­­­Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Н. Н. Пустовалова, Н. В. Пацей

**ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

**в 2-х частях**

**Часть 1**

**[Лабораторный практикум](#_30j0zll)**

Минск 2024

**ОГЛАВЛЕНИЕ**

**Лабораторная работа № 1.** [**Системы счисления**](#3mzq4wv)

**Лабораторная работа № 2.** [**Способы представления алгоритмов**](#1ci93xb)

**Лабораторная работа № 3.** [**Основные элементы языка С++. MS Visual Studio 2010**](#qsh70q)

**Лабораторная работа № 4.** [**Ввод и вывод информации**](#3fwokq0)

**Лабораторная работа № 1. Системы счисления**

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Перевести несколько чисел (например, 12, 77, 436 и др.) из восьмеричной системы счисления в двоичную.  Перевести несколько чисел (например, B8, 359, AA, 81 и др.) из шестнадцатеричной системы счисления в двоичную. | 12(8 c/c) =001 010(2 c/c)  77(8 c/c) =111 111(2 c/c)  436(8 c/c) =100 011 110(2 c/c)  22(8 c/c) =010 010(2 c/c)  6(8 c/c) =110(2 c/c)  B8(16 c/c) =1011 1000(2 c/c)  359(16 c/c) =0011 0101 1001(2 c/c)  AA(16 c/c) =1010 1010(2 c/c)  81(16 c/c) =1000 0001(2 c/c)  16(16 c/c) =0001 0110(2 c/c)  C4(16 c/c) =1100 0100(2 c/c) |
| 2. Перевести несколько чисел (например, 101111001, 0110, 011 и др.) из двоичной системы счисления в восьмеричную.  Перевести несколько чисел (например, 1111, 10101010 и др.) из двоичной системы счисления в шестнадцатеричную. | 101 111 001(2 c/c) =101 111 001 (2 c/c) = 571(8 c/c)  0110(2 c/c) =000 110(2 c/c) = 06(8 c/c)  011(2 c/c) =011(2 c/c) = 3(8 c/c)  01110(2 c/c) =001 110(2 c/c) = 16(8 c/c)  00100(2 c/c) =000 100(2 c/c) = 04(8 c/c)  1111(2 c/c)  =1111 (2 c/c) =F(16 c/c)  10101010 (2 c/c)  =1010 1010 (2 c/c) =AA (16 c/c)  00111010(2 c/c)  =0011 1010 (2 c/c) =3A (16 c/c)  110111(2 c/c)  =0011 0111 (2 c/c) =37(16 c/c) |
| 3. Перевести несколько чисел (например, 153, 236 и др.) из десятичной системы счисления в двоичную. | 153(10 c/c) = 1001 1001 (2 c/c) =1∙2^7+0∙2^6+0∙2^5+1∙2^4+1∙2^3+0∙2^2+0∙2^1+1∙2^0 = 153(10 c/c)  153/2=76 1  76/2=38 0  38/2=19 0  19/2=9 1  9/2=4 1  4/2=2 0  2/2=1 0  1/2=0 1  236(10 c/c) =1110 1100 (2 c/c) = 1∙2^7+1∙2^6+1∙2^5+0∙2^4+1∙2^3+1∙2^2+0∙2^1+0∙2^0 =236(10с/с)  236/2=118 0  118/2=59 0  59/2=29 1  29/2=14 1  14/2=7 0  7/2=3 1  3/2=1 1  1/2=0 1  128(10 c/c) = 1000 0000(2 c/c) = 1∙2^7+0∙2^6+0∙2^5+0∙2^4+0∙2^3+0∙2^2+0∙2^1+0∙2^0 =128(10с/с)  128/2=64 0  64/2=32 0  32/2=16 0  16/2=8 0  8/2=4 0  4/2=2 0  2/2=1 0  1/2= 0 1  14(10 c/c) = 1110(2 c/c) = 1\*2^3+1\*2^2 +1\*2^1 +0\*2^0 = 14(10 c/c).  14/2=7 0  7/2=3 1  3/2=1 1  1/2=0 1 |
|  |  |

4. Выполнить задания из таблицы, представленной ниже, в соответствии с вариантом. Номер варианта определяет преподаватель.

| **№ вар.** | **Условие,решение** |
| --- | --- |
| **5** | Выполнить перевод чисел  11(8 c/c) →  001 001(2 c/c) → 9 (16 c/c);  0∙2^5+0∙2^4 + 1∙2^3+0∙2^2+0∙2^1+1∙2^0 =9  1111(2 c/c) →15 (10c/c);   1∙2^3+1∙2^2+1∙2^1+1∙2^0 =15  168(10 c/c) →1010 1000  (2 c/c);  168/2=84 0  84/2=42 0  42/2=21 0  21/2=10 1  10/2=5 0  5/2=2 1  2/2=1 0  1/2= 0 1  103(8 c/c) → 001 000 011 (2 c/c) →043(16 c/c);  0000 0100 0011  1110111(2 c/c) →  167(8c/c) →77(16c/c);  1)001 110 111 | 2)0111 0111  24(10 c/c) → 11000(2 c/c).  24/2=12 0  12/2=6 0  6/2=3 0  3/2 =1 1  1/2 = 0 1 |

**Лабораторная работа № 2. Способы представления алгоритмов**

Алгоритм– это система правил, определяющая последовательность действий над некоторыми объектами и приводящая к достижению поставленной цели после конечного числа шагов.

Для записи алгоритма решения задачи применяется:

− словесно-формульное описание;

− блок-схема (отдельные блоки представлены ниже).

|  |  |  |  |
| --- | --- | --- | --- |
| Блок начала-конца алгоритма | Блок ввода-вывода данных | Блок вычислений | Условный блок |
| Предопределенный процесс (использование готовых программ) | Без имени-3  Счетный цикл | Без имени-3  Комментарий | Соединитель (используется при разрыве схемы) |

Различают алгоритмы линейные, разветвляющиеся, циклические.

*Линейный* – это такой алгоритм, в котором все действия выполняются последовательно друг за другом и только один раз. Схема представляет собой последовательность блоков, которые располагаются сверху вниз в порядке их выполнения.

С помощью алгоритмов *разветвляющейся* структуры можно описать задачи, в которых выбор направления вычислений осуществляется по итогам проверки некоторого условия.

Для решения многих задач характерно многократное повторение отдельных участков вычислений. Для решения таких задач применяются алгоритмы *циклической* структуры.

|  |  |
| --- | --- |
| Словесно-формульное описание алгоритма решения квадратного уравнения | Блок-схема алгоритма решения квадратного уравнения |
| 1. Ввести числовые значения переменных **а**, **b** и **с**.  2. Вычислить **d** по формуле  **d = b² - 4ас**.  3. Если **d < 0**, то перейти к п. 4, иначе перейти к п. 5.  4. Напечатать сообщение «Корней нет» и перейти к п. 7  5. Вычислить:  Без имени-3  6. Напечатать значения **х1** и **х2**  7. Конец вычислений. |  |

**Задание**

1. В приложении Word разработать словесно-формульное описание и блок-схему алгоритма в соответствии со своим вариантом для решения задачи из таблицы. Номер варианта определяет преподаватель.

| **№ вар.** | **Условие задачи :** Даны действительные числа **x, y, z**. Поменять значения в переменных так, чтобы **x <= y <= z**. |
| --- | --- |
| 5 | нет  1.Начало  2. Ввод значения x,y,z  3.Условие: Если x>=y , то переходим к пункту 4. Если условие не выполняется , то переходим к 5 пункту схемы.  4. Поменять x и y .  5.Условие: Если y>=z , то переходим к пункту 6. Если условие не выполняется , то переходим к 7 пункту схемы  6. Поменять y и z.  7. Условие: Если x>=y , то переходим к пунку 8. Если условие не выполняется , то переходим к 9 пункту схемы.  8. Поменять x и y.  9.Вывод значений x,y,z.  10.Конец  нет  Поменять y и z  да  y >= z  Начало  да  Ввод x,y,z  Поменять x и y  x > =y |

Конец

нет

да

Вывести x, y, z

Поменять x и y

x > =y

2. В соответствии со своим вариантом из таблицы лабораторной работы № 5 разработать словесно-формульное описание и ***блок-схему*** алгоритма для решения задачи.

**Решение**

1. Словесно-формульное описание и блок-схема алгоритма в соответствии со своим вариантом для решения задачи из таблицы. Номер варианта определяет преподаватель.

ВАРИАНТ №5 **Определить, есть ли среди заданных целых чисел A, B, C, D хотя бы одно нечётное.**

|  |  |
| --- | --- |
| Словесно-формульное описание алгоритма решения квадратного уравнения | Блок-схема алгоритма решения квадратного уравнения |
| 1. Начало  2. Ввод значений A,B,C,D  3. Если число A нечётное - (то есть остаток от деления числа А на 2 , который соответсвенно не равен нулю)- мы пеходим к 4 пункту и цикл завершается. в ином случае число A чётное и цикл продалжается и переходим к 5 пункту.  4. Есть нечётное число переход к 11.  5. Если число B нечётное - (то есть остаток от деления числа B на 2 , который соответсвенно не равен нулю)- )- мы пеходим к 6 пункту и цикл завершается. в ином случае число A чётное и цикл продалжается и переходим к 7 пункту.  6. Есть нечётное число переход к 11.  7. Если число С нечётное - (то есть остаток от деления числа С на 2 , который соответсвенно не равен нулю)- мы пеходим к 8 пункту и цикл завершается. в ином случае число A чётное и цикл продалжается и переходим к 9 пункту.  8. Есть нечётное число переход к 11.  9. Если число D нечётное - (то есть остаток от деления числа D на 2 , который соответсвенно не равен нулю)- мы пеходим к 10 пункту и цикл завершается. в ином случае число A чётное и цикл продалжается и переходим к 11 пункту.  10. Есть нечётное число .  11.Конец | Начало  нет  да  B mod 2 ≠ 0  да  Ввод значений A, B, C, D  Есть нечётное число­­  A mod 2 ≠ 0  ↓  2  ↓  2  Есть нечётное число­­  нет |

Есть нечётное число­­

↓

2

↓

2

Есть нечётное число­­

↓

2

2

Конец

нет

нет

да

да

D mod 2 ≠ 0

C mod 2 ≠ 0

**Лабораторная работа № 3. Основные элементы языка С+**

**+.**

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Создать новый *проект* с именем, которое состоит из фамилии студента и номера лабораторной работы.  В проекте создать новый *файл* с программой под именем **main.cpp** | **/\* Пример 1 \*/**  **#include <iostream>**  **void main()**  **{**  **float x = 3, y;**  **y = x \* x + sin(x);**  **std::cout << y;**  **}** |
| 2. Ввести программу, текст которой приведен справа, проанализировать ее.  Выполнить программу, ознакомиться с результатом. |
| 3. В предыдущем проекте заменить текст программы на программу, записанную в правой части данного пункта, выполнить ее. | Исходные данные:  **#include <iostream>**  **void main()**  **{**  **double t, u, k = 4, a = 4.1, x = 5e-5;**  **t = 2 \* tan(k) / a + log(3 + x) + exp(x);**  **u = sqrt(t + 1) - sin(x) \* cos(t);**  **std::cout << "t=" << t;**  **std::cout << "u=" << u;**  **}**    **k = 4; a = 4.1;**  **x = 5∙10-5.**  Вычислить:  **t = 2tg(k)/a + ln(3 + x) + ex,**  **u = √(t + 1) − sin(x)cos(t)** |

4. В таблице ниже приведены условия задач. В соответствии со своим вариантом разработать программу для решения задачи. Опробовать работу программы и проанализировать результаты.

Возможны ситуации, когда исходные данные заданы некорректно и при выполнении программы возникают ошибки (деление на ноль, корень из отрицательного числа и т. п.). В таком случае надо выполнить программу с другими исходными данными.

5. К номеру своего варианта прибавить 3 и написать программу для новых исходных данных (для вариантов с 14 по 16 перейти к вариантам с 1 по 3).

| **№ вар** | **Формулы для вычислений** | **Исходные данные** | **№ вар** | **Формулы для вычислений** | **Исходные данные** |
| --- | --- | --- | --- | --- | --- |
| 5 |  |  | 8 |  |  |

**Решение**

| **№ вар** | **Код программы** | **Скриншот результата** | **№ вар** | **Код программы** | **Скриншот результата** |
| --- | --- | --- | --- | --- | --- |
| 5 | **#include <iostream>**  **void main()**  **{**  **double w,v,b=40,x=1.1,**  **a=5e-6;**    **w=(a+b)\* tan(x) /(x+1) ;**  **v=1/2 \*b- sqrt(w-a\*b)**  **std::cout << "w=" << w;**  **std::cout << "v=" << v;**  **}** | C:\Users\HP\Pictures\Screenshots\Снимок экрана (37).png | 8 | **#include <iostream>**  **void main()**  **{**  **double s, t, x = 0.1, j = 12, y = 5e-6;**  **s = 0.4\*x-1/j\* tan(y);**  **t = s - sin(s);**  **std::cout << "s=" << s;**  **std::cout << "t=" << t;**  **}** | C:\Users\HP\Pictures\Screenshots\Снимок экрана (39).png |

Начало

Начало

**№8**

**№5**

,b=40;

x=1.1;

a=5e-6.

x = 0.1;

j = 12;

y = 5e-6.

Конец

Конец

Вывод значений w и v

s = 0.4\*x-1/j\* tan(y);

t = s - sin(s).

Вывод значений w и v

w=(a+b)\* tan(x) /(x+1) ;

v=1/2 \*b- sqrt(w-a\*b)

| **№ вар** | **Код программы** | **Скриншот результата** | **№ вар** | **Код программы** | **Скриншот результата** |
| --- | --- | --- | --- | --- | --- |
| 16 | **#include <iostream>**  **void main()**  **{**  **double w,y,m=6,z=0.05e-5;**  **;**  **w=4\*z\*y-7e-2\*y;**  **y=cos(5\*m)/sin(sqr(0.4\*m));**  **std::cout << "w=" << w;**  **std::cout << "y=" << v;**  **}** | C:\Users\HP\Pictures\Screenshots\Снимок экрана (41).png | 3 | **#include <iostream>**  **void main()**  **{**  **double d,f,z=1.5e-6,i=6,x=4.5;**  **d=tg(-x\*1)/sqr(x-z);**  **f=sin(2\*d)/d;**  **std::cout << "d=" << d;**  **std::cout << "f=" << f;**  **}** | C:\Users\HP\Pictures\Screenshots\Снимок экрана (42).png |

Начало

Начало

**№16**

**№3**

I=-6;

,x=4.5;

z=1.5e-6.

m=6;

z=0.05e-5.



Конец

Конец

Вывод значений

d и f

Вывод значений

w и y.

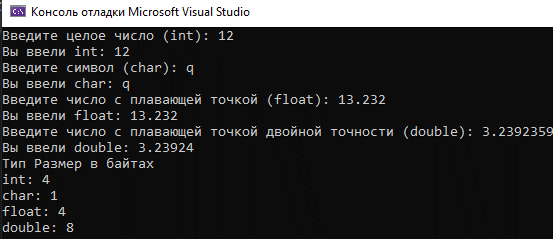
w=4\*z\*y-7e-2\*y;

y=cos(5\*m)/sin(sqr(0.4\*m)).

| **№ вар** | **Код программы** | **Скриншот результата** | **№ вар** | **Код программы** | **Скриншот результата** |
| --- | --- | --- | --- | --- | --- |
| 13 | **#include <iostream>**  **void main()**  **{**  **double**  **t,y,a=1.5,c= 9,d=0.5e-8;**  **t=(d\*c+a\*sqr(c-1));**  **y=0.5\*t/d+exp(a);**  **std::cout << "t=" << t;**  **std::cout << "y=" << y;**  **}** | C:\Users\HP\Pictures\Screenshots\Снимок экрана (45).png |  |  |  |

**Лабораторная работа № 4. Ввод и вывод информации**

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Выполнить программу, записанную справа, которая использует *потоковый* вывод данных. Проанализировать ее текст.  Добавить в программу ввод и вывод переменных различных типов. | #include <iostream>  void main()  {  setlocale(LC\_CTYPE, "Russian");  using namespace std;  int t;  cout << "Введите t=";  cin >> t;  cout << "t=" << t << endl;  cout << "Тип Размер в байтах" << endl;  cout << "int: " << sizeof(int) << endl;  cout << "char: " << sizeof(char)<< endl;  cout << "float: " << sizeof(float) << endl;  cout << "double: " << sizeof(double) << endl;  // sizeof определяет размер объекта в байтах    }    Скриншот результата |



#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

int t;

cout << "Введите целое число (int): ";

cin >> t;

cout << "Вы ввели int: " << t << endl;

char c;

cout << "Введите символ (char): ";

cin >> c;

cout << "Вы ввели char: " << c << endl;

float f;

cout << "Введите число с плавающей точкой (float): ";

cin >> f;

cout << "Вы ввели float: " << f << endl;

double d;

cout << "Введите число с плавающей точкой двойной точности (double): ";

cin >> d;

cout << "Вы ввели double: " << d << endl;

cout << "Тип Размер в байтах" << endl;

cout << "int: " << sizeof(int) << endl;

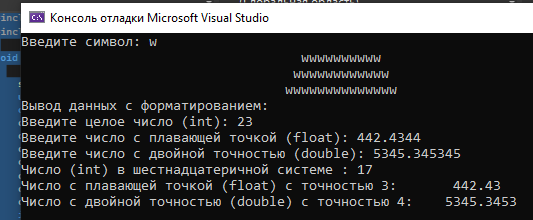
cout << "char: " << sizeof(char) << endl;

cout << "float: " << sizeof(float) << endl;

cout << "double: " << sizeof(double) << endl;

}

|  |  |
| --- | --- |
| 2. Опробовать работу программы, приведенной в правой части.  Добавить в программу ввод и вывод переменных с использованием манипуляторов. | #include <iostream>  #include <iomanip>  void main()  {  setlocale(LC\_CTYPE, "Russian");  using namespace std;  char c, probel; probel = ' ';  cout << "Введите символ: "; cin >> c;  cout << setw(35) << setfill(probel) << probel;  cout << setw(10) << setfill(c) << c << endl;  cout << setw(34) << setfill(probel) << probel;  cout << setw(12) << setfill(c) << c << endl;  cout << setw(33) << setfill(probel) << probel;  cout << setw(14) << setfill(c) << c << endl;  } |



#include <iostream>

#include <iomanip>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

char c, probel; probel = ' ';

cout << "Введите символ: "; cin >> c;

cout << setw(35) << setfill(probel) << probel;

cout << setw(10) << setfill(c) << c << endl;

cout << setw(34) << setfill(probel) << probel;

cout << setw(12) << setfill(c) << c << endl;

cout << setw(33) << setfill(probel) << probel;

cout << setw(14) << setfill(c) << c << endl;

int i;

float f;

double d;

cout << fixed; // Фиксированный формат вывода для чисел с плавающей точкой

cout << "Вывод данных с форматированием:\n";

cout << "Введите целое число (int): ";

cin >> i;

cout << "Введите число с плавающей точкой (float): ";

cin >> f;

cout << "Введите число с двойной точностью (double): ";

cin >> d;

cout << "Число (int) в шестнадцатеричной системе : "

<< setfill(probel) << hex << i << endl;

cout << "Число с плавающей точкой (float) с точностью 3: "

<< setw(12) << setprecision(2) << setfill(probel)<<f << endl;

cout << "Число с двойной точностью (double) с точностью 4: "

<< setprecision(4) << setw(12) << setfill(probel) << d << endl;

