следующие поля: адрес проживания, плата за квартиру, отчисления в фонд капитального ремонта, плата за стационарный телефон. Рассчитать суммарную плату и составить отчет.

Оплата ЖКХ

Адрес проживания	Плата за квартиру	Отчисления в фонд	Плата за телефон	Суммарная плата
Итого				

2. Создать массив из 5 структур, каждая из которых содержит следующие поля: номер киоска, название газеты, стоимость газеты в розничной продаже, количество экземпляров. Рассчитать общую стоимость каждой газеты в киоске и составить отчет.

Стоимость печатной продукции

Номер киоска	Название газеты	Розничная цена	Количество экземпляров	Общая стоимость
Итого				

3. Создать массив из 5 структур, каждая из которых содержит следующие поля: номер маршрута, номер автобуса, количество мест, цена билета. Вычислить стоимость одной поездки при полной загруженности автобуса и составить отчет.

Стоимость проезда

Номер маршрута	Номер автобуса	Кол-во мест	Цена билета	Стоимость всех билетов
Итого				

4. Создать массив из 5 структур, каждая из которых содержит следующие поля: номер тарифа, номер месяца, суточная стоимость звонков, количество дней в месяце, стоимость доступа в интернет. Рассчитать месячный платеж и составить отчет, приведенный ниже.

Платежи за услуги связи

Номер тарифа	Номер месяца	Кол-во дней в месяце	Суточная стоимость звонков	Стоимость интернета	Общая стоимость
Итого					

5. Создать массив из 5 структур, каждая из которых содержит следующие поля: код операции, тип прибора, годные, брак. Рассчитать процент выхода годных приборов и составить отчет. Процент выхода годных приборов вычисляют по формуле: годные * 100 / (годные + брак).

Выход годных приборов

Код операции	Тип прибора	Годные	Брак	Процент
Итого				

6. Создать массив из 5 структур, каждая из которых содержит следующие поля: изделие, план производства, фактическое количество. Рассчитать процент выполнения плана и составить отчет. Процент выполнения плана вычисляют по формуле: фактическое количество * 100 / план производства.

Выполнение плана

Изделие	План	Фактическое кол-во	Процент выполнения
Итого			

7. Создать массив из 5 структур, каждая из которых содержит следующие поля: ФИО, количество дней отпуска, среднее количество рабочих дней в месяце, средняя месячная заработная плата. Чтобы вычислить отпускные необходимо среднюю месячную заработную плату разделить на среднее количество рабочих дней в месяце и умножить на количество дней отпуска. Вычислить отпускные и составить ведомость такого типа.

Отпускные выплаты

Порядковый номер	ФИО	Кол-во дней отпуска	Кол-во дней работы	Средняя заработная плата	Отпускные
Итого					

8. Создать массив из 5 структур, каждая из которых содержит следующие поля: ФИО студента, количество лабораторных работ по программи-

рованию, количество сделанных работ по программированию, количество лабораторных работ по экономике, количество сделанных работ по экономике. Рассчитать процент выполнения лабораторных работ и выдать отчет. Процент выполнения лабораторных работ вычисляют по формуле: количество сделанных работ * 100 / количество лабораторных работ.

Выполнение лабораторных работ

ФИО	Программирование			Экономика		
	План	Факт	Процент	План	Факт	Процент
Итого				Итого		

9. Создать массив из 5 структур, каждая из которых содержит следующие поля: индекс товара, наименование, сорт, количество, цена. Рассчитать стоимость товара и составить ведомость такого типа.

Товарная ведомость

Индекс товара	Наименование	Сорт	Количество	Цена	Стоимость
Итого					

10. Создать массив из 5 структур, каждая из которых содержит следующие поля: разряд, часовая тарифная ставка, количество отработанных часов. Рассчитать зарплату рабочих-повременщиков и составить отчет.

Ведомость на оплату

ФИО	Разряд	Часовая тариф. ставка	Отработано часов	Заработная плата
Итого				

11. Создать массив из 5 структур, каждая из которых содержит следующие поля: название детали, название материала, цена за единицу материала, количество деталей, норма расхода материала на деталь. Рассчитать расход материала на все количество деталей и выдать ведомость.

Расходная ведомость

Деталь	Материал	Количество	Норма	Цена	Стоимость
Итого					

Вариант № 12

12. Создать массив из 5 структур, имеющие следующие поля: номер рейса автобуса, наименование рейса, количество проданных билетов, цена за билет. Рассчитать стоимость проданных билетов и выдать ведомость, приведенную ниже.

Справка о проданных билетах

Номер рейса	Наименование рейса	Кол-во продан. билетов	Цена	Стоимость
Итого				

13. Создать массив из 5 структур, имеющие поля: марка автомобиля, грузоподъемность, количество рейсов в день, количество рабочих дней в месяце. Рассчитать общее количество перевезенных грузов (количество рейсов в день * количество рабочих дней в месяце). Создать отчет, приведенный ниже.