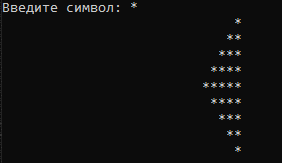
}

|  |  |  |  |
| --- | --- | --- | --- |
| 3. Выполнить программу, записанную в правой части, которая использует *форматированный* ввод-вывод данных.  Изменить программу так, чтобы выводилась своя фамилия, имя и отчество, факультет, номер группы русскими буквами.  При выводе использовать управляющие коды.  Внести изменения в программы лабораторной работы № 3 с тем, чтобы осуществлялся форматированный ввод и вывод данных. | #include <stdio.h>  #include <clocale>  #include <conio.h>  void main()  {  setlocale(LC\_CTYPE, "Russian");  printf("\n\t Фамилия: Езерский\n");  printf("\t Имя: Никита\n");  printf("\t Отчество: Витальевич\n");  printf("\t Факультет: Информационных технологий\n");  printf("\t Номер группы: десять\n");  \_getch();  }   |  |  | | --- | --- | | **#include <stdio.h>**  **#include <conio.h>**  **void main()**  **{**  **printf("\n\t Privet\n");**  **printf("\n... Press key");**  **\_getch();**  **}**    **Измененный код лаб.раб. №3(5)**  #include <iostream>  void main()  {  setlocale(LC\_CTYPE, "Russian");  using namespace std;  double b;  cout << "Введите чему равно число b: ";  cin >> b;  double x;  cout << "Введите чему равно число x: ";  cin >> x;  double a;  cout << "Введите чему равно число a: ";  cin >> a;  double w, v;  w = (a + b) \* tan(x) / (x + 1);  v = 1 / 2 \* b - sqrt(w - a \* b);  std::cout << "w=" << w<<endl;  std::cout << "v=" << v;  } | Заголовочный файл с именем **stdio.h** может использоваться при форматированном вводе-выводе данных.  Заголовочный файл **conio.h**. поддерживает функцию **\_getch()**, которая ожидает нажатия клавиши на клавиатуре.  Функция **printf()** печатает информацию. | |
| 4. Изучить *символьный* ввод и вывод данных. Написать программу, реализующую диалог, используя пример, записанный в правой части. | #include <windows.h>  #include <iostream>  void main()  {  SetConsoleOutputCP(1251);  SetConsoleCP(1251);  char name[60];  puts ( "Как вас зовут? ");  gets\_s(name);    }    **Код диалога**  void main()  {  SetConsoleOutputCP(1251);  SetConsoleCP(1251);  char name[60];  puts ( "Как вас зовут? ");  gets\_s(name);  std::cout << "Привет, " << name << "!" << std::endl;  char year[20];  puts("Сколько вам лет?");  gets\_s(year);  std::cout <<"Супер "<<name << " !!!" << year << " лет отличный возраст!" << std::endl;  char hob[60];  puts("Какое у вас хобби");  gets\_s(hob);  std::cout << hob <<" интерессное увлечение, " << name << ", сам бы таким с удовольствием занимался !!!" << std::endl;    } |

6. В соответствии со своим вариантом написать программы по условиям, приведенным в таблице ниже.

| **№ вар.** | **Условие задачи** |
| --- | --- |
| 5 | 1. Нарисовать треугольник по центру консольного окна, закрашенный введенным символом. |
| 2. Животновод в начале зимы повышает цену на молоко на **р**%, а летом снижает на столько же процентов. Изменится ли цена на молоко и если да, то в какую сторону и на сколько, через **n** лет? |

| **№ вар.** | **Решение задачи** | |
| --- | --- | --- |
| 5 | 1. Нарисовать треугольник по центру консольного окна, закрашенный введенным символом. | |
| **Код программы** | **Скриншот результата** |
|  |  |



#include <iostream>

#include <iomanip>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

char c, probel; probel = ' ';

cout << "Введите символ: "; cin >> c;

cout << setw(29) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(28) << setfill(probel) << probel;

cout << setw(2) << setfill(c) << c << endl;

cout << setw(27) << setfill(probel) << probel;

cout << setw(3) << setfill(c) << c << endl;

cout << setw(26) << setfill(probel) << probel;

cout << setw(4) << setfill(c) << c << endl;

cout << setw(25) << setfill(probel) << probel;

cout << setw(5) << setfill(c) << c << endl;

cout << setw(26) << setfill(probel) << probel;

cout << setw(4) << setfill(c) << c << endl;

cout << setw(27) << setfill(probel) << probel;

cout << setw(3) << setfill(c) << c << endl;

cout << setw(28) << setfill(probel) << probel;

cout << setw(2) << setfill(c) << c << endl;

cout << setw(29) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

}

| **№ вар.** | **Решение задачи** | |
| --- | --- | --- |
| 5 | 2. Животновод в начале зимы повышает цену на молоко на **р**%, а летом снижает на столько же процентов. Изменится ли цена на молоко и если да, то в какую сторону и на сколько, через **n** лет? | |
| **Код программы** | **Скриншот результата** |
| #include <iostream>  #include <iomanip>  void main() {  setlocale(LC\_CTYPE, "Russian");  double x, y, z;  int years;  // Ввод начальной цены молока  std::cout << "Введите начальную цену молока: ";  std::cin >> x;  // Ввод процента изменения цены  std::cout << "Введите процент изменения цены: ";  std::cin >> y;  // Ввод количества лет  std::cout << "Введите количество лет: ";  std::cin >> years;  // Рассчитываем итоговую цену через n лет  z = x;  for (int i = 0; i < years; ++i) {  // Увеличение цены на зимний период  z += z \* (y / 100);  // Уменьшение цены на летний период  z -= z \* (y / 100);  }  // Форматированный вывод  std::cout << std::fixed << std::setprecision(2); // Устанавливаем фиксированный формат и 2 знака после запятой  std::cout << "Итоговая цена на молоко через " << years << " лет: " << z << " руб." << std::endl;  } |  |

|  |  |
| --- | --- |
| 6 | 1. Нарисовать звезду, закрашенную введенным символом. |
| 2. Определить расстояние на плоскости между двумя точками с заданными координатами **M1**(**x1,y1**) и **M2**(**x2,y2**). |

#include <iostream>

#include <iomanip>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

char c, probel; probel = ' ';

cout << "Введите символ: "; cin >> c;

cout << setw(35) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(34) << setfill(probel) << probel;

cout << setw(3) << setfill(c) << c << endl;

cout << setw(33) << setfill(probel) << probel;

cout << setw(5) << setfill(c) << c << endl;

cout << setw(32) << setfill(probel) << probel;

cout << setw(7) << setfill(c) << c << endl;

cout << setw(31) << setfill(probel) << probel;

cout << setw(9) << setfill(c) << c << endl;

cout << setw(30) << setfill(probel) << probel;

cout << setw(11) << setfill(c) << c << endl;

cout << setw(26) << setfill(probel) << probel;

cout << setw(19) << setfill(c) << c << endl;

cout << setw(25) << setfill(probel) << probel;

cout << setw(21) << setfill(c) << c << endl;

cout << setw(23) << setfill(probel) << probel;

cout << setw(25) << setfill(c) << c << endl;

cout << setw(21) << setfill(probel) << probel;

cout << setw(29) << setfill(c) << c << endl;

cout << setw(23) << setfill(probel) << probel;

cout << setw(25) << setfill(c) << c << endl;

cout << setw(25) << setfill(probel) << probel;

cout << setw(21) << setfill(c) << c << endl;

cout << setw(26) << setfill(probel) << probel;

cout << setw(19) << setfill(c) << c << endl;

cout << setw(30) << setfill(probel) << probel;

cout << setw(11) << setfill(c) << c << endl;

cout << setw(31) << setfill(probel) << probel;

cout << setw(9) << setfill(c) << c << endl;

cout << setw(32) << setfill(probel) << probel;

cout << setw(7) << setfill(c) << c << endl;

cout << setw(33) << setfill(probel) << probel;

cout << setw(5) << setfill(c) << c << endl;

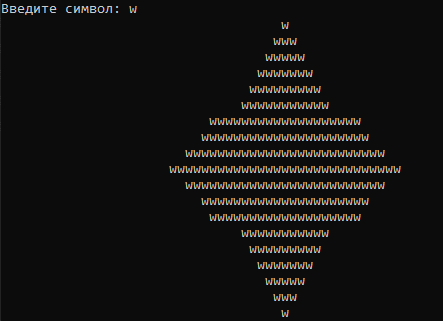
cout << setw(34) << setfill(probel) << probel;

cout << setw(3) << setfill(c) << c << endl;

cout << setw(35) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

}



#include <iostream>

#include <iomanip>

void main()

{

char c, probel; probel = ' ';

cout << "Введите символ: "; cin >> c;

cout << setw(35) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(34) << setfill(probel) << probel;

cout << setw(3) << setfill(c) << c << endl;

cout << setw(33) << setfill(probel) << probel;

cout << setw(5) << setfill(c) << c << endl;

cout << setw(31) << setfill(probel) << probel;

cout << setw(9) << setfill(c) << c << endl;

cout << setw(29) << setfill(probel) << probel;

cout << setw(13) << setfill(c) << c << endl;

cout << setw(27) << setfill(probel) << probel;

cout << setw(17) << setfill(c) << c << endl;

cout << setw(29) << setfill(probel) << probel;

cout << setw(13) << setfill(c) << c << endl;

cout << setw(31) << setfill(probel) << probel;

cout << setw(9) << setfill(c) << c << endl;

cout << setw(33) << setfill(probel) << probel;

cout << setw(5) << setfill(c) << c << endl;

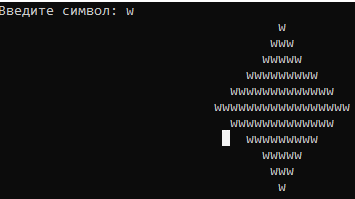
cout << setw(34) << setfill(probel) << probel;

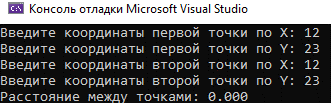
cout << setw(3) << setfill(c) << c << endl;

cout << setw(35) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

}





#include <iostream>

#include <cmath>

#include <iomanip>

void main() {

setlocale(LC\_CTYPE, "Russian");

using namespace std;

double x1, y1, x2, y2;

std::cout << "Введите координаты первой точки по X: ";

std::cin >> x1;

std::cout << "Введите координаты первой точки по Y: ";

std::cin >> y1;

std::cout << "Введите координаты второй точки по X: ";

std::cin >> x2;

std::cout << "Введите координаты второй точки по Y: ";

std::cin >> y2;

// Вычисление расстояния между точками

double distance = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));

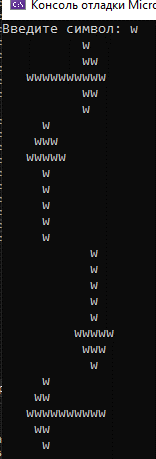
std::cout << std::fixed << std::setprecision(3);

std::cout << "Расстояние между точками: " << distance << std::endl;

}

fffhfff

|  |  |
| --- | --- |
| 9 | 1. Нарисовать разнонаправленные стрелки (→ ↑ ↓ ←), состоящие из введенного символа. |
| 2. Дано действительное число **а**. Не пользуясь никакими другими операциями, кроме умножения, получить **а4** за две операции. |



#include <iostream>

#include <iomanip>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

char c, probel; probel = ' ';

cout << "Введите символ: "; cin >> c;

cout << setw(10) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(10) << setfill(probel) << probel;

cout << setw(2) << setfill(c) << c << endl;

cout << setw(3) << setfill(probel) << probel;

cout << setw(10) << setfill(c) << c << endl;

cout << setw(10) << setfill(probel) << probel;

cout << setw(2) << setfill(c) << c << endl;

cout << setw(10) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl ;

cout << setw(5) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(4) << setfill(probel) << probel;

cout << setw(3) << setfill(c) << c << endl;

cout << setw(3) << setfill(probel) << probel;

cout << setw(5) << setfill(c) << c << endl;

cout << setw(5) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(5) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(5) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(5) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(5) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(11) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(11) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(11) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(11) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(11) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(9) << setfill(probel) << probel;

cout << setw(5) << setfill(c) << c << endl;

cout << setw(10) << setfill(probel) << probel;

cout << setw(3) << setfill(c) << c << endl;

cout << setw(11) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(5) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

cout << setw(4) << setfill(probel) << probel;

cout << setw(2) << setfill(c) << c << endl;

cout << setw(3) << setfill(probel) << probel;

cout << setw(10) << setfill(c) << c << endl;

cout << setw(4) << setfill(probel) << probel;

cout << setw(2) << setfill(c) << c << endl;

cout << setw(5) << setfill(probel) << probel;

cout << setw(1) << setfill(c) << c << endl;

}



#include <iostream>

#include <iomanip>

void main() {

setlocale(LC\_CTYPE, "Russian");

using namespace std;

double a;

// Ввод числа a

std::cout << "Введите число a: ";

std::cin >> a;

// Первая операция: a^2

double a2 = a \* a;

// Вторая операция: (a^2)^2 = a^4

double a4 = a2 \* a2;

std::cout << std::fixed << std::setprecision(3);

std::cout << "a^4 = " << a4 << std::endl;

}

|  |  |
| --- | --- |
| 4 | 1. Нарисовать круг по центру консольного окна, закрашенный введенным символом. |
| 2. Даны числа **X, Y, Z**. Определить среднее арифметическое и среднее геометрическое их модулей. |

#include <iostream>

#include <iomanip>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

char c, probel; probel = ' ';

cout << "Введите символ: "; cin >> c;

cout << setw(9) << setfill(probel) << probel;

cout << setw(6) << setfill(c) << c << endl;

cout << setw(8) << setfill(probel) << probel;

cout << setw(8) << setfill(c) << c << endl;

cout << setw(7) << setfill(probel) << probel;

cout << setw(10) << setfill(c) << c << endl;

cout << setw(7) << setfill(probel) << probel;

cout << setw(10) << setfill(c) << c << endl;

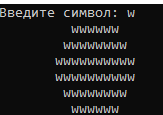
cout << setw(8) << setfill(probel) << probel;

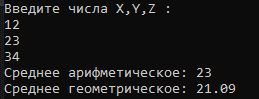
cout << setw(8) << setfill(c) << c << endl;

cout << setw(9) << setfill(probel) << probel;

cout << setw(6) << setfill(c) << c << endl;

}





#include <iostream>

#include <iomanip>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

double a, b,c, arifm, geom;

cout << fixed;

cout << "Введите числа X,Y,Z :" << endl;

cin >> a >> b >>c;

arifm = (a + b+c) / 3;

geom = std::cbrt(a \* b \* c);

cout << "Среднее арифметическое: " << setprecision(0) << arifm << endl;

cout << "Среднее геометрическое: "<< setprecision(2) << geom<<endl;

}

**Лабораторная работа № 5. Разветвляющиеся программы**

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Выполнить программу, приведенную в правой части, несколько раз с различными значениями переменной **j**.  Переделать программу с тем, чтобы ввод и вывод осуществлялся с помощью *потоковых* операторов ввода-вывода. | #include <iostream>  using namespace std;  int main()  {  setlocale(LC\_CTYPE, "Russian");  double b, z, x = 4.e-3, s = 1.1, j;  cout << "Введите j ";  cin >> j;  b = s + (5 \* x + j);  z = b < 1.5 ? sqrt(x + 0.3 \* j) + b : abs(x \* j + b);  cout << "j=" << j << " z=" << z << endl;  return 0;  } |
| 2. Выполнить программу, приведенную в правой части.  Изменить программу с тем, чтобы ввод и вывод осуществлялся с помощью *потоковых* операторов ввода-вывода. | #include <iostream>  using namespace std;  int main()  {  setlocale(LC\_CTYPE, "Russian");  float x, y, z, mn, mx;  cout << "Введите x=";  cin >> x;  cout << "Введите y=";  cin >> y;  cout << "Введите z=";  cin >> z;  if ((x + y + z) < (x \* y \* z))  mx = (x \* y \* z);  else  mx = (x + y + z);  mn = x;  if (mn > y) mn = y;  if (mn > z) mn = z;  cout << "max(x + y + z, xyz) \* min(x, y, z)= " << mx\*mn;  return 0;  } |

|  |  |
| --- | --- |
| 3. Выполнить программу с использованием оператора выбора **switch**, приведенную в правой части,  Записать условие задачи.  Условие задачи:  1)Спросить: Хотите ли вы приобрести автомобиль?,  2)если да, то спросить какой именно и в зависимости от модели дать оценку ему.  3)Если нет, то сообщить что и без автомобиля хорошо.  4)В случае неверного ввода вывести соответствующее сообщение. | #include <iostream>  using namespace std;  int main()  {  setlocale(LC\_CTYPE, "Russian");  int k;  puts("вы хотите приобрести автомобиль?(1-да, 2-нет)");  std::cin >> k;  switch (k)  {  case 1: {  puts("Какой? (1-Майбах, 2-Тойота, 3-Лада)");  std::cin >> k;  switch (k)  {  case 1: puts("Не слишком дорого?"); break;  case 2: puts("Хороший выбор"); break;  case 3: puts("Сомнительный выбор"); break;  }  break;  }  case 2: puts("Полезно ходить пешком"); break;  default: puts("Некорректный вариает"); break;  }  return 0;  } |

**4. Написать и выполнить программу по первой блок-схеме лабораторной работы № 2.**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE,"Russian");

double a, b, c, d, x1, x2;

cout << fixed;

cout<< " Введите а:";

cin >> a;

cout<< " Введите b:";

cin >> b;

cout<< " Введите с:";

cin >> c;

d = b \* b - 4 \* a \* c;

if (d < 0)

cout<< "Корней нет" ;

else

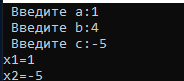
x1 = (-b + sqrt(d)) / (2 \* a);

x2 = (-b - sqrt(d)) / (2 \* a);

cout << "x1="<< x1 << endl << "x2=" << x2 << endl;

return 0;

}



**5. В соответствии со своим вариантом написать программу по условию, приведенному в таблице ниже. Использовать операторы *if* и (или) *тернарные* операторы.**

|  |  |
| --- | --- |
| Вариант 5 | Определить, есть ли среди заданных целых чисел **A**, **B**, **C**, **D** хотя бы одно нечётное. |

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

int A, B, C, D;

// Ввод данных

cout << "Введите четыре числа: ";

cin >> A >> B >> C >> D;

// Проверка на наличие хотя бы одного нечетного числа

if (A % 2 != 0 || B % 2 != 0 || C % 2 != 0 || D % 2 != 0) {

cout << "Среди чисел есть хотя бы одно нечетное." << endl;

}

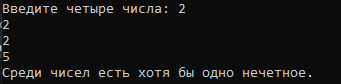
else {

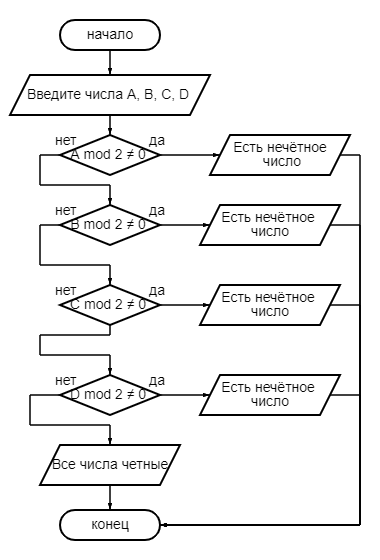
cout << "Все числа четные." << endl;

}

return 0;

}





**6. Написать программу, реализующую диалог на любую тему с использованием оператора switch.**

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "Russian");

int k;

puts("Ведёте ли вы здоровый образ жизни? (1-да, 2-нет)");

std::cin >> k;

switch (k)

{

case 1: {

puts("Чем вы предпочитаете заниматься на выходных?(1-лежать на диване, 2-провести время с друзьями, 3-позаниматься физкультурой)");

std::cin >> k;

switch (k)

{

case 1: puts("Это скучное время препровождение, советую позаниматься чем-то ещё "); break;

case 2: puts("Хороший выбор, с друзьями всегда весело и время пролетает незаметно "); break;

case 3: puts("Это хороший выбор, после спорта посоветовал бы встретиться с друзьями"); break;

}

break;

}

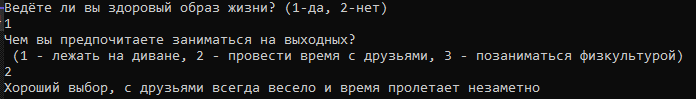
case 2: puts("Очень зря"); break;

default: puts("Некорректный вариает"); break;

}

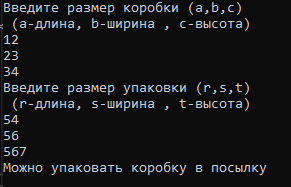
return 0;

}



**7. Дополнительные задачи (3 на выбор)**

5. Можно ли коробку размером a х b х c упаковать в посылку размером r х s х t?



#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

int a, b, c, r, s, t;

cout << "Введите размер коробки (a,b,c)\n (a-длина, b-ширина , c-высота)" << endl;

cin >> a;

cin >> b;

cin >> c;

cout << "Введите размер упаковки (r,s,t)\n (r-длина, s-ширина , t-высота)" << endl;

cin >> r;

cin >> s;

cin >> t;

if (r > a ,s>b , t>c ) {

cout << "Можно упаковать коробку в посылку" << endl;

}

else {

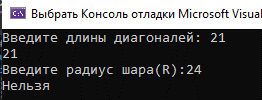
cout << "Нельзя упаковать коробку в посылку " << endl;

}

return 0;

}

4. Пройдет ли шар радиуса r через ромбообразное отверстие с диагоналями p и q?



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "RU");

double p, q, r, R;

cout << "Введите длины диагоналей: ";

cin >> p;

cin >> q;

cout << "Введите радиус шара(R):";

cin >> R;

r = (p \* q) / (2 \* sqrt(pow(p, 2) + pow(q, 2)));

if (R <= r) {

cout << "Можно " << endl;

}

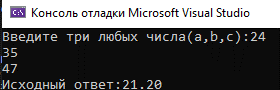
else {

cout << "Нельзя " << endl;

}

}

1. Введены с клавиатуры три числа. Найти сумму тех чисел, которые делятся на 5. Если таких чисел нет, то вывести текст «Error». Использовать только тернарные операторы.



#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

double a,b,c,s;

cout << fixed;

cout << "Введите три любых числа(a,b,c):" ;

cin >> a;

cin >> b;

cin >> c;

s = (a + b + c) / 5;

if ( s>0) {

cout <<"Исходный ответ:"<< setprecision(2) << s << endl;

}

else {

cout << "<<Error>> " << endl;

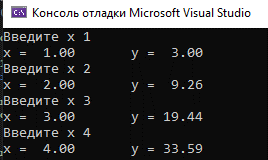
}

return 0;

}

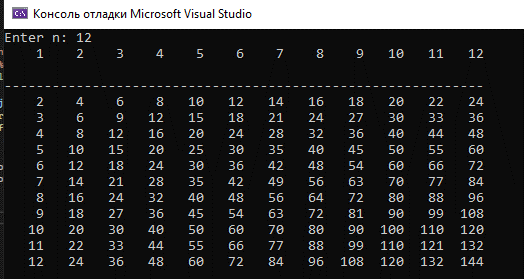
**Лабораторная работа № 6. Циклические программы**

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Выполнить программу с использованием оператора **for**, записанную в правой части.  Изучить блок-схему алгоритма. | Пример. Вычислить значение **y** по формулам:  ,  где **x = {4; 3,5; 7; 1}**.  В программе, представленной ниже, тело цикла **for** заключено в фигурные скобки. В переменной **n** находится номер значения **х** из массива. Вычисляется выражение **n = 0** и проверяется условие **n < 4**. Если оно истинно, то выполняются операторы цикла.  Затем вычисляется **n = n + 1**, вновь проверяется условие **n < 4** и, если оно истинно, то вновь выполняются операторы цикла.  Так продолжается до тех пор, пока **n** меньше 4. Если это условие не выполнится, то управление передается оператору, следующему за телом цикла.  В переменной **sd** вычисляется выражение (1 / 3), которое используется при вычислении корня третьей степени из **х**.  Результаты выводятся в виде действительных чисел, занимают 5 позиций, из которых 2 отводятся для записи дробной части.   |  |  | | --- | --- | | Блок-схема алгоритма | Программа | |  | **#include <iostream>**  **void main()**  **{**  **setlocale(LC\_CTYPE, "Russian");**  **float z, y, x, sd;**  **sd = 1.0 / 3.0;**  **for (int n = 0; n < 4; n++)**  **{**  **printf("Введите x ");**  **scanf\_s("%f", &x);**  **z = 2 \* pow(x, 2);**  **y = z + pow(x, sd);**  **printf("x = %5.2f\t", x);**  **printf("y = %5.2f\n", y);**  **}**  **}** | |



|  |  |
| --- | --- |
| 2. Выполнить программу с использованием оператора **while**, записанную в правой части.  Опробовать способ прерывания работы программы с помощью клавиш **Ctrl − Break**. | Пример. Вычислить значение **y** по формулам: , где **x** = 3(0,1)4 − меняется от 3 до 4 с шагом 0,1.  В программе тело цикла **while** заключено в фигурные скобки.  Если условие в скобках (**x < 4.1**) истинно, то выполняются операторы цикла до тех пор, пока оно не станет ложным. Тогда управление передается оператору, следующему за телом цикла.  Если в программе будет отсутствовать оператор **x = x + 0.1**, то программа зациклится. Надо **прервать** выполнение программы, нажав **Ctrl − Break**, и исправить ошибку.  Запись **(float)1 / 3** −это приведение типов. Тип данных в скобках указывает, в какой тип необходимо преобразовать операнд справа.  **#include <iostream>**  **using namespace std;**  **void main()**  **{**  **float z, y, x = 3;**  **while (x < 4.1)**  **{**  **z = 2 \* pow(x, 2);**  **y = z + pow(x, (float)1 / 3);**  **cout << "x=" << x << "\t";**  **cout << " y=" << y << endl;**  **x = x + 0.1;**  **}**  **}** |
|  | #include <iostream>  using namespace std;  void main()  {  float z, y, x = 3;  while (x < 4.1)  {  z = 2 \* pow(x, 2);  y = z + pow(x, (float)1 / 3);  cout << "x=" << x << "\t";  cout << " y=" << y << endl;    }  }  **прервал** выполнение программы, нажав **Ctrl − Break**, и исправил ошибку. |
| 3. Выполнить программу с использованием оператора do while, записанную в правой части.  Изучить блок-схему алгоритма. | **Пример. Вычислить значение y по формулам: ,**  **где x = 3(0,1)4 − меняется от 3 до 4 с шагом 0,1.**  **Оператор цикла do while называется оператором цикла с постусловием.**  **Сначала выполняется тело цикла, затем проверяется условие, записанное в скобках**  **(x < 4.1). Если оно истинно, то выполняются операторы цикла до тех пор, пока оно не станет ложным. Тогда управление передается оператору, следующему за телом**  **цикла.**   |  |  | | --- | --- | | Блок-схема алгоритма | Программа | | Без имени-1 | **#include <stdio.h>**  **#include <cmath>**  **void main()**  **{**  **float z, y, x = 3;**  **do**  **{**  **z = 2 \* pow(x, 2);**  **y = z + pow(x, (float)1/3);**  **printf("x=%5.2f\t", x);**  **printf("y=%5.2f\n", y);**  **x = x + 0.1;**  **}**  **while (x < 4.1);**  **}** | |

|  |  |
| --- | --- |
| 4. Выполнить программу, содержащую вложенный цикл. Записать условие задачи.  Оформить вывод результатов, используя различные возможности операторов вывода. | Вместо **#include <stdio.h>** в заголовочной части записать:  **#include <iostream>**  **#include <iomanip>**  **using namespace std;**    Перед **printf("\n");** разместить операторы:    **if (i == 1)**  **{ cout << endl;**  **cout << setw(5\*n) << setfill('-') <<**  **'-'<<endl;**  **}**  **#include <stdio.h>**  **void main()**  **{**  **int n, i, j;**  **printf("Enter n: ");**  **scanf\_s("%d", &n);**  **for (i = 1; i <= n; i++)**  **{**  **for (j = 1; j <= n; j++)**  **printf("%5d", i \* j);**  **printf("\n");**  **}**  **}**  Пояснить результат**. (Условие задачи)**  Программа выводит сообщение пользователю **Enter n: ,в которой пользователь указывает n-ое число.** Затем образуется n-ое кол-во строк и столбцов. Соответственно столб начинается с единицы, все последующие значения (первого)столбца = 1+1, а значения в ряде увеличиваются на n значений от n числа стоящего в ряду. |



#include <iostream>

#include <iomanip>

using namespace std;

void main()

{

int n, i, j;

printf("Enter n: ");

scanf\_s("%d", &n);

for (i = 1; i <= n; i++)

{

for (j = 1; j <= n; j++)

printf("%5d", i \* j);

printf("\n");

if (i == 1)

{

cout << endl;

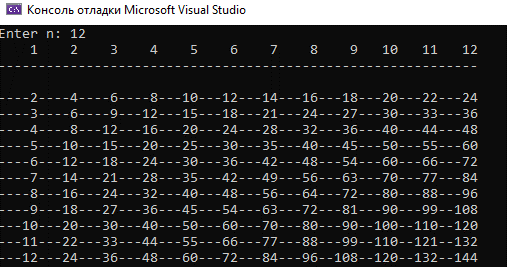
cout << setw(5 \* n) << setfill('-') <<

'-' << endl;

}

}

}



#include <stdio.h>

#include <iostream>

#include <iomanip>

using namespace std;

void main()

{

int n, i, j;

printf("Enter n: ");

scanf\_s("%d", &n);

for (i = 1; i <= n; i++)

{

for (j = 1; j <= n; j++)

cout << setw(5) << i \* j;

if (i == 1)

{

cout << endl;

cout << setw(5 \* n) << setfill('-') <<

'-' << endl;

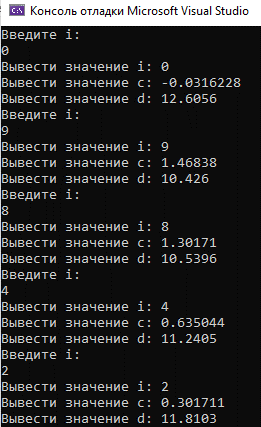
}

cout << "\n";

}

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 вариант |  |  | i = 0(0,5)3 | x = 1(0,1)2,  i = {1,3; −8; 0,2} |



**Исходные данные**

#include <iostream>

#include <cmath>

using namespace std;

void main()

{

setlocale(LC\_ALL, "Russian");

float y, x, k ,i ,c , d;

y = 1.6e-4;

x = -1;

k = 6;

for (int n = 0; n < 5; n++)

{

cout << "Введите i: " << endl;

cin >> i;

c = i / k - sqrt(y) / 0.4;

d = exp(1 - c) + 4.9 \* (pow(x, 2) + 1);

cout << "Вывести значение i: " << i << endl;

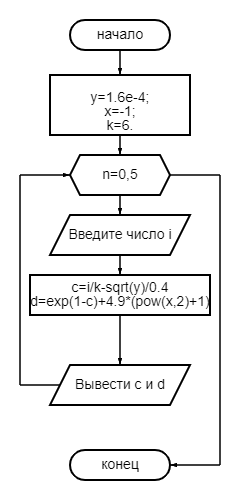
cout << "Вывести значение c: " << c << endl;

cout << "Вывести значение d: " << d << endl;

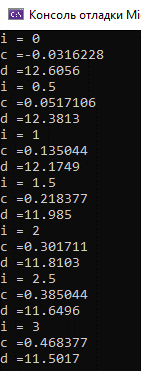
}

}

**(for):**



**While:**



#include <iostream>

#include <cmath>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

float c, d,y = 1.6e-4, x = -1, k = 6, i = 0;

while ( i<3.1)

{

c = i / k - sqrt(y) / 0.4;

d = exp(1 - c) + 4.9 \* (pow(x, 2) + 1);

cout << "i = " << i << endl;

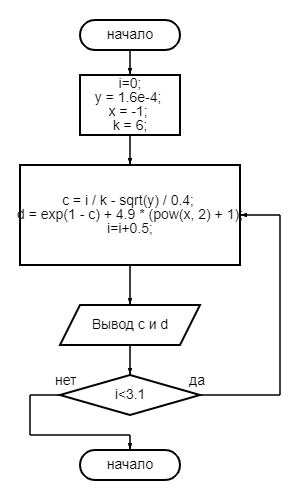
cout<<"c ="<< c<< endl;

cout<<"d ="<< d<<endl;

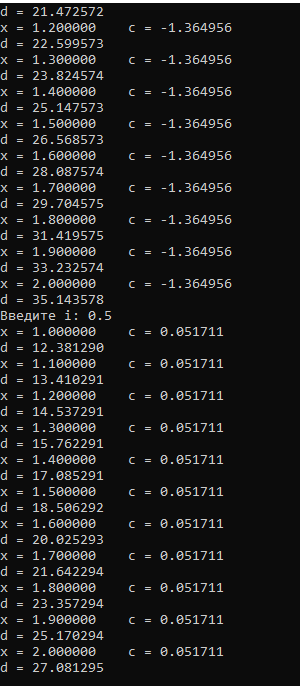
i = i + 0.5;

}

}



**Двойной цикл:**



#include <iostream>

#include <iomanip>

#include <cmath>

void main()

{

setlocale(LC\_CTYPE, "Russian");

float i, c, d,x=1, y = 1.6e-4, k = 6;

for (int n = 0; n < 3; n++)

{

printf("Введите i: ");

scanf\_s("%f", &i);

while (x < 2.1)

{

c = i / k - sqrt(y) / 0.4;

d = exp(1 - c) + 4.9 \* (pow(x, 2) + 1);

printf("x = %f\t", x);

printf("c = %f\n", c);

printf("d = %f\n", d);

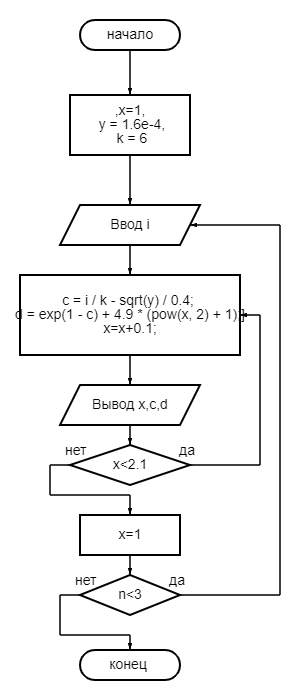
x = x + 0.1;

}

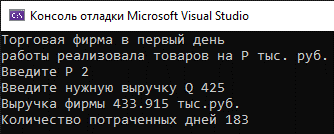
x = 1;

}

}



**Дополнительные задания:**



#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

float P, Q, days = 1, m;

printf("Торговая фирма в первый день \nработы реализовала товаров на P тыс. руб.\nВведите Р ");

scanf\_s("%f", &P);

printf("Введите нужную выручку Q ");

scanf\_s("%f", &Q);

m = P \* 1, 03;

do

{

m = m \* 1.03;

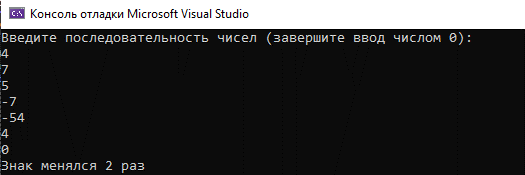
days = days + 1;

} while (m < Q);

printf("Выручка фирмы %.3f тыс.руб.\n", m);

printf("Количество потраченных дней %.0f\n", days);

}



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int n, pred = 0, znak = 0;

cout << "Введите последовательность чисел (завершите ввод числом 0):" << endl;

cin >> n;

while (n != 0) {

if ((pred != 0 && (pred > 0 && n < 0) || (pred < 0 && n>0)))

{

znak++;

}

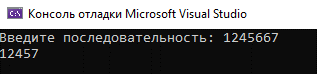
pred = n;

cin >> n;

}

cout << "Знак менялся "<< znak << " раз" << endl;

}



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int number, result = 0, a = 1, digit;

cout << "Введите последовательность: ";

cin >> number;

while (number != 0) {

digit = number % 10;

if (digit != 3 && digit != 6) {

result += digit \* a;

a \*= 10;

}

number /= 10;

}

cout << result << endl;

}

**Лабораторная работа № 7. Отладка программ**

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Записать условие задачи для программы, представленной в правой части.  Выполнить прокрутку для одной из программ лабораторной работы № 6. | Термином *прокрутка* обозначают выполнение программы вручную так, как если бы она выполнялась на компьютере.  **#include <stdio.h>**  **void main()**  **{ float y, x = 0;**  **while (x < 7)**  **{ if (x < 4)**  **y = x;**  **else**  **if (x > 4)**  **y = x \* x;**  **else**  **y = 1;**  **printf("x=%f\t", x);**  **printf("y=%5.2f\n", y);**  **x = x + 2;**  **}**  **}**  Запись прокрутки для данного примера может выглядеть следующим образом:  x = 0;  0 <= 6; 0 < 4; y = x = 0; вывод x = 0 и y = 0;  x = x + 2 = 0 + 2 = 2;  2 <= 6; 2 < 4; y = x = 2; вывод x = 2 и y = 2;  x = x + 2 = 2 + 2 = 4;  4 <= 6; 4 = 4; y = 1; вывод x = 4 и y = 1;  x = x + 2 = 4 + 2 = 6;  6 = 6; 6 > 4; y = x ∙ x = 6 ∙ 6 = 36; вывод x = 6 и y = 36;  x = x + 2= 6 + 2 = 8;  8 > 7. Выполнение программы закончено. |

x=3;

(3<4.1);

Вывод x = 3 и y= 19.4422;

x=x+0.1=3+0.1=3.1;

x=3.1;

(3.1<4.1);

Вывод x = 3.1 и y= ;

x=x+0.1=3.1+0.1=3.2;

x=3.2;

(3.2<4.1);

Вывод x = 3.2 и y=;

x=x+0.1=3.2+0.1=3.3;

x=3.3;

(3.3<4.1);

Вывод x = 3.3 и y=;

x=x+0.1=3.3+0.1=3.4;

x=3.4;

(3.4<4.1);

Вывод x = 3.4 и y= ;

x=x+0.1=3.4+0.1=3.5;

x=3.5;

(3.5<4.1);

Вывод x = 3.5 и y= ;

x=x+0.1=3.5+0.1=3.6;

x=3.6;

(3.6<4.1);

Вывод x = 3.6 и y= ;

x=x+0.1=3.6+0.1=3.7;

x=3.7;

(3.7<4.1);

Вывод x = 3.7 и y= ;

x=x+0.1=3.7+0.1=3.8;

x=3.8;

(3.8<4.1);

Вывод x = 3.8 и y= ;

x=x+0.1=3.8+0.1=3.9;

x=3.9;

(3.9<4.1);

Вывод x = 3.9 и y= ;

x=x+0.1=3.9+0.1=4.0;

x=4.0;

(4.0<4.1);

Вывод x = 4 и y= ;

x=x+0.1=4+0.1=4.1;

x=4.1;

(4.1<4.1);

Вывод x =4.1 и y= ;

x=x+0.1=4.1+0.1=4.2;

x=4.2;

(4.2<4.1);

Выполнение программы закончено

#include <iostream>

using namespace std;

void main()

{

float z, y, x = 3;

while (x < 4.1)

{

z = 2 \* pow(x, 2);

y = z + pow(x, (float)1 / 3);

cout << "x=" << x << "\t";

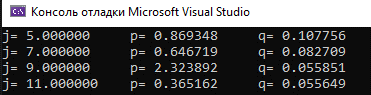
cout << " y=" << y << endl;

x = x + 0.1;

}

}

| **№ Варианта** | **Формулы для вычислений** | **Исходные данные** |
| --- | --- | --- |
| 5 |  |  |



#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

float y = 1.4e-3, x = 0.2, m = 4, j = 5, p,q;

while (j <= 11){

p = exp(sin(j / x)) \* log(x / y) \* x;

if (p <= y \* y) {

q = sqrt(p / m);

}

else if (p > y \* y) {

q = sqrt(2 \* x) / (j + p);

}

printf("j= %f\t", j);

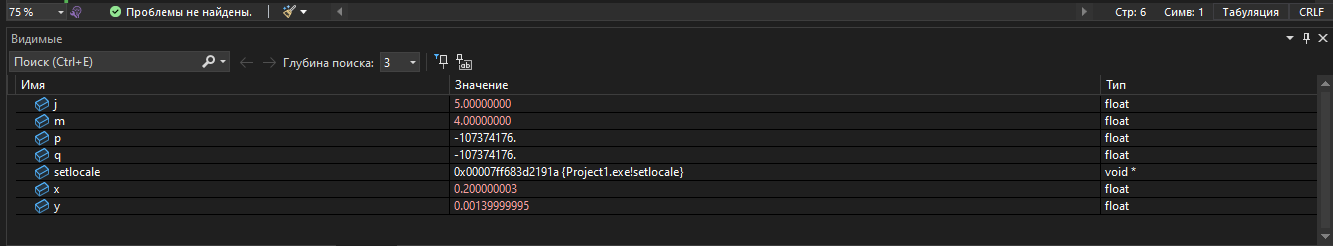
printf("p= %f\t", p);

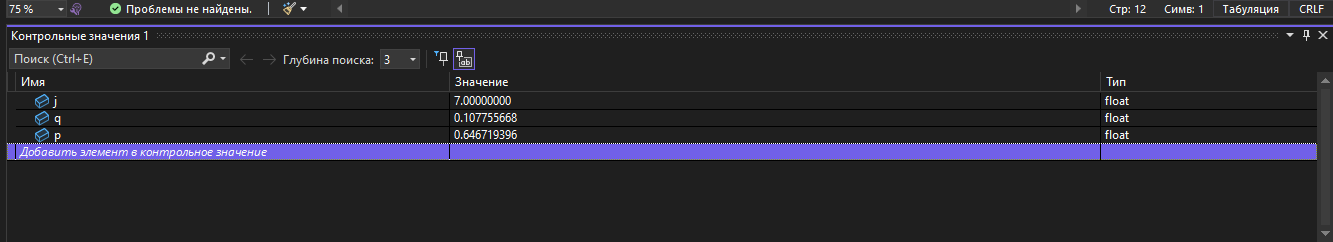
printf("q= %f\n", q);

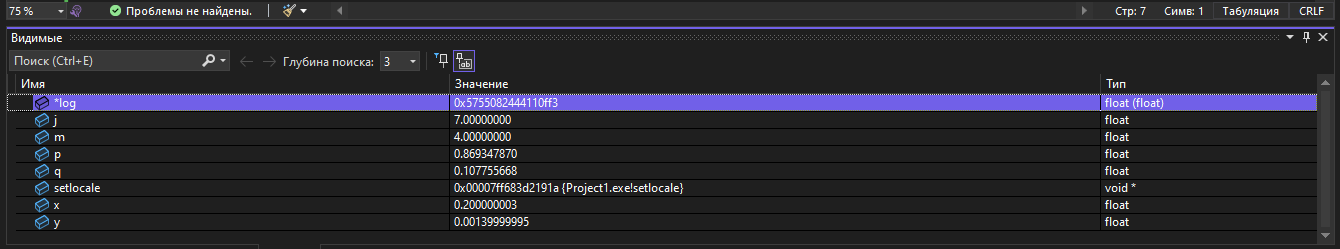
j=j + 2;

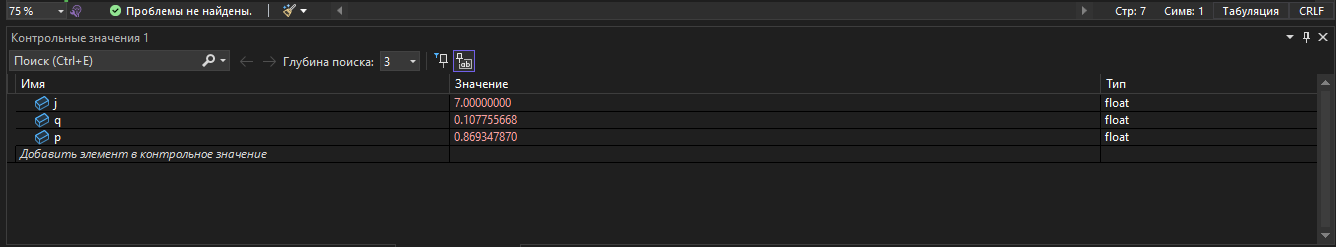
}

}





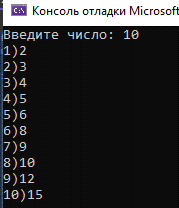




Задание 7

Построить первые **N** натуральных чисел, делителями которых являются только числа 2, 3 и 5.

5



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

int N, found = 0, number\_n = 2;

cout << "Введите число: " ;

cin >> N;

while (found < N) {

int test = number\_n;

while (test % 2 == 0) test /= 2;

while (test % 3 == 0) test /= 3;

while (test % 5 == 0) test /= 5;

if (test == 1) {

found++;

cout<< found << ")" << number\_n << endl;

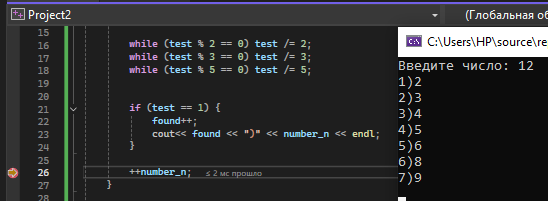
}

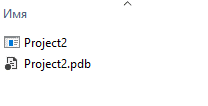
++number\_n;

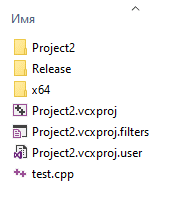
}

return 0;

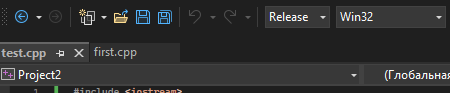
}

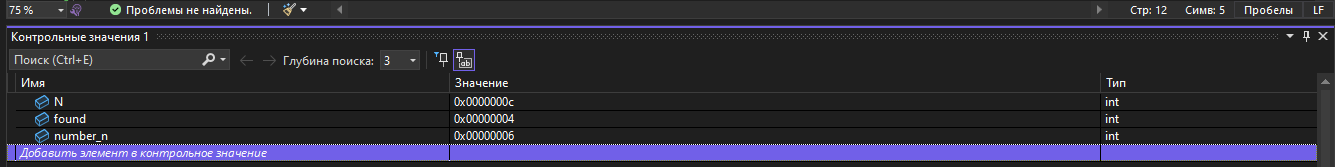


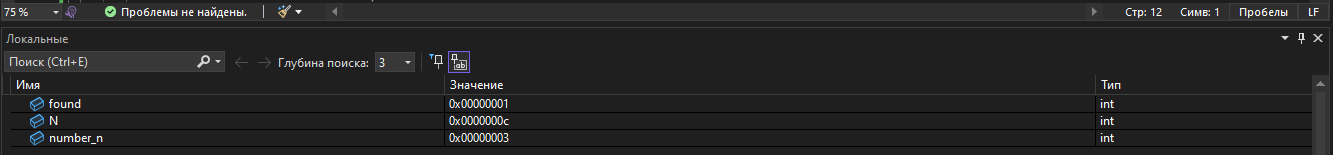




**Исполняемый файл(Release)**







Дополнительное задание

1

Найти натуральное число, состоящее из трёх цифр, с возрастающими слева направо цифрами, являющееся полным квадратом. Число является полным квадратом, если квадратный корень из него – простое число (число 121 – полный квадрат, т. к. 121=11 ∙ 11, а 11 – простое число).



#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

int n=10,a , b ,c , d;

while ( n <= 31) {

d= n\*n ;

a = d/100;

b = (d / 10) % 10;

c = d % 10;

if (c > b && b > a) {

printf("Найденное число: %d\n", d);

break;

}

n = n+1;

}

}

3

Имеются два сосуда. В первом сосуде находится C1 литров воды, во втором – C2 литров воды. Из первого сосуда переливают половину воды во второй сосуд, затем из второго переливают половину в первый сосуд, и т. д. Сколько воды окажется в обоих сосудах после 12 переливаний?

#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

float C1, C2, a, b;

int n = 1;

printf("Введите кол-во воды в первом сосуде (в литрах): ");

scanf\_s("%f", &C1);

printf("Введите кол-во воды во втором сосуде (в литрах): ");

scanf\_s("%f", &C2);

printf("Количество воды в первом сосуде (в литрах):\t\t");

printf("Количество воды во втором сосуде (в литрах):\n");

while (n <= 12) {

a = C1 / 2;

C1 = C1 - a;

C2 = C2 + a;

printf("%d",n);

printf(")");

printf("%.2f\t\t\t\t\t\t\t", C1);

printf("%d", n);

printf(")");

printf("%.3f\n", C2);

n = n + 1;

b = C2 / 2;

C2 = C2 - b;

C1 = C1 + b;

printf("%d", n);

printf(")");

printf("%.3f\t\t\t\t\t\t", C1);

printf("%d", n);

printf(")");

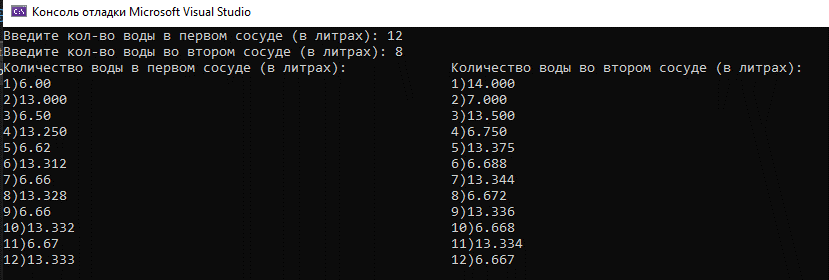
printf("%.3f\n", C2);

n = n + 1;

}

}

}



. Три приятеля были свидетелями нарушения правил дорожного движения. Номер автомобиля – четырехзначное число – никто полностью не запомнил. Из показаний следует, что номер делится на 2, на 7 и на 11, в записи номера участвуют только две цифры, сумма цифр номера равна 30. Составить алгоритм и программу для определения номера автомашины.

4

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

int sum;

for (int i = 1000; i < 9999; i++) {

int q = i / 1000;

int w = (i / 100) % 10;

int e = (i / 10) % 10;

int r = i % 10;

sum = q + w + e + r;

int t = i / 100;

int y= i % 100;

if (i % 2 == 0 && i%7 ==0 && i%11==0 && sum==30) {

cout << "Номер равен : " << i;

}

}

}



**Лабораторная работа № 8. Вычисление сумм, произведений, экстремумов**

1. Выполнить прокрутку и записать условие задачи для программы, представленной в правой части.

Изменить программу так, чтобы вычислялась сумма квадратов четырех значений переменной **a**, а также произведение значений **а**.

Произвести отладку всех вариантов.

Написать программу вычисления факториала:

5! = 1 ∙ 2 ∙ 3 ∙ 4 ∙ 5.

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

float a, sum = 0;

int i;

const int size = 4;

for (i = 0; i < size; i++)

{

cout << "Введите a" << i<< endl;

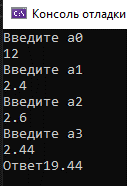
cin >> a;

sum = sum + a;

}

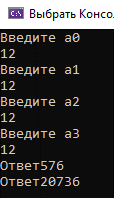
cout << "Ответ" << sum << endl;

}



Условие задачи: Есть переменные(числа с плавающей точкой) a и sum, sum с изначальным значением нуль. Есть целочисленная переменная i.Есть константная целочисленная переменная size с значением 4.В условие цикла for сказано , что переменная i принимает значение 0 и должна быть строго меньше константного значения size, до тех пор пока условие будет выполняться i=i+1.

Осуществляется вывод “Введите a” и переменная i.Затем осуществляется ввод числа(числа с плавающей точкой).Введенное число суммируется с переменной sum.Цикл повторяется до тех пор пока i<size.Затем выводится Ответ с переменной sum.



#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

float a, sum = 0 ,b;

int i;

double p = 1;

const int size = 4;

for (i = 0;i<size; i++)

{

cout << "Введите a" << i<< endl;

cin >> a;

b = a \* a;

sum = sum + b;

p = p \* a;

}

cout << "Ответ" << sum << endl;

cout << "Ответ" << p << endl;

}

(i=0; size=4; i<size; i=i+1)

Вывод:Введите a i=0;

Ввод: a=произвольное число;

sum=sum+a;

(i=1; size=4; i<size; i=i+1)

Вывод:Введите a i=1;

Ввод: a=произвольное число;

sum=sum+a;

(i=2; size=4; i<size; i=i+1)

Вывод:Введите a i=2;

Ввод: a=произвольное число;

sum=sum+a;

(i=3; size=4; i<size; i=i+1)

Вывод:Введите a i=3;

Ввод: a=произвольное число;

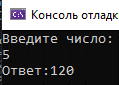
sum=sum+a;

(i=4; size=4; i<size(4<4)-не верно)

Цикл завершен ;

Вывод: Ответ sum.

Прокрутка



#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "Russian");

int f=1,b, a;

cout << "Введите число:" << endl;

cin >> a;

for (b=1;b <= a; b++ )

{

f = f \* b;

}

cout << "Ответ:" << f << endl;

return 0;

}

2. Выполнить прокрутку и записать условия задач для программ, представленных в правой части.

Прокрутка:

Вывод «Enter x:»; Ввод x;

sum = sum + x; count = count + 1; sv = sum / count;

Вывод: sv;

Вывод: if continue input 'y' else 'n'»;

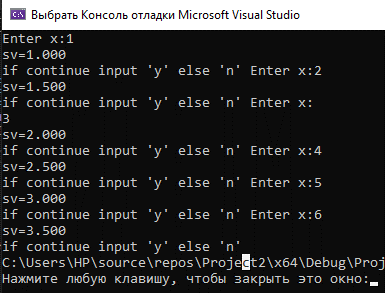
Ввод ch;

If (ch = ‘n’) break

else

Вывод «Enter x:»; Ввод x;

…



**#include <stdio.h>**

**#include <conio.h>**

**void main()**

**{**

**char ch;**

**float sv, x, sum = 0;**

**float count = 0;**

**do**

**{**

**printf("Enter x:");**

**scanf\_s("%f", &x);**

**sum += x;**

**count++;**

**sv = sum / count;**

**printf("sv=%1.3f\n", sv);**

**printf("if continue input 'y' else 'n' ");**

**ch = \_getch();**

**}**

**while (ch != 'n');**

**}**

Условие задачи:

Программа выполняет подсчет среднего арифметического вводимых чисел. После ввода каждого числа программа отображает текущее среднее значение. Программа принимает числа до тех пор , пока пользователь не нажмёт n.После чего программа завершается.

Прокрутка:

t = pow(n, log(n)) = pow(2, log(2)) = 1.616807;

p = pow(log(n), n) = pow(log(2), 2) = 0.480453;

a = t / p = 1.616807 / 0.480453 = 3.365171;

sum = sum + a = 0 + 3.365171 = 3.365171;

t = pow(n, log(n)) = pow(3, log(3)) = 3.343269;

p = pow(log(n), n) = pow(log(3), 3) = 1.325969;

a = t / p = 3.343269 / 1.325969 = 2.521378;

sum = sum + a = 3.365171 + 2.521378 = 5.886549;

…

t = pow(n, log(n)) = pow(9, log(9)) = 124.935287;

p = pow(log(n), n) = pow(log(9), 9) = 1193.630981;

a = t / p = 124.935287 / 1193.630981 = 0.104668;

sum = sum + a = 10.354051 + 0.104668 = 10.458719;

Вывод “S = “ и sum;



#include <stdio.h>

#include <cmath>

void main()

{

float sum = 0, a, t, p;

for (int n = 2; n < 10; n++)

{

t = pow(n, log((float)n));

p = pow(log((float)n), n);

a = t / p;

sum += a;

}

printf("S=%f\n", sum);

}

Условие задачи:

Программа выполняет вычисления 9 раз и выводит сумму полученных значений.

3. Выполнить прокрутку и записать условие задачи для программы, представленной в правой части.

Внести изменения в программу так, чтобы выводилось не только само значение переменной из массива **b**, но и номер этого значения. Даны пять значений переменной :

**b** = {9; −3,5; 54; −2,8; 40}.

**Даны пять значений переменной**

**b = {9; −3,5; 54; −2,8; 40}.**

**m = 9999;**

**n = 1;**

**count = 5;**

**Первый проход (i = 0)**

**Пользователь ввел 9**

**b = 9**

**Если 9 < 9999**

**m = b = 9**

**n = i = 0**

**Второй проход (i = 1)**

**Пользователь ввел -3.5**

**b = -3.5**

**Если -3.5<9**

**m = b = -3.5**

**n = i = 1;**

**Третий проход (i = 2)**

**Пользователь ввел 54**

**b=54**

**n = i = 2;**

**Если 54 < -3.5**

**Четвёртый проход (i = 3)**

**Пользователь ввел -2.8**

**b = -2.8**

**n = i = 3**

**Если -2.8 < -3.5**

**Пятый проход (i = 4)**

**Пользователь ввел 40**

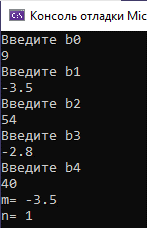
**b = 40**

**n = i = 4**

**Если 40 < -3.5**

**Вывод m = -3.5**

**Вывод n = 1**



**#include <iostream>**

**using namespace std;**

**void main()**

**{**

**setlocale(LC\_CTYPE, "Russian");**

**int count = 5, n = 1;**

**float b, m = 9999;**

**for (int i = 0; i < count; i++)**

**{**

**cout << "Введите b" << i << endl;**

**cin >> b;**

**if (b < m)**

**{ m = b;**

**n = i;**

**}**

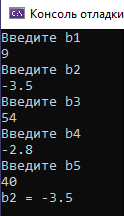
**}**

**cout << "m= " << m << endl;**

**cout << "n= " << n << endl;**

**}**

Условие задачи: В программу, вводится 5 значений. Из этих значений осуществляется вывод минимального , а также соответствующий номер в последовательности.



#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int count = 6, n = 1;

float b, m = 9999;

for (int i = 1; i < count; i++)

{

cout << "Введите b" << i << endl;

cin >> b;

if (b < m)

{

m = b;

n = i;

}

}

cout << "b" << n << " = " << m << endl;

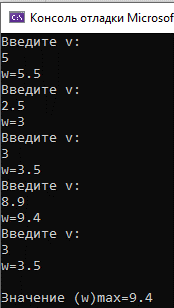
}



Задание 4

Вариант 5(1)





#include <iostream>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

float v, w, a = 0.5, i, m = 0;

int b=7;

for (i = 0; i < 5; i++)

{ cout << "Введите v: "<< endl;

cin >> v;

if (v > 0) {

w = a + v;

}

else if (v <= 0) {

w = b / v;

}

cout << "w=" << w<<endl;

if (w > m)

{

m = w;

}

}

cout << "\n" << "Значение (w)max=" << m << endl;

return 0;

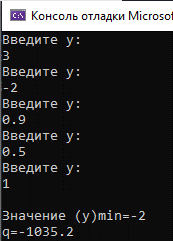
}





Задание 4

Вариант 5(2)



#include <iostream>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

float y,i, p = 9999,d,k=1,q;

for (i = 1;i <= 5; i++)

{

cout << "Введите y: " << endl;

cin >>y;

d = y - 5;

k = k \* d;

if (y < p)

{

p = y;

}

}

q = k + p;

cout << "\n" << "Значение (y)min=" << p << endl;

cout << "q=" << q << endl;

return 0;

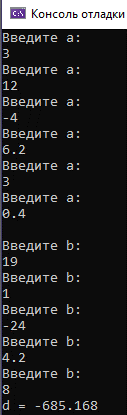
}

Задание 4

Вариант 8(1)







#include <iostream>

#include <cmath>

#include <math.h>

using namespace std;

void main()

{

setlocale(LC\_ALL, "RUSSIAN");

float n1 = 6, n2 = 5, c = 0.7, i, b, a, sum1 = 0, sum2 = 0, d;

for (i = 0; i < n1; i++)

{

cout << "Введите a: "<<endl;

cin >> a;

sum1 += a;

}

cout << "\n";

for (i = 0; i < n2; i++)

{

cout << "Введите b: " << endl;

cin >> b;

sum2 += (b - 1) \* (b - 1);

}

d = sum1 - c \* sum2;

cout << "d = " << d;

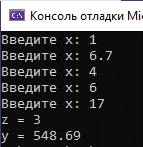
}





Задание 4

Вариант 8(2)



#include <iostream>

#include <cmath>

#include <math.h>

using namespace std;

void main()

{

setlocale(LC\_ALL, "RUSSIAN");

float x, i, z = 9999, sum1 = 0, y;

for (i = 0; i < 5; i++)

{

cout << "Введите x: ";

cin >> x;

x += 2;

if (x < z)

z = x;

sum1 += x \* x;

}

y = sum1 + z;

cout << "z = " << z<<endl;

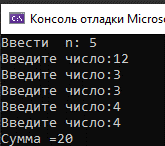
cout << "y = " << y;

}

Дополнительные задания

В последовательности из **n** целых чисел найти и вывести значение суммы четных элементов.

1



#include <iostream>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

float i,n, sum = 0;

int x;

cout << "Ввести n: ";

cin >> n;

for (int i = 1; i < n + 1; i++) {

cout << "Введите число:";

cin >> x;

if (x % 2 == 0) {

sum += x;

}

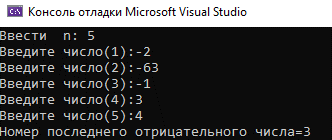
}

cout <<"Сумма =" << sum;

}

В последовательности из **n** целых чисел найти и вывести порядковый номер последнего отрицательного элемента.

2



#include <iostream>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

float i, n,f;

int x;

cout << "Ввести n: ";

cin >> n;

for (int i = 1; i < n + 1; i++) {

cout << "Введите число"<<"("<<i<<"):";

cin >> x;

if (x < 0) {

f=i;

}

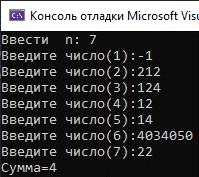
}

cout <<"Номер последнего отрицательного числа=" <<f;

}

3

В последовательности из **n** вещественных чисел найти количество элементов, стоящих между минимальным и максимальным значениями.



#include <iostream>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

int n, i, min1, max1, sum;

float x, max = -9999, min = 9999;

cout << "Ввести n: ";

cin >> n;

for (i = 1; i < n + 1; i++) {

cout << "Введите число"<<"("<<i<<"):";

cin >> x;

if (x > max) {

max = x;

max1 = i;

}

if (x < min) {

min = x;

min1 = i;

}

}

sum = max1 - min1;

if(sum<0){sum=sum\*(-1);}

sum -= 1;

cout <<"Сумма=" <<sum;

}

**Лабораторная работа № 9. Решение инженерных задач на основе циклических программ**

Вариант 5

5. В соответствии со своим вариантом написать программы для вычисления площади криволинейной трапеции по исходным данным из таблицы, приведенной ниже, методом ***трапеций*** и методом ***парабол***. Для всех вариантов принять **n** = 200. Сравнить результаты, которые должны отличаться на небольшую величину.

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Функция f(x)** | **Границы отрезков** |
| 5 | 5 – x2 | a = 8, b = 12(Числа изменил a=0,b=3) |

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

float n = 200, a = 0, b = 3, h, x, s = 0;

h = (b - a) / n;

x = a;

while (x < (b - h)) {

s = s + h \* ((5 - x \* x) + 5 - (x + h) \* (x + h)) / 2;

x = x + h;

}

cout << "Метод трапеции: " << s;

}



#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

float n = 200, a = 0, b = 3, s1 = 0, s2 = 0, i = 1, x, h, S;

h = (b - a) / (2 \* n);

x = a + 2 \* h;

while (i < n)

{

s2 = s2 + (5 - x \* x);

x = x + h;

s1 = s1 + (5 - x \* x);

x = x + h;

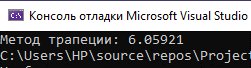
i = i + 1;

}

S = h / 3 \* ((5 - a \* a) + 4 \* (5 - (a + h) \* (a + h)) + 4 \* s1 + 2 \* s2 + (5 - b \* b));

cout << "Метод параболы: " << S;

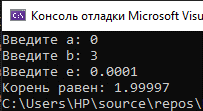
}



6. В соответствии со своим вариантом найти отрезок (значения **a** и **b**), который содержит один корень, ***отделив корни*** уравнения ***графическим*** методом для исходных данных из таблицы, приведенной ниже. Если корней несколько, то выбрать один из отрезков.

|  |  |
| --- | --- |
| **№ варианта** | **Функция f(x)** |
| 5 | 2 – x2 + x |





#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

double a, b, e, x;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

cout << "Введите e: ";

cin >> e;

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;

if (((2 - x \* x + x) \* (2 - a \* a + a)) <= 0)

b = x;

else

a = x;

}

x = (a + b) / 2;

cout << "Корень равен: " << x;

}

**Дополнительные задания**

Вариант 10

5. В соответствии со своим вариантом написать программы для вычисления площади криволинейной трапеции по исходным данным из таблицы, приведенной ниже, методом ***трапеций*** и методом ***парабол***. Для всех вариантов принять **n** = 200. Сравнить результаты, которые должны отличаться на небольшую величину.

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Функция f(x)** | **Границы отрезков** |
| 10 | x2 – 4 | a = 4, b = 8 |

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

float n = 200, a = 4, b = 8, h, x, s = 0;

h = (b - a) / n;

x = a;

while (x < (b - h)) {

s = s + h \* ((x\*x-4) + (x + h) \* (x + h)-4) / 2;

x = x + h;

}

cout << "Метод трапеции: " << s;

}



#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

float n = 200, a = 4, b = 8, s1 = 0, s2 = 0, i = 1, x, h, S;

h = (b - a) / (2 \* n);

x = a + 2 \* h;

while (i < n)

{

s2 = s2 + (x \* x - 4);

x = x + h;

s1 = s1 + (x \* x - 4);

x = x + h;

i = i + 1;

}

S = h / 3 \* ((a \* a - 4) + 4 \* ((a + h) \* (a + h) - 4) + 4 \* s1 + 2 \* s2 + (b \* b - 4));

cout << "Метод параболы: " << S;

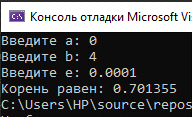
}



6. В соответствии со своим вариантом найти отрезок (значения **a** и **b**), который содержит один корень, ***отделив корни*** уравнения ***графическим*** методом для исходных данных из таблицы, приведенной ниже. Если корней несколько, то выбрать один из отрезков.

|  |  |
| --- | --- |
| **№ варианта** | **Функция f(x)** |
| 10 | ex + 2x2 – 3 |





#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

double a, b, e, x;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

cout << "Введите e: ";

cin >> e;

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;

if (((exp(x)+2\*x\*x-3) \* (exp(a)+2\*a\*a-3)) <= 0)

b = x;

else

a = x;

}

x = (a + b) / 2;

cout << "Корень равен: " << x;

}

Вариант 7

5. В соответствии со своим вариантом написать программы для вычисления площади криволинейной трапеции по исходным данным из таблицы, приведенной ниже, методом ***трапеций*** и методом ***парабол***. Для всех вариантов принять **n** = 200. Сравнить результаты, которые должны отличаться на небольшую величину.

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Функция f(x)** | **Границы отрезков** |
| 7 | e x + 2 | a = 5, b = 11 |

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

float n = 200, a = 5, b = 11, h, x, s = 0;

h = (b - a) / n;

x = a;

while (x < (b - h)) {

s = s + h \* (exp(x)+2+exp(h+x )+2) / 2;

x = x + h;

}

cout << "Метод трапеции: " << s;

}



#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

float n = 200, a = 5, b = 11, s1 = 0, s2 = 0, i = 1, x, h, S;

h = (b - a) / (2 \* n);

x = a + 2 \* h;

while (i < n)

{

s2 = s2 + (exp(x) + 2);

x = x + h;

s1 = s1 + (exp(x)+2);

x = x + h;

i = i + 1;

}

S = h / 3 \* ((exp(a) + 2) + 4 \* (exp(a+h) + 2) + 4 \* s1 + 2 \* s2 + (exp(b) + 2));

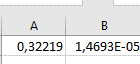
cout << "Метод параболы: " << S;

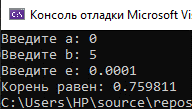
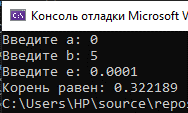
}



6. В соответствии со своим вариантом найти отрезок (значения **a** и **b**), который содержит один корень, ***отделив корни*** уравнения ***графическим*** методом для исходных данных из таблицы, приведенной ниже. Если корней несколько, то выбрать один из отрезков.

|  |  |
| --- | --- |
| **№ варианта** | **Функция f(x)** |
| 7 | x3 + 3x – 1 |





#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

double a, b, e, x;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

cout << "Введите e: ";

cin >> e;

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;

if (((pow(x,3)+3\*x-1) \* (pow(a, 3) + 3 \* a - 1)) <= 0)

b = x;

else

a = x;

}

x = (a + b) / 2;

cout << "Корень равен: " << x;

}

Вариант 11

5. В соответствии со своим вариантом написать программы для вычисления площади криволинейной трапеции по исходным данным из таблицы, приведенной ниже, методом ***трапеций*** и методом ***парабол***. Для всех вариантов принять **n** = 200. Сравнить результаты, которые должны отличаться на небольшую величину.

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Функция f(x)** | **Границы отрезков** |
| 11 | sin(x) + 1 | a = 1, b = 3 |

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

float n = 200, a = 1, b = 3, h, x, s = 0;

h = (b - a) / n;

x = a;

while (x < (b - h)) {

s = s + h \* (sin(x)+1 + sin(x + h) + 1) / 2;

x = x + h;

}

cout << "Метод трапеции: " << s;

}



#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

float n = 200, a = 1, b = 3, s1 = 0, s2 = 0, i = 1, x, h, S;

h = (b - a) / (2 \* n);

x = a + 2 \* h;

while (i < n)

{

s2 = s2 + (sin(x)+1);

x = x + h;

s1 = s1 + (sin(x) + 1);

x = x + h;

i = i + 1;

}

S = h / 3 \* ((sin(a) + 1) + 4 \* (sin(a + h) + 1) + 4 \* s1 + 2 \* s2 + (sin(b) + 1));

cout << "Метод параболы: " << S;

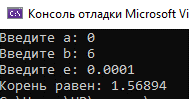
}



6. В соответствии со своим вариантом найти отрезок (значения **a** и **b**), который содержит один корень, ***отделив корни*** уравнения ***графическим*** методом для исходных данных из таблицы, приведенной ниже. Если корней несколько, то выбрать один из отрезков.

|  |  |
| --- | --- |
| **№ варианта** | **Функция f(x)** |
| 11 | 2x + x3 – 7 |





#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

double a, b, e, x;

cout << "Введите a: ";

cin >> a;

cout << "Введите b: ";

cin >> b;

cout << "Введите e: ";

cin >> e;

while (abs(a - b) > 2 \* e) {

x = (a + b) / 2;

if (((2\*x+pow(x,3)-7) \* (2 \* a + pow(a, 3) - 7)) <= 0)

b = x;

else

a = x;

}

x = (a + b) / 2;

cout << "Корень равен: " << x;

}

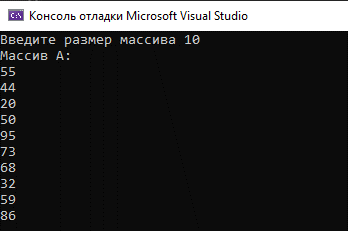
**Лабораторная работа № 10. Одномерные массивы**

1.

1. Выполнить программу, записанную в правой части. Опробовать второй вариант генерации чисел, записанный в комментарии.

Добавить в программу операторы вычисления суммы элементов массива **А**.

Произвести отладку.



#include <iostream>

#include <ctime>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

const int N=100;

int i, sz, A[N], rmn = 0, rmx = 99;

cout << "Введите размер массива ";

cin >> sz;

cout << "Массив A:" << endl;

srand((unsigned)time(NULL));

for (i = 0; i < sz; i++)

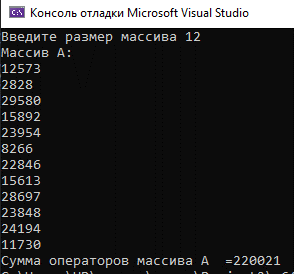
{

A[i] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);

cout << A[i] << endl;

}

}



#include <iostream>

#include <ctime>

#include <locale>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

const int N=100;

int i, sz, A[N], rmn = 0, rmx = 99;

float sum = 0;

cout << "Введите размер массива ";

cin >> sz;

cout << "Массив A:" << endl;

srand((unsigned)time(NULL));

for (i = 0; i < sz; i++)

{

A[i] = rand();

cout << A[i] << endl;

sum = sum+A[i];

}

cout << "Сумма операторов массива A =" <<sum;

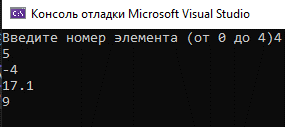
}

2

2. Выполнить программу, записанную в правой части. Записать ее условие.

Добавить в программу операторы вычисления среднего значения элементов исходного массива.

Произвести отладку.



#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int i, k, size = 4;

float massivA[] = { 5, -4, 17.1, 9, 1 };

cout << "Введите номер элемента (от 0 до 4)";

cin >> k;

for (i = k; i <= size; i++)

massivA[i] = massivA[i + 1];

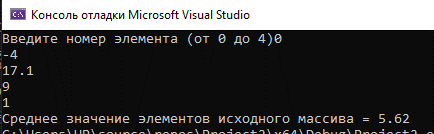
size--;

for (i = 0; i <= size; i++)

cout << massivA[i] << endl;

}

УСЛОВИЕ: Программа удаляет элемент массива по указанному номеру и выводит исходный massivA после удаления элемента из него.



#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int i, k, size = 4;

float massivA[] = { 5, -4, 17.1, 9, 1 },sum=0,ex;

cout << "Введите номер элемента (от 0 до 4)";

cin >> k;

for (i = 0; i <= size; i++) {

sum += massivA[i];

}

ex = sum / (size + 1);

for (i = k; i <= size; i++)

massivA[i] = massivA[i + 1];

size--;

for (i = 0; i <= size; i++)

cout << massivA[i] << endl;

cout << "Среднее значение элементов исходного массива = " << ex;

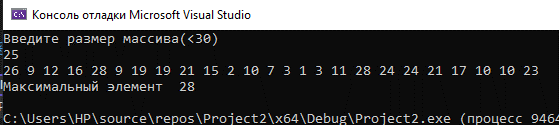
}

3

3. Выполнить программу, приведенную в правой части.

Внести изменения с тем, чтобы вычислялся минимальный элемент массива.

Произвести отладку.



#include <iostream>

#include <locale>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int maxSize = 30;

int n, i, kmax = 0;

int a[maxSize];

cout << "Введите размер массива(<30)" << endl;

cin >> n;

if (n > 30)

return;

srand((unsigned)time(NULL));

for(i=0;i<n;i++)

{

a[i] = rand() % 30;

cout << a[i] << " ";

}

cout << endl;

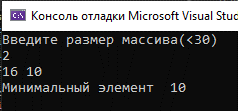
for (i = 1; i < n; i++)

if (a[i] > a[kmax])

kmax = i;

cout << "Максимальный элемент "<< a[kmax] << endl;

}



#include <iostream>

#include <locale>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int maxSize = 30;

int n, i, kmax = 0,kmin=9999 ,r;

int a[maxSize];

cout << "Введите размер массива(<30)" << endl;

cin >> n;

if (n > 30)

return;

srand((unsigned)time(NULL));

for(i=0;i<n;i++)

{

a[i] = rand() % 30;

cout << a[i] << " ";

}

cout << endl;

for (i = 0; i < n; i++)

if (a[i] < kmin)

kmin = a[i];

cout << "Минимальный элемент " <<kmin << endl;

}

#include <iostream>

#include <locale>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int maxSize = 30;

int n, i, kmax = 0,kmin=9999 ,r;

int a[maxSize];

cout << "Введите размер массива(<30)" << endl;

cin >> n;

if (n > 30)

return;

srand((unsigned)time(NULL));

for(i=0;i<n;i++)

{

a[i] = rand() % 30;

cout << a[i] << " ";

}

cout << endl;

for (i = 0; i < n; i++)

if (a[i] < kmin)

kmin = a[i];

for (i = 1; i < n; i++)

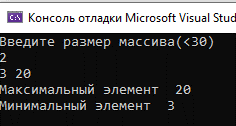
if (a[i] > a[kmax])

kmax = i;

cout << "Максимальный элемент " << a[kmax] << endl;

cout << "Минимальный элемент " <<kmin << endl;

}



4

4. Выполнить прокрутку программы, приведенной в правой части. Записать условие.

Опробовать программу для массивов **A** и **B** разного размера.

Условие: Пользователь вводит 5 чисел для массива А и 5 чисел для массива B.Затем формируется общий массив состоящий из массивов А и В (10 чисел) и выводится.

**#include <stdio.h>**

**void main()**

**{ const int size = 5;**

**int masA[size], masB[size], masC[size\*2];**

**int k = 0, j = 0, i = 0;**

**printf("A:\n");**

**for (int n = 0; n < size; n++)**

**scanf\_s("%d", &masA[n]);**

**printf("B:\n");**

**for (int n = 0; n < size; n++)**

**scanf\_s("%d", &masB[n]);**

**do**

**{ if (masA[k] <= masB[j])**

**masC[i++] = masA[k++];**

**else**

**masC[i++] = masB[j++];**

**if (k == size)**

**for (; j < size; j++)**

**masC[i++] = masB[j];**

**if (j == size)**

**for (; k < size; k++)**

**masC[i++] = masA[k];**

**}**

**while (i < size \* 2);**

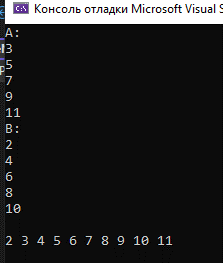
**printf("\n");**

**for (i = 0; i < size \* 2; i++)**

**printf("%d ", masC[i]);**

**printf("\n");**

**}**



do

if (masA[k] <= masB[j])

masC[i++] = masA[k++];

else

masC[i++] = masB[j++];

if (k == size)

for (; j < size; j++)

masC[i++] = masB[j];

if (j == size)

for (; k < size; k++)

masC[i++] = masA[k];

while (i < size \* 2);

Прокрутка:

Вывод: ‘A:’

n=0,n<5

Ввод: n

n++

n=1,n<5

Ввод: n

n++

n=2,n<5

Ввод: n

n++

n=3,n<5

Ввод: n

n++

n=4,n<5

Ввод: n

n++

n=5,5<5 – ложь (цикл завершен)

Вывод: ‘A:’

n=0,n<5

Ввод: n

n++

n=1,n<5

Ввод: n

n++

n=2,n<5

Ввод: n

n++

n=3,n<5

Ввод: n

n++

n=4,n<5

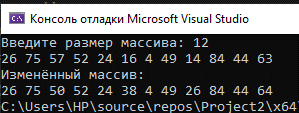
Ввод: n

n++

n=5,5<5 – ложь (цикл завершен)

1. В массиве **С** каждый третий элемент заменить полусуммой двух предыдущих.

5



#include <iostream>

#include <locale>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int n, i, C[N];

cout << "Введите размер массива: ";

cin >> n;

srand((unsigned)time(NULL));

for (i = 0; i < n; i++)

{

C[i] = rand() % 100;

cout << C[i] << " ";

}

cout << "\n";

for (int i = 2; i < n; i += 3) {

C[i] = (C[i - 1] +C[i - 2]) / 2;

}

cout<<"Изменённый массив:"<<endl;

for (int i = 0; i < n; i++) {

cout<< C[i] << " ";

}

}



2. Задан массив из **n** символов. Определить, симметричен ли он, т. е. читается ли он одинаково слева направо и справа налево.

#include <iostream>

#include <locale>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int n, i, C[N], Symmetric = 1;

cout << "Введите размер массива: ";

cin >> n;

srand((unsigned)time(NULL));

for (i = 0; i < n; i++)

{

C[i] = rand() % 100;

cout << C[i] << " ";

}

cout << endl;

for (int i = 0; i < n / 2; i++) {

if (C[i] != C[n - 1 - i]) {

Symmetric = 0;

}

}

if (Symmetric) {

printf("Массив симметричен.\n");

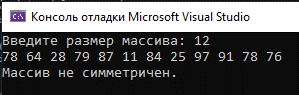
}

else {

printf("Массив не симметричен.\n");

}

}

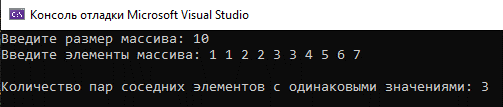


Дополнительные задания

Подсчитать количество пар соседних элементов массива с одинаковыми значениями.

ч

2



#include <iostream>

#include <locale>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int n, i, C[N];

cout << "Введите размер массива: ";

cin >> n;

cout << "Введите элементы массива: ";

for (i = 0; i < n; i++)

{

cin >> C[i];

}

cout << endl;

int count = 0;

for (i = 0; i < n - 1; ++i) {

if (C[i] == C[i + 1]) {

++count;

}

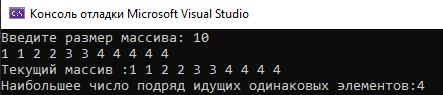
}

cout << "Количество пар соседних элементов с одинаковыми значениями: " << count << endl;

}

Найти в массиве наибольшее число подряд идущих одинаковых элементов.

3



#include <iostream>

#include <locale>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int n, i, C[N];

cout << "Введите размер массива: ";

cin >> n;

for (i = 0; i < n; i++)

{

cin >> C[i];

}

cout << "Текущий массив :";

for (i = 0; i < n; i++) {

cout << C[i] << " ";

}

cout << endl;

int max =1 ;

int current=1;

for (i = 0; i < n - 1; i++) {

if (C[i] == C[i + 1]) {

current++;

}

else{

if (current > max) {

max = current;

}

current = 1;

}

}

if (current > max) {

max = current;

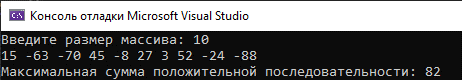
}

cout << "Наибольшее число подряд идущих одинаковых элементов:"<< max;

}

Найти непрерывную последовательность положительных чисел, сумма элементов в которой максимальная. Максимальный размер массива 100 элементов. Диапазон значений от -100 до 100.

6



#include <iostream>

#include <locale>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int n, i, C[N];

cout << "Введите размер массива: ";

cin >> n;

srand((unsigned)time(NULL));

for (i = 0; i < n; i++)

{

C[i] = rand() % 201 - 100;

cout << C[i] << " ";

}

cout << endl;

int maxSum = 0, currentSum = 0;

int startIndex = 0, endIndex = 0, tempStart = 0;

for (int i = 0; i < n; i++) {

if (C[i] > 0) {

currentSum += C[i];

if (currentSum > maxSum) {

maxSum = currentSum;

startIndex = tempStart;

endIndex = i;

}

} else {

currentSum = 0;

tempStart = i + 1;

}

}

if (maxSum > 0) {

cout << "Максимальная сумма положительной последовательности: " << maxSum << endl;

} else {

cout << "Нет положительных чисел в массиве." << endl;

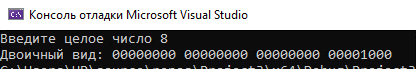
}

}

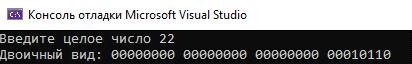
**Лабораторная работа № 11. Битовые операции**

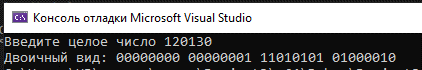
Изучить использование битовых операций и маскирования числа, опробовав программу, записанную в правой части, с различными исходными числами.

1.









**#include <iostream>**

**using namespace std;**

**void main()**

**{ setlocale(LC\_CTYPE, "Russian");**

**unsigned int value; int i;**

**const unsigned int mask = 1 << 31;**

**cout << "Введите целое число ";**

**cin >> value;**

**cout << "Двоичный вид: ";**

**for (i = 1; i <= 32; i++)**

**{ putchar(mask & value ? '1' : '0');**

**value <<= 1;**

**if (i % 8 == 0) putchar(' ');**

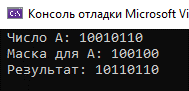
**}**

**}**

Выполнить программу, записанную в правой части. Ознакомиться с результатом.

Опробовать программу, изменяя различные биты различных чисел.

2



**#include <iostream>**

**using namespace std;**

**void main()**

**{**

**setlocale(LC\_CTYPE, "Russian");**

**int A = 150; char tmp[33];**

**\_itoa\_s(A, tmp, 2);**

**cout << " Число А: " << tmp << endl;**

**\_itoa\_s(0x24, tmp, 2);**

**cout << " Маска для А: " << tmp << endl;**

**\_itoa\_s(A | 0x24, tmp, 2);**

**cout << " Результат: " << tmp <<endl<<endl;**

**}**

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int A = 75; char tmp[33];

\_itoa\_s(A, tmp, 2);

cout << " Число А: " << tmp << endl;

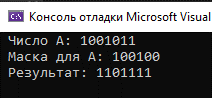
\_itoa\_s(0x24, tmp, 2);

cout << " Маска для А: " << tmp << endl;

\_itoa\_s(A | 0x24, tmp, 2);

cout << " Результат: " << tmp << endl << endl;

}



3

В программе, записанной в правой части, используются различные битовые операции.

Внести изменения в программу с тем, чтобы проверялось число на кратность четырем.

**#include <iostream>**

**using namespace std;**

**void main()**

**{ setlocale(LC\_CTYPE, "Russian");**

**int A; char tmp[33];**

**cout << "Введите число ";**

**cin >> A;**

**\_itoa\_s(A, tmp, 2);**

**cout << "Число в двоичном виде = "<<tmp<<endl;**

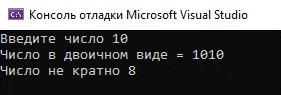
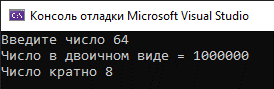
**if ((A & 7) == 0)**

**cout << "Число кратно 8" << endl;**

**else**

**cout << "Число не кратно 8" << endl;**

**}**

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int A; char tmp[33];

cout << "Введите число ";

cin >> A;

\_itoa\_s(A, tmp, 2);

cout << "Число в двоичном виде = " << tmp << endl;

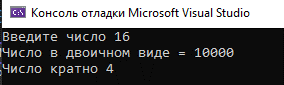
if ((A & 3) == 0)

cout << "Число кратно 4" << endl;

else

cout << "Число не кратно 8" << endl;

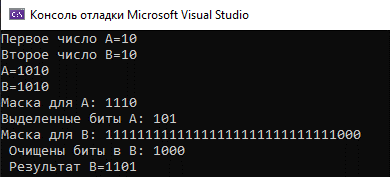
}

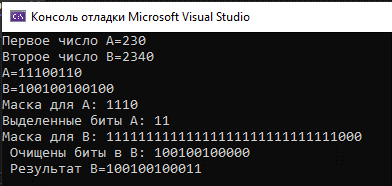


В правой части приведен пример программы, демонстрирующей использование битовых операций.

Проанализировать текст программы и написать пояснения.

4





**#include <iostream>**

**using namespace std;**

**void main()**

**{**

**setlocale(LC\_CTYPE, "Russian");**

**char tmp[33];**

**int A, B, maskA = 14;**

**int maskB = ~maskA >> 1;**

**cout << "Первое число А="; cin >> A;**

**cout << "Второе число В="; cin >> B;**

**\_itoa\_s(A, tmp, 2);**

**cout << "A=" << tmp << endl;**

**\_itoa\_s(B, tmp, 2);**

**cout << "B=" << tmp << endl;**

**\_itoa\_s(maskA, tmp, 2);**

**cout << "Маска для А: " << tmp << endl;**

**\_itoa\_s((A & maskA) >> 1, tmp, 2);**

**cout << "Выделенные биты А: " << tmp << endl;**

**\_itoa\_s(maskB, tmp, 2);**

**cout << "Маска для В: " << tmp << endl;**

**\_itoa\_s(B & maskB, tmp, 2);**

**cout << " Очищены биты в B: " << tmp << endl;**

**\_itoa\_s(((B & maskB) | ((A & maskA) >> 1)), tmp, 2);**

**cout << " Результат B=" << tmp << endl;**

**}**

Пояснение: #include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

char tmp[33];

int A, B;

int maskA = 14; // Маска для извлечения 3 битов (01110 в двоичном виде)

int maskB = ~maskA >> 1; // Маска для очищения 3 битов в числе B

cout << "Первое число А=";

cin >> A;

cout << "Второе число В=";

cin >> B;

\_itoa\_s(A, tmp, 2); // Преобразование числа A в двоичную строку

cout << "A=" << tmp << endl;

\_itoa\_s(B, tmp, 2); // Преобразование числа B в двоичную строку

cout << "B=" << tmp << endl;

\_itoa\_s(maskA, tmp, 2); // Преобразование маски A в двоичную строку

cout << "Маска для А: " << tmp << endl;

\_itoa\_s((A & maskA) >> 1, tmp, 2); // Извлечение 3 битов из A и сдвиг вправо на 1 бит, затем преобразование в строку

cout << "Выделенные биты А: " << tmp << endl; // Вывод извлеченных битов из A

\_itoa\_s(maskB, tmp, 2); // Преобразование маски B в двоичную строку

cout << "Маска для В: " << tmp << endl;

\_itoa\_s(B & maskB, tmp, 2); // Очистка 3 битов в B с помощью маски B и преобразование в строку

cout << " Очищены биты в B: " << tmp << endl; // Вывод числа B с очищенными битами

\_itoa\_s(((B & maskB) | ((A & maskA) >> 1)), tmp, 2); // Вставка извлеченных битов из A в B и преобразование в строку

cout << " Результат B=" << tmp << endl;

}

1. Определить, насколько в числе **А** больше значащих битов, равных единице, чем битов, равных нулю.

5

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

int A;

cout << "Введите число A: ";

cin >> A;

const int BITS = 31; // Количество битов для int

// Вывод числа A в двоичном виде

cout << "Число A в двоичном виде: ";

for (int i = BITS ; i >= 0; --i) {

cout << ((A >> i) & 1); // Сдвиг вправо и проверка младшего бита

// Разделение на байты (по 8 бит)

if (i % 8 == 0 && i != 0) {

cout << ' ';

}

}

cout << endl;

int count1 = 0; // Счётчик битов, равных 1

int count0 = 0; // Счётчик битов, равных 0

// Подсчёт количества единиц и нулей

for (int i = 0; i < BITS; ++i) {

if (A & (1 << i)) {

count1++; // Если текущий бит равен 1

}

else {

count0++; // Если текущий бит равен 0

}

}

// Вывод результатов

cout << "Количество единиц: " << count1 << endl;

cout << "Количество нулей: " << count0 << endl;

// Сравнение количества единиц и нулей

if (count1 > count0) {

cout << "В числе A больше значащих битов равных единице, чем битов равных нулю на (" << count1 - count0 << ")" << endl;

}

else if (count1 < count0) {

cout << "В числе A больше значащих битов равных нулю, чем битов равных единице на (" << count0 - count1 << ")" << endl;

}

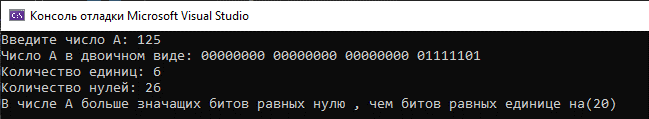
else {

cout << "Количество значащих битов, равных единице, равно количеству битов, равных нулю." << endl;

}

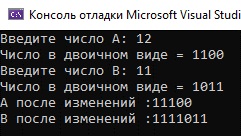
return 0;

}



2.Установить в 1 **n** битов в числе **А** вправо от позиции **p**, заменить ими **m** битов числа **В**, начиная с позиции **q**.

5



#include <iostream>

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

int A,B; // Используйте префикс 0b для двоичных значений

int p = 2;

int q = 4;

int m = 3;

char tmp[33];

cout << "Введите число А: ";

cin >> A;

\_itoa\_s(A, tmp, 2);

cout << "Число в двоичном виде = " << tmp << endl;

cout << "Введите число B: ";

cin >> B;

\_itoa\_s(B, tmp, 2);

cout << "Число в двоичном виде = " << tmp << endl;

int mask = ((1 << m) - 1) << p;

A |= mask;

int extractedBits = (A >> p) & ((1 << m) - 1);

int clearMask = ~(((1 << m) - 1) << q);

B &= clearMask;

B |= (extractedBits << q);

\_itoa\_s(A, tmp, 2);

cout << "A после изменений :" << tmp << endl;

\_itoa\_s(B, tmp, 2);

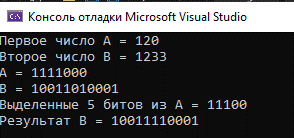
cout << "B после изменений :" << tmp << endl;

}

Дополнительные задания

2

1. Извлечь 5 битов числа **A**, начиная со второго и вставить их в число **B**, начиная с третьего бита.



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

char tmp[33];

int A, B;

cout << "Первое число A = ";

cin >> A;

cout << "Второе число B = ";

cin >> B;

\_itoa\_s(A, tmp, 2);

cout << "A = " << tmp << endl;

\_itoa\_s(B, tmp, 2);

cout << "B = " << tmp << endl;

int maskA = 0b11111 << 1;

int extractedBits = (A & maskA) >> 1;

\_itoa\_s(extractedBits, tmp, 2);

cout << "Выделенные 5 битов из A = " << tmp << endl;

int maskB = ~(0b11111 << 2);

B = B & maskB;

B = B | (extractedBits << 2);

\_itoa\_s(B, tmp, 2);

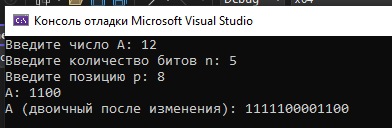
cout << "Результат B = " << tmp << endl;

return 0;

}

2.Установить в числе **А n** битов вправо от позиции **p**.

2



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

int A, n, p;

char tmp[33];

cout << "Введите число A: ";

cin >> A;

cout << "Введите количество битов n: ";

cin >> n;

cout << "Введите позицию p: ";

cin >> p;

\_itoa\_s(A, tmp, 2);

cout << "A: " << tmp << endl;

int mask = (1 << n) - 1;

mask <<= p;

A |= mask;

\_itoa\_s(A, tmp, 2);

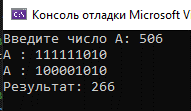
cout << "A (двоичный после изменения): " << tmp << endl;

return 0;

}

12

1. Ввести целое число **A**. Инвертировать все биты с 4 по 8, включая эти биты. Вывести полученное число.



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

int A;

cout << "Введите число A: ";

cin >> A;

char tmp[33];

\_itoa\_s(A, tmp, 2);

cout << "A : " << tmp << endl;

int mask = ((1 << 4) - 1) << 4;

A ^=mask;

\_itoa\_s(A, tmp, 2);

cout << "A : " << tmp << endl;

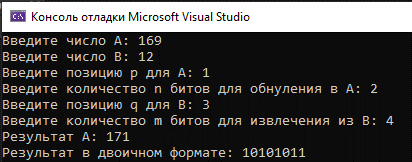
cout << "Результат: " << A << endl;

return 0;

}

2.Установить в **n** битов в числе **А** вправо от позиции **p**, заменить ими **m** битов числа **В**, начиная с позиции **q**.

12



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

char tmp[33];

int A, B, p, n, q, m;

cout << "Введите число A: ";

cin >> A;

cout << "Введите число B: ";

cin >> B;

cout << "Введите позицию p для A: ";

cin >> p;

cout << "Введите количество n битов для обнуления в A: ";

cin >> n;

cout << "Введите позицию q для B: ";

cin >> q;

cout << "Введите количество m битов для извлечения из B: ";

cin >> m;

// Обнуление n битов в A начиная с позиции p

int maskA = ~(((1 << n) - 1) << p); // Создаем маску с нулями на нужных позициях

A &= maskA; // Применяем маску, чтобы обнулить n битов

// Извлечение m битов из B начиная с позиции q

int BitsB = (B >> q) & ((1 << m) - 1); // Сдвигаем B и выделяем m битов

// Установка извлеченных битов в A начиная с позиции p

A |= (BitsB << p); // Сдвигаем извлеченные биты и устанавливаем их в A

// Результат

cout << "Результат A: " << A << endl;

// Вывод в двоичном формате

\_itoa\_s(A , tmp, 2);

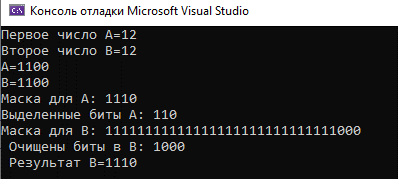
cout << "Результат в двоичном формате: "<<tmp;

return 0;

}

1. Ввести целое число **A**. Извлечь 3 бита числа A, начиная со второго по счету справа, и вставить их в число **B**, начиная с первого бита по счету справа.

16



**#include <iostream>**

**using namespace std;**

**void main()**

**{**

**setlocale(LC\_CTYPE, "Russian");**

**char tmp[33];**

**int A, B, maskA = 14;**

**int maskB = ~maskA >> 1;**

**cout << "Первое число А="; cin >> A;­­**

**cout << "Второе число В="; cin >> B;**

**\_itoa\_s(A, tmp, 2);**

**cout << "A=" << tmp << endl;**

**\_itoa\_s(B, tmp, 2);**

**cout << "B=" << tmp << endl;**

**\_itoa\_s(maskA, tmp, 2);**

**cout << "Маска для А: " << tmp << endl;**

**\_itoa\_s((A & maskA) >> 1, tmp, 2);**

**cout << "Выделенные биты А: " << tmp << endl;**

**\_itoa\_s(maskB, tmp, 2);**

**cout << "Маска для В: " << tmp << endl;**

**\_itoa\_s(B & maskB, tmp, 2);**

**cout << " Очищены биты в B: " << tmp << endl;**

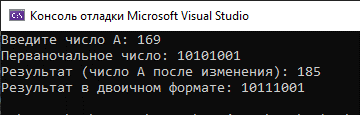
**\_itoa\_s(((B & maskB) | ((A & maskA) >> 1)), tmp, 2);**

**cout << " Результат B=" << tmp << endl;**

**}**

2.Установить в два бита числа **А**, начиная с четвертого по счету справа.

16



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

char tmp[33];

int A;

cout << "Введите число A: ";

cin >> A;

\_itoa\_s(A, tmp, 2);

cout << "Перваночальное число: " << tmp << endl;

int mask = (1 << 4) | (1 << 5);

A |= mask;

cout << "Результат (число A после изменения): " << A << endl;

\_itoa\_s(A, tmp, 2);

cout << "Результат в двоичном формате: "<<tmp<<endl;

return 0;

}

**Лабораторная работа № 12. Использование указателей**

1.

Изучить работу с *указателями*, выполнив программы, записанные в правой части.

Использовать средства отладки, проанализировать память компьютера.

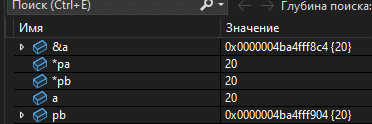
Записать условия задач.

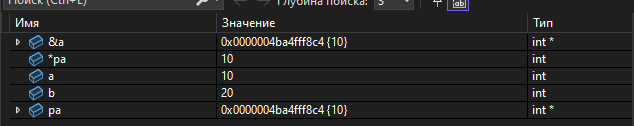
Условие задачи №1

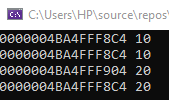
1) Вывести адреса переменных и их значения.

2) Присвоить одной переменной значение другой через указатель.

3) Проанализировать изменения в памяти с помощью отладчика.







#include <iostream>

using namespace std;

void main()

{

int a = 10, \* pa, b = 20, \* pb;

pa = &a;

cout << &a << " " << a << endl;

cout << pa << " " << \*pa << endl;

pb = &b;

cout << pb << " " << \*pb << endl;

\*pa = \*pb;

cout << &a << " " << a << endl;

}

**#include <iostream>**

**void main()**

**{ using namespace std;**

**float a, \*pa, s = 0; int i;**

**pa = &a;**

**for (i = 1; i <= 4; i++)**

**{ cout << "a=";**

**cin >> a;**

**s += \*pa;**

**}**

**cout << "s=" << s << endl;**

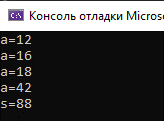
**}**

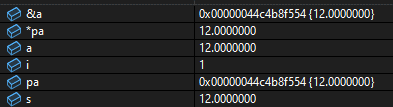
Условие задачи №2  
1) Пользователь вводит значение переменной a четыре раза.

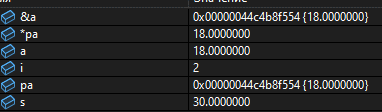
2) Указатель pa используется для доступа к переменной a (через разыменование \*pa).

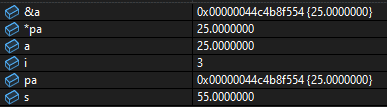
3) Сумма значений, вводимых в a, вычисляется и сохраняется в переменной s.

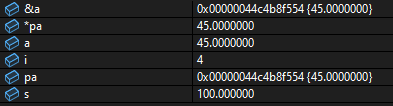
По завершении работы программа выводит сумму s .













2. В правой части записаны фрагменты программ с использованием *указателя на константу, константного указателя, константного указателя на константу*.

Убрать ошибочные операторы, дописать операторы вывода и выполнить программы на компьютере.

Написать программу генерации элементов *массива* **А** из случайных чисел, их вывода и определения максимального элемента массива двумя способами (с указателями и без них).

#include <iostream>

using namespace std;

void main()

{

int a = 9;

const int\* pa = &a; // ошибка (нельзя менять значение переменной **а**)

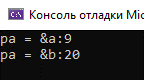
cout << "pa = &a:" << \*pa << endl;

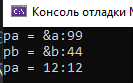
int b = 20;

pa = &b;

cout << "pa = &b:" << \*pa << endl;

}





#include <iostream>

using namespace std;

void main()

{

int a = 99;

int\* const pa = &a;

cout << "pa = &a:" << \*pa << endl;

int b = 44;

int\* const pb = &b;// ошибка (константный указатель ме-нять нельзя)

cout << "pb = &b:" << \*pb << endl;

\*pa = 12;

cout << "pa = 12:" << \*pa << endl;

}

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int n, i, A[N];

cout << "Введите размер массива: ";

cin >> n;

srand((unsigned)time(NULL));

for (i = 0; i < n; i++)

{

\*(A + i) = rand() % 99;

cout << A[i] << " ";

}

cout << endl;

int\* p = A;

int max1 = \*p;

for (i = 0; i < n; i++) {

if (\*(p + i) > max1) {

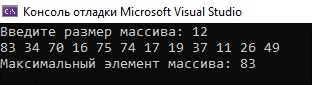
max1 = \*(p + i);

}

}

cout << "Максимальный элемент массива: " << max1 << endl;

}



#include <iostream>

using namespace std;

void main()

{

int a = 99;

const int\* const pa = &a;

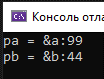
cout << "pa = &a:" << \*pa << endl; // ошибка (нельзя менять содержимое переменной а)

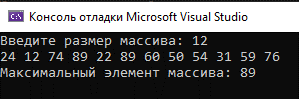
int b = 44;

const int\* const pb = &b; // ошибка (константный указатель на константу менять нельзя)

cout << "pb = &b:" << \*pb << endl;

}





#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int n, i, A[N];

cout << "Введите размер массива: ";

cin >> n;

srand((unsigned)time(NULL));

for (i = 0; i < n; i++)

{

A[i] = rand() % 99;

cout << A[i] << " ";

}

cout << endl;

int max=A[0];

for (i = 0; i < n; i++) {

if (A[i] > max) {

max=A[i];

}

}

cout << "Максимальный элемент массива: " << max << endl;

}

3. Выполнить программу, приведенную в правой части, которая разработана с использованием указателей.

Внести изменения с тем, чтобы программа стала содержать ошибки. Исследовать их с помощью отладки.

#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

int i, k, sz = 5;

float A[] = { 5, -4, 17.1, 9, 1 };

cout << "Введите номер элемента (от 0 до 4) " << endl;

cin >> k;

cout << endl;

for (i = k; i < sz - 1; i++)

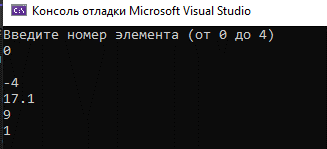
\*(A + i) = \*(A + i + 1);

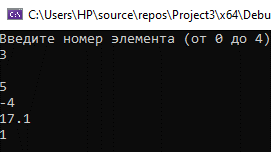
sz--;

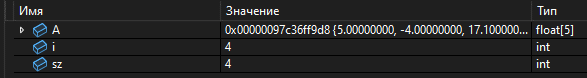
for (i = 0; i < sz; i++)

cout << \*(A + i) << endl;

}







#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

int i, k, sz = 5;

float A[] = { 5, -4, 17.1, 9, 1 };

cout << "Введите номер элемента (от 0 до 4) " << endl;

cin >> k;

cout << endl;

for (i = k; i < sz; i++)

\*(A + i) = \*(A + i + 1);

sz--;

for (i = 0; i < sz; i++)

cout << \*(A + i) << endl;

}

4. Изучить отличия ссылок от указателей. Выполнить программы, записанные в правой части.

Чтобы обратиться к данным по ссылке, достаточно использовать имя ссылки, как обычную переменную.

Для доступа к данным через указатель используется оператор разыменования ( \*), чтобы получить значение по адресу.



**#include <iostream>**

**void main()**

**{ int val = 1;**

**std::cout<<"value = "<<val;**

**int &rv = val;**

**rv = 5;**

**std::cout<<" value = "<<val;**

**}**



#include <iostream>

void main()

{

int val = 1;

std::cout << "value = " << val;

int& rv = val;

rv = 5;

int\* pval = &rv;

std::cout << " value = " << \*pval;

}

1. Разделить массив на две части, поместив в первую элементы, большие среднего арифметического их суммы, а во вторую − меньшие.

5

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int n, i;

float A[N];

float\* ptr = A;

float sum = 0;

cout << "Введите размер массива: ";

cin >> n;

cout << "Введите элементы массива: ";

for (int i = 0; i < n; i++) {

cin >> \*(ptr + i);

sum += \*(ptr + i);

}

float avg = sum / n;

cout << "Среднее арифметическое: " << avg << endl;

cout << "Элементы массива, которые больше среднего:";

for (i = 0; i < n; i++) {

if (\*(ptr + i) > avg) {

cout << \*(ptr + i) << " ";

}

}

cout << endl;

cout << "Элементы массива, которые меньше среднего:";

for (i = 0; i < n; i++) {

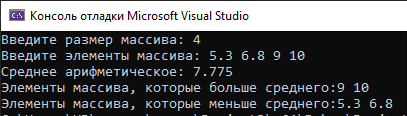
if (\*(ptr + i) < avg) {

cout << \*(ptr + i) << " ";

}

}

}



2. Даны два массива **x** и **y**. Найти количество одинаковых элементов в массивах **A** и **B**, т. е. количество пар **a[i]** **= b[j]** для некоторых **i** и **j**.

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int n, i;

float A[N];

float\* ptr = A;

cout << "Введите размер массива x: ";

cin >> n;

cout << "Введите элементы массива: ";

for ( i = 0; i < n; i++) {

cin >> \*(ptr + i);

}

int j;

float B[N];

float \* ptr1 = B;

cout << "Введите размер массива y: ";

cin >> j;

cout << "Введите элементы массива: ";

for (i = 0; i < j; i++) {

cin >> \*(ptr1 + i);

}

int count=0;

for (int i = 0; i < n; i++) {

for (int k = 0; k < j; k++) {

if (\*(ptr + i) == \*(ptr1 + k)) {

count++;

break;

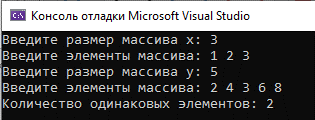
}

}

}

cout << "Количество одинаковых элементов: " << count << endl;

}



Дополнительные задания

1. Вводится последовательность из **n** натуральных чисел. Определить наименьшее натуральное число, отсутствующее в последовательности.

11

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int n, i;

int A[N];

int\* ptr = A;

cout << "Введите размер последовательности из натуральных чисел: ";

cin >> n;

cout << "Введите n натуральных чисел: ";

for ( i = 0; i < n; i++) {

cin >> \*(ptr + i);

if (\*(ptr + i) < 0) {

cout << "Введено некорректное число" << endl;

break;

}

}

int min = \*ptr;

for (i = 1; i < n; i++) {

if (\*(ptr + i) < min) {

min = \*(ptr + i);

}

if (min == 1) {

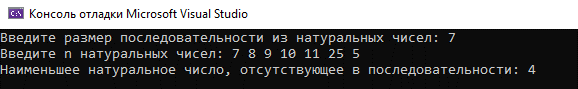
min++;

}

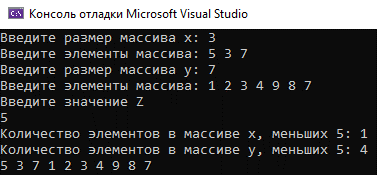
}

cout << "Наименьшее натуральное число, отсутствующее в последовательности: " << min-1 << endl;

}



2. В двух массивах подсчитать количество элементов, меньших значения **z**. Вывести первым массив, имеющий наименьшее их количество.



#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int n, i;

float A[N];

float\* ptr = A;

cout << "Введите размер массива x: ";

cin >> n;

cout << "Введите элементы массива: ";

for (i = 0; i < n; i++) {

cin >> \*(ptr + i);

}

int j;

float B[N];

float\* ptr1 = B;

cout << "Введите размер массива y: ";

cin >> j;

cout << "Введите элементы массива: ";

for (i = 0; i < j; i++) {

cin >> \*(ptr1 + i);

}

cout << "Введите значение Z" << endl;

int Z;

cin >> Z;

int countA=0;

for (i = 0; i < n; i++) {

if (\*(ptr + i) < Z) {

countA++;

}

}

cout << "Количество элементов в массиве x, меньших " << Z << ": " << countA << endl;

int countB=0;

for (i = 0; i < j; i++) {

if (\*(ptr1 + i) < Z) {

countB++;

}

}

cout << "Количество элементов в массиве y, меньших " << Z << ": " << countB << endl;

if (countA < countB) {

for (i = 0; i < n; i++) {

cout << \*(ptr + i) << " ";

}

for (i = 0; i < j; i++) {

cout << \*(ptr1 + i) << " ";

}

}

else if (countB < countA) {

for (i = 0; i < j; i++) {

cout << \*(ptr1 + i) << " ";

}

for (i = 0; i < n; i++) {

cout << \*(ptr + i) << " ";

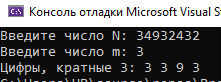
}

}

}

1. Ввести целое число N. Выделить из этого числа цифры, кратные m, и записать их в одномерный массив.

1



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Ru");

int N, m;

cout << "Введите число N: ";

cin >> N;

cout << "Введите число m: ";

cin >> m;

int result[100];

int count = 0;

while (N > 0) {

int digit = N % 10;

if (digit % m == 0) {

\*(result + count) = digit;

count++;

}

N /= 10;

}

cout << "Цифры, кратные " << m << ": ";

for (int i = 0; i < count; ++i) {

cout << \*(result + i) << " ";

}

return 0;

}

2. Заданы два массива по 10 целых чисел в каждом. Найти наибольшее среди чисел первого массива, которое не входит во второй массив.

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int i;

float A[N];

float\* ptr = A;

int n=10;

cout << "Введите элементы массива X: ";

for (i = 0; i < n; i++) {

cin >> \*(ptr + i);

}

float B[N];

float\* ptr1 = B;

int j=10;

cout << "Введите элементы массива Y: ";

for (i = 0; i < j; i++) {

cin >> \*(ptr1 + i);

}

int max = 0;

int found = 0;

for (i = 0; i < n; i++) {

int exists = 0;

for (int k = 0; k < j; k++) {

if (\*(ptr + i) == \*(ptr1 + k)) {

exists = 1;

break;

}

}

if (exists == 0 && \*(ptr + i) > max) {

max = \*(ptr + i);

found = 1;

}

}

if (found == 1) {

cout << "Наибольшее число из первого массива, которое не входит во второй: " << max << endl;

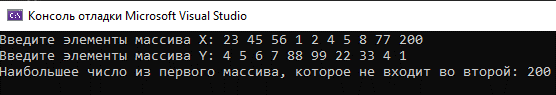
}

else {

cout << "Все числа первого массива входят во второй массив." << endl;

}

}



Дан массив **С**, не содержащий нулевых элементов. Преобразовать массив так, чтобы вначале шли отрицательные элементы, а затем положительные

6

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

const int N = 100;

int i,n;

int A[N];

int\* ptr = A;

cout << "Введите размер массива С:";

cin>>n;

cout << "Введите элементы массива: ";

for (i = 0; i < n; i++) {

cin >> \*(ptr + i);

}

// Сортировка массива: отрицательные элементы в начале, положительные в конце

int left = 0; // Указатель на начало массива

int right = n - 1; // Указатель на конец массива

while (left < right) {

// Ищем первый положительный элемент слева

while (left < n && \*(ptr + left) < 0) {

left++;

}

// Ищем первый отрицательный элемент справа

while (right > 0 && \*(ptr + right) > 0) {

right--;

}

// Меняем местами элементы, если указатели ещё не пересеклись

if (left < right) {

int temp = \*(ptr + left);

\*(ptr + left) = \*(ptr + right);

\*(ptr + right) = temp;

}

}

if (\*(ptr + i) != 0) {

// Вывод преобразованного массива

cout << "Преобразованный массив: ";

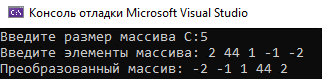
for (int i = 0; i < n; i++) {

cout << \*(ptr + i) << " ";

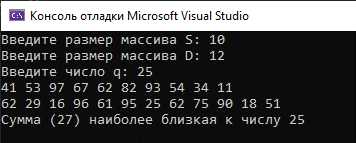
}

}

}



2.Дан массив **S**, содержащий **n1** элементов, массив **D**, содержащий **n2** элементов, и число **f**. Найти сумму вида **s[i] + d[j],** наиболее близкую к числу **f**.



#include <iostream>

void main()

{

setlocale(LC\_ALL, "Russian");

using namespace std;

const int max\_size = 99;

int n1, n2, f, sum, nearq, near = 9999;

float s[max\_size], d[max\_size];

cout << "Введите размер массива S: "; cin >> n1;

cout << "Введите размер массива D: "; cin >> n2;

cout << "Введите число q: "; cin >> f;

for (int i = 0; i < n1; i++)

{

\*(s + i) = rand() % 99;

cout << \*(s + i) << " " ;

}

cout << endl;

for (int j = 0; j < n2; j++)

{

\*(d + j) = rand() % 99;

cout << \*(d + j) <<" ";

}

cout << endl;

for (int i = 0; i < max(n1, n2); i++)

{

for (int j = 0; j < min(n1, n2); j++)

{

nearq = \*(s + i) + \*(d + j);

if (abs(nearq - f) < near)

{

near = abs(nearq - f);

sum = nearq;

}

}

}

cout << "Сумма (" << sum<<") " << " наиболее близкая к числу " << f << endl;

}

**Лабораторная работа № 13. Обработка символьной информации**

1. Изучить способы преобразования символов, выполнив программы в правой части.

Выполнить прокрутки программ и записать условия.

Опробовать программы с различным текстом.

1

**#include <iostream>**

**void main()**

**{ int num = 5;**

**char symb, new\_symbol = ' ';**

**symb = num + '0';**

**std::cout<<symb<<' ';**

**if (symb >= '0' && symb <= '9')**

**num = symb - '0';**

**std::cout<<num<<' ';**

**symb = 'b';**

**if (symb >= 'a' && symb <= 'z')**

**new\_symbol = symb - 'a' + 'A';**

**std::cout<<new\_symbol<<' ';**

**}**

Прокрутка:

symb = num + '0';  
Вывод: num  
 (symb >= '0' && symb <= '9')  
num = symb - '0';  
Вывод: num   
symb = 'b';  
(symb >= 'a' && symb <= 'z')  
new\_symbol = symb - 'a' + 'A';  
Вывод: new\_symbol  
Конец программы

Условие задачи:   
Преобразовать число в символ, в зависимости от их типа (цифра или буква), и вывести результаты преобразования



#include <iostream>

void main()

{

int num = 5;

char symb, new\_symbol = ' ';

char\* pc; pc = &symb;

\*pc = num + '0';  
std::cout << \*pc << ' ';

std::cout << symb << ' ';

if (symb >= '0' && symb <= '9')

num = symb - '0';

std::cout << num << ' ';

symb = 'b';

if (symb >= 'a' && symb <= 'z')

new\_symbol = symb - 'a' + 'A';

std::cout << new\_symbol << ' ';

}

Условие заадчи:  
При использовании указателя преобразуйте следующее число в символьное, зависимости от их диапазона , выводятся результаты преобразования



Прокрутка:  
pc = &symb;  
\*pc = num + '0';  
Вывод:\*pc  
Вывод:symb  
(symb >= '0' && symb <= '9')  
num = symb - '0';  
Вывод:num  
symb = 'b';  
(symb >= 'a' && symb <= 'z')  
new\_symbol = symb - 'a' + 'A';  
Вывод:new\_symbol

Условие задачи: С помощью указателя подсчитать количество символов в строке и вывести результат.

#include <iostream>

void main()

{

char str[] = "Text";

int count = 0;

char\* pstr;

pstr = &str[0];

if (str)

while (\*pstr++)

++count;

std::cout << count;

}



Прокрутка:  
pstr = &str[0];  
(str)  
 (\*pstr++)

++count;  
(\*pstr++)

++count;

(\*pstr++)

++count;

(\*pstr++)

++count;

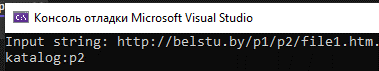
Вывод:cout

Выполнить программу, приведенную в правой части.

Объяснить принцип использования *функций стандартной библиотеки*.

2

принцип использования функций стандартной библиотеки: strlen()-Используется для расчета длины строки, printf- Форматированный вывод, strncpy\_s()-Применяется для копирования частей строк.



#include <cstring>

#include <iostream>

using namespace std;

void main()

{

char str[50], s[20];

int i, j, i1, i2;

cout << "Input string:";

cin >> str;

for (i = strlen(str); i >= 0; i--)

if (str[i] == '\/') break;

i2 = i;

for (j = (i2 - 1); j >= 0; j--)

if (str[j] == '\/') break;

i1 = j;

if (i2 == i1)

printf(" '\/' ");

else

{

strncpy\_s(s, &str[i1 + 1], i2 - i1 - 1);

s[i2 - i1 - 1] = 0;

cout << "katalog:" << s << "\n";

}

}

3

Изучить способы работы с символьными массивами, выполнив программу в правой части.

#include <iostream>

using namespace std;

void main()

{

char s, t[20];

int i, j, n;

cout << "Input string: ";

gets\_s(t);

cout << "Input symbol: ";

cin >> s;

n = strlen(t);

for (i = 0; i <= n; i++)

if (t[i] == s)

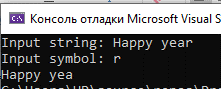
for (j = i; j <= n; j++)

t[i] = t[i + 1];

for (i = 0; i < n; i++)

cout << t[i];

}

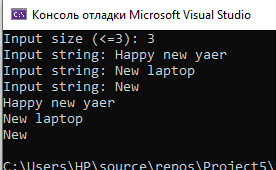


4

Выполнить программу в правой части.

Закомментировать строку **cin.ignore…**

Объяснить возникшие проблемы при выполнении программы.



#include <iostream>

using namespace std;

void main()

{

char t[3][20];

int i, n;

cout << "Input size (<=3): ";

cin >> n;

cin.ignore(cin.rdbuf()->in\_avail());// очищает буфер

for (i = 0; i < n; i++)

{

cout << "Input string: ";

gets\_s(t[i]);

}

for (i = 0; i < n; i++)

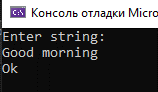
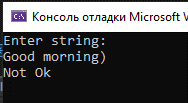
puts(t[i]);

}

В правой части записаны два варианта решения задачи. Выполнить программы и объяснить различия между ними.

Внести изменения в программу с тем, чтобы проверялось не только количество скобок, но и правильность их расстановки (первой в тексте должна быть открывающая скобка).

5

#include <stdio.h>

void main()

{

char s[256];

int i, count;

puts("Enter string: ");

gets\_s(s);

for (count = i = 0; s[i] != 0; i++)

{

if (s[i] == '(') count++;

if (s[i] == ')') count--;

}

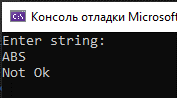
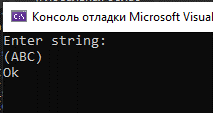
if (!count)

puts("Ok\n");

else

puts("Not Ok\n");

}

#include <stdio.h>

void main()

{

char s[256];

int i=0, count=0;

puts("Enter string: ");

gets\_s(s);

for (count > 0; s[i] != 0; i++)

{

if (s[i] == '(') count++;

if (s[i] == ')') count--;

}

if (!count && s[0] == '(')

puts("Ok\n");

else

puts("Not Ok\n");

}

**#include <stdio.h>**

**void main()**

**{**

**char s[256];**

**int count;**

**char \*ps;**

**puts("Enter string: ");**

**gets\_s(s);**

**for(count = 0, ps = s; \*ps != 0; ps++)**

**{**

**if (\*ps == '(') count++;**

**if (\*ps == ')') count--;**

**}**

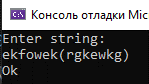
**if (count == 0)**

**puts("Ok\n");**

**else**

**puts("Not Ok\n");**

**}**



#include <stdio.h>

void main()

{

char s[256];

int count;

char\* ps;

puts("Enter string: ");

gets\_s(s);

for (count = 0, ps = s; \*ps != 0; ps++)

{

if (\*ps == '(') count++;

if (\*ps == ')') count--;

}

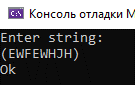
if (!count && s[0] == '(')

puts("Ok\n");

else

puts("Not Ok\n");

}



ВАРИАНТ 5

1. Исключить из строки группы символов, расположенные между скобками вместе со скобками. Предполагается, что нет вложенных скобок.

6

#include <stdio.h>

#include <iostream>

void main()

{

char s[256];

char out[256];

int r = 0;

int i, count;

puts("Enter string: ");

gets\_s(s);

for ( i = 0; s[i] != 0; i++)

{

if (s[i] == '(') {

while (s[i] != ')') {

++i;

}

}

else {

out[r++] = s[i];

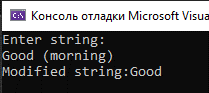
out[r] = '\0';

}

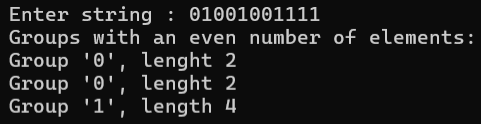
}

std::cout << "Modified string:" << out << std::endl;

}



2. Дана строка, состоящая из групп нулей и единиц. Найти группы с четным количеством символов.

****

#include <iostream>

void main()

{

using namespace std;

char s[256], current = '\0', \*ps;

int i, count;

cout << "Enter string : ";

cin >> s;

cout << "Groups with an even number of elements:\n";

for (count = 0, ps = s; \*ps != '\0'; ps++)

{

if (\*ps == current)

count++;

else {

if (count % 2 == 0 && count > 0)

cout << "Group '" << current << "', lenght " << count << endl;

current = \*ps;

count = 1;

}

}

if (count % 2 == 0 && count > 0)

cout << "Group '" << current << "', length " << count << endl;

}

Дополнительные задания:

1. Отредактировать заданное предложение, удаляя из него все слова с нечетными номерами и переворачивая слова с четными номерами. Пример: HOW DO YOU DO преобразовать в OD OD.

#include <iostream>

#include <Windows.h>

using namespace std;

void main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char s[200], word[100], result[200] = "";

int wordStart = 0, wordEnd = 0, wordLen, resLen = 0, wordCount = 0;

cout << "Enter text: ";

cin.getline(s, 200);

for (int i = 0; s[i] != '\0'; ++i) {

if (s[i] == ' ' || s[i + 1] == '\0') {

if (s[i] == ' ')

wordEnd = i;

else

wordEnd = i + 1;

wordLen = wordEnd - wordStart;

for (int j = 0; j < wordLen; ++j) {

word[j] = s[wordStart + j];

}

word[wordLen] = '\0';

wordCount++;

if (wordCount % 2 == 0) {

for (int j = wordLen - 1; j >= 0; j--) {

result[resLen++] = word[j];

}

result[resLen++] = ' ';

}

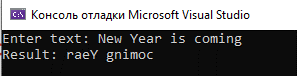
wordStart = wordEnd + 1;

}

}

cout << "Result: " << result << endl;

}



1. В заданной последовательности слов найти все слова, имеющие заданное окончание.

#include <cstring>

#include <iostream>

#include <Windows.h>

void main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

using namespace std;

char s[200], ending[20], word[100], result[200] = "";

int wordStart = 0, wordEnd = 0, wordLen, resLen = 0;

cout << "Enter words: ";

cin.getline(s, 100);

cout << "Introduces ending: ";

cin.getline(ending, 10);

for (int i = 0; s[i] != '\0'; ++i) {

if (s[i] == ' ' || s[i + 1] == '\0') {

if (s[i] == ' ')

wordEnd = i;

else

wordEnd = i + 1;

wordLen = wordEnd - wordStart;

for (int j = 0; j < wordLen; ++j) {

word[j] = s[wordStart + j];

}

word[wordLen] = '\0';

if (wordLen >= strlen(ending)) {

bool matches = false;

for (int j = 0; j < strlen(ending); ++j) {

if (word[wordLen - strlen(ending) + j] != ending[j]) {

matches = true;

break;

}

}

if (!matches) {

for (int j = 0; j < wordLen; ++j) {

result[resLen++] = word[j];

}

result[resLen++] = ' ';

}

}

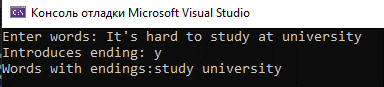
wordStart = wordEnd + 1;

}

}

cout << "Words with endings:" << result << endl;

}



Из заданного предложения удалить те слова, которые уже встречались в предложении раньше.

#include <iostream>

#include <Windows.h>

using namespace std;

void main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char s[200], word[100], result[200] = "";

int wordStart = 0, wordEnd = 0, resLen = 0, wordLen;

cout << "Enter text: ";

cin.getline(s, 100);

for (int i = 0; s[i] != '\0'; ++i) {

if (s[i] == ' ' || s[i + 1] == '\0') {

if (s[i] == ' ')

wordEnd = i;

else

wordEnd = i + 1;

wordLen = wordEnd - wordStart;

for (int j = 0; j < wordLen; ++j) {

word[j] = s[wordStart + j];

}

word[wordLen] = '\0';

bool isDuplicate = false;

int checkIndex = 0;

while (result[checkIndex] != '\0') {

int matchIndex = 0;

// Ищем слово в result

while (result[checkIndex] != '\0' && result[checkIndex] != ' ' &&

word[matchIndex] == result[checkIndex]) {

++matchIndex;

++checkIndex;

}

if (result[checkIndex] == ' ' || result[checkIndex] == '\0') {

if (matchIndex == wordLen) { // Совпадение

isDuplicate = true;

break;

}

}

while (result[checkIndex] != '\0' && result[checkIndex] != ' ') {

++checkIndex;

}

if (result[checkIndex] == ' ') {

++checkIndex;

}

}

if (!isDuplicate) {

for (int j = 0; j < wordLen; ++j) {

result[resLen++] = word[j];

}

result[resLen++] = ' ';

}

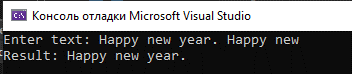
wordStart = wordEnd + 1;

}

}

cout << "Result: " << result << endl;

}



**Лабораторная работа № 14. Многомерные массивы**

Выполнить программу, записанную в правой части. Внести изменения в программу с тем, чтобы инициализировался другой массив. Осуществить вывод этого массива в виде матрицы.

1

#include <iostream>

void main()

{

const int ci = 3;

const int cj = 2;

int a[ci][cj] = { { 1, 2 }, { 3, 4 }, { 5, 6 } };

for (int i = 0; i < ci; i++) {

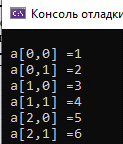
for (int j = 0; j < cj; j++) {

std::cout << "\n a[" << i << "," << j << "] =" << a[i][j];

}

}

}





#include <iostream>

void main()

{

const int ci = 3;

const int cj = 2;

int a[ci][cj] = { { 5, 6 }, { 7,8 }, { 9, 10 } };

for (int i = 0; i < ci; i++) {

for (int j = 0; j < cj; j++) {

std::cout << a[i][j] << " ";

}

std::cout << "\n";

}

}

Выполнить программы, записанные в правой части для одной и той же задачи, условие которой надо определить. Одна из программ использует указатели.

2

Условие задачи:  
Дан двумерный массив размером n×м заполненный целыми числами. Выведите промежуточную сумму после каждого шага.

#include <iostream>

void main()

{

const int n = 3, m = 2;

int h = 0;

int A[n][m] = { 0, 1, 2, 3, 4, 5 };

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

h += A[i][j];

std::cout << h<<" ";

}

}

}



#include <iostream>

void main()

{

const int n = 3, m = 2;

int h = 0;

int C[n][m] = { 0, 1, 2, 3, 4, 5 };

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

h += \*(\*(C + i) + j);

std::cout << h<< " ";

}

}

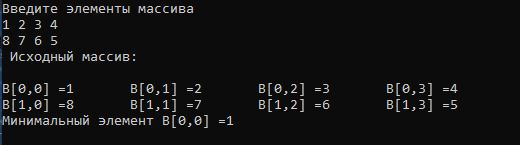
}



Изучить способы работы с двумерными массивами, выполнив программу в правой части и записав ее условие.

Внести изменения в программу с тем, чтобы определялись минимальные элементы каждой строки. Для доступа к элементам матрицы использовать указатели.

3



#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

const int n = 2, m = 4;

int B[n][m];

int i, j, row = 0, column = 0;

cout << "Введите элементы массива" << endl;

for (i = 0; i < n; i++)

for (j = 0; j < m; j++)

cin >> B[i][j];

int min = B[0][0];

for (int i = 0; i < n; i++)

for (int j = 0; j < m; j++)

if (min > B[i][j])

{

min = B[i][j];

column = i;

row = j;

}

cout << " Исходный массив:" << endl;

for (int i = 0; i < n; i++)

{

cout << "\n";

for (int j = 0; j < m; j++)

cout << "B[" << i << "," << j << "] =" << B[i][j] << "\t";

}

cout << endl;

cout << "Минимальный элемент B[" << column << "," << row << "] =" << min << endl;

}

Условие задачи:  
Пользователь вводит элементы массива  
Переменная min преобразуется в значение первого элемента массива B[0][0]  
программа сравнивает текущее значение с min  
Если найден элемент, меньший, чем min, он становится новым значением min, индексы строк ( column) и столбца ( row) обновляются  
Выводится минимальное значение.

#include <iostream>

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

const int n = 2, m = 4;

int B[n][m];

int\* pB = &B[0][0];

cout << "Введите элементы массива:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cin >> \*(pB + i \* m + j); // Используем указатель для доступа к элементам

}

}

// Вывод исходного массива

cout << "\nИсходный массив:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cout << "B[" << i << "," << j << "] = " << \*(pB + i \* m + j) << "\t";

}

cout << endl;

}

cout << endl;

for (int i = 0; i < n; i++) {

int min = \*(pB + i \* m); // Инициализируем минимальный элемент первого столбца строки

for (int j = 1; j < m; j++) {

if (min > \*(pB + i \* m + j)) {

min = \*(pB + i \* m + j); // Сравнение с элементом текущего столбца

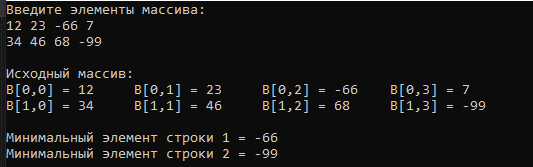
}

}

cout << "Минимальный элемент строки " << i+1 << " = " << min << endl;

}

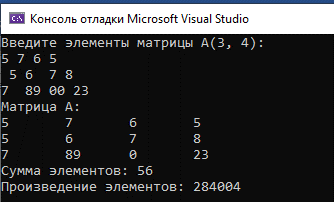
}



Вариант 5

1. Дана целочисленная матрица **A(n, m)**. Вычислить сумму и произведение тех её элементов, которые при делении на два дают нечётное число.

4



#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

const int n = 3;

const int m = 4;

int A[n][m];

int sum = 0;

int proiz = 1;

int count = 0;

cout << "Введите элементы матрицы A(" << n << ", " << m << "):" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cin >> A[i][j];

}

}

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

if ((A[i][j] / 2) % 2 == 1) {

sum += A[i][j];

proiz \*= A[i][j];

count++;

}

}

}

cout << "Матрица A:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cout << A[i][j] << "\t";

}

cout << endl;

}

if (count > 0) {

cout << "Сумма элементов: " << sum << endl;

cout << "Произведение элементов: " << proiz << endl;

}

else {

cout << "Подходящих элементов в матрице нет." << endl;

}

return 0;

}

2. Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент, и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

const int n = 3;

const int m = 4;

int A[n][m];

cout << "Введите элементы матрицы A(3,4):" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cin >> A[i][j];

}

}

int rowWithPositive = -1;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

if (A[i][j] > 0) {

rowWithPositive = i;

break;

}

}

if (rowWithPositive != -1) break;

}

if (rowWithPositive > 0) {

for (int j = 0; j < m; j++) {

A[rowWithPositive - 1][j] = -A[rowWithPositive - 1][j];

}

}

cout << "Матрица после обработки:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cout << A[i][j] << "\t";

}

cout << endl;

}

if (rowWithPositive != -1) {

cout << "Cтрока с положительным элементом имеет номер : " << rowWithPositive + 1 << endl;

}

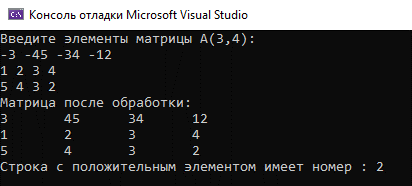
else {

cout << "В матрице нет строк с положительными элементами." << endl;

}

return 0;

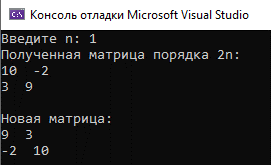
}



Дополнительные задания

Дана квадратная матрица порядка **2n**, элементы которой фо рмируются случайным образом и находятся в пределах от −10 до 10. Получить новую матрицу, переставляя ее блоки размера **n×n** в соответствии со схемой.

1



­ #include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

float A[50][50], elements[50 \* 50], temp;

int i, j, n, buff;

cout << "Введите n: ";

cin >> n;

cout << "Полученная матрица порядка 2n: \n";

for (i = 0; i < 2 \* n; i++)

{

for (j = 0; j < 2 \* n; j++)

{

A[i][j] = (rand() % 21) - 10;

cout << A[i][j] << " ";

}

cout << endl;

}

for (i = 0; i < n; ++i)

{

for (j = 0; j < n; ++j)

{

buff = A[i][j];

A[i][j] = A[n + i][n + j];

A[n + i][n + j] = buff;

buff = A[i][n + j];

A[i][n + j] = A[n + i][j];

A[n + i][j] = buff;

}

}

cout <<endl<< "Новая матрица: \n";

for (i = 0; i < 2 \* n; i++)

{

for (j = 0; j < 2 \* n; j++)

{

cout << A[i][j] << " ";

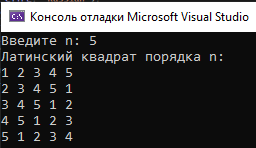
}

cout << endl;

}

}

2. Латинским квадратом порядка **n** называется квадратная таблица размером **nхn**, каждая строка и каждый столбец которой содержат все числа от 1 до **n**. Для заданного **n** в матрице **L(n, n**) построить латинский квадрат порядка **n**.



#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

int L[50][50];

int i, j, n;

cout << "Введите n: " << endl; cin >> n;

cout << "Латинский квадрат порядка n: \n";

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

L[i][j] = (i + j) % n + 1;

cout << L[i][j] << " ";

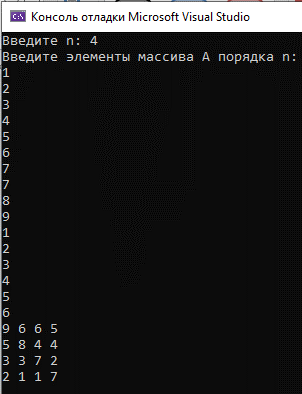
}

cout << endl;

}

}

3. Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине − в позиции (2, 2), следующий − в позиции (3, 3) и т. д., заполнив таким образом всю главную диагональ.



#include <iostream>

void main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

float A[50][50], elements[50 \* 50], temp;

int i, j, n, k = 0;

cout << "Введите n: " ;

cin >> n;

cout << "Введите элементы массива А порядка n: \n";

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

cin >> A[i][j];

elements[k++] = A[i][j];

}

}

for (i = 0; i < k - 1; ++i)

{

for (j = i + 1; j < k; ++j)

{

if (elements[i] < elements[j])

{

temp = elements[i];

elements[i] = elements[j];

elements[j] = temp;

}

}

}

for (i = k = 0; i < n; i++)

{

A[i][i] = elements[k++];

}

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

if (i != j)

{

A[i][j] = elements[k++];

}

}

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

cout << A[i][j] << " ";

}

cout