Справка о перевозках

Марка автомобиля	Грузоподъемность	Кол-во рейсов в день	Кол-во рабочих дней в месяце	Общее кол-во грузов
Итого				

14. Создать массив из 5 структур, имеющие поля: ФИО студента, шифр группы, наименование дисциплины, общее количество занятий, количество пропущенных занятий. Рассчитать процент посещаемости ((общее количество занятий – количество пропущенных занятий) * 100 / общее количество занятий). Создать отчет, приведенный ниже.

Справка о посещаемости

ФИО студента	Шифр группы	Дисциплина	Общ. кол-во занятий	Кол-во пропущеных занятий	Процент посещаемости
Итого					

15. Создать массив из 5 структур, каждая из которых содержит следующие поля: табельный номер, ФИО работника, оклад, количество отработанных дней, количество рабочих дней в месяце. Рассчитать заработную плату и выдать ведомость.

Справка о заработной плате

Табельный №	ФИО	Оклад	Кол-во отраб. дней	Кол-во раб. дней	Заработная плата
Итого					

16. Создать массив из 5 структур, каждая из которых содержит следующие поля: код и наименование дисциплин, количество часов лекций, практики и лабораторных работ. Рассчитать общее количество часов занятий и выдать ведомость.

Учебный план

Код	Наименование Дисциплины	Количество часов			
		Лекции	Практика	Лаб. работа	Итого
Итого					

17. Создать массив из 5 структур, каждая из которых содержит следующие поля: табельный номер, ФИО, расценка за деталь, количество уже сделанных деталей. Рассчитать стоимость сделанных деталей и выдать ведомость следующего вида.

Ведомость на оплату

Табельный номер	ФИО	Расценка за деталь	Количество	Стоимость
Итого				

18. Создать массив из 5 структур, каждая из которых содержит следующие поля: номер счета, ФИО вкладчика, сумма денег на вкладе, годовой процент. Рассчитать годовой процент в рублях и составить ведомость следующего вида.

Справка о годовом проценте

Номер счета	ФИО	Сумма	Годовой итог	
			в процентах	в рублях
Итого				

19. Создать массив из 5 структур, каждая из которых содержит следующие поля: марка автомобиля, норма расхода бензина, л/км; среднемесячный пробег, км; цена бензина. Рассчитать стоимость израсходованного бензина и выдать ведомость такого типа.

Расходная ведомость

Марка автомобиля	Норма расхода бензина	Пробег	Цена	Стоимость
Итого				

20. Создать массив из 5 структур, каждая из которых содержит следующие поля: код станка, время фактической работы станка, продолжительность смены. Рассчитать процент загруженности станка и составить ведомость такого типа.

Справка о загруженности оборудования

Код РТК	Время фактической работы	Продолжительность смены	Процент загруженности
Итого			

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сергеев М.Ю. Программирование вычислительных задач на С++: учеб. пособие / М.Ю. Сергеев. Воронеж: ФГБОУ ВПО «Воронежский государственный технический университет», 2015. - 192 с.

2. Гребенникова, Н. И. Программирование на языке высокого уровня: лабораторный практикум [Электронный ресурс]. - / Н. И. Гребенникова, М. Ю. Сергеев, Т. И. Сергеева. - ФГБОУ ВО «Воронежский государственный технический университет», 2021.

3. Павловская, Т.А. С/С++. Программирование на языке высокого уровня [Текст]/ Т.А. Павловская. - СПб.: Лидер, 2010. - 461 с.

4. Павловская, Т.А. С/С++. Структурное и объектно-ориентированное программирование [Текст]: практикум / Т.А. Павловская, Ю.А. Щупак. - СПб.: Питер, 2010. - 352 с.

5. Холопкина Л.В. Программирование на С++: учебное пособие. – Воронеж, ВГТУ. –2011.–184 с.

6. Культин Н.В. Основы программирования в Microsoft Visual C++ 2010. – СПб.: БХВ-Петербург, 2012. – 384 с.

7. Павловская Т.А. С/С++. Процедурное и объектно-ориентированное программирование: учебник для вузов. – СПб.: Питер, 2014.

8. Гагарина Л.Г. Программирование на языке высокого уровня. Программирование на языке С++: учебное пособие. – М.: Форум, 2012.

9. Брайан У.К., Деннис М.Р. Язык программирования С. – М.: Вильямс, 2015.

10.Прата С. Язык программирования С. Лекции и упражнения. – М.: Вильямс, 2016.

ОГЛАВЛЕНИЕ

ПРОГРАММИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ № 1-6
по дисциплине «Основы программирования и алгоритмизации»
для студентов направления 38.03.05
«Бизнес-информатика”
(профиль «Информационные системы в бизнесе»)
очной и заочной форм обучения

Составители:
Гребенникова Наталия Ивановна
Сергеев Михаил Юрьевич
Сергеева Татьяна Ивановна

Компьютерный набор Н.И. Сергеевой

Подписано к изданию 26.01.2022.
Уч.-изд. л. 1,9.

ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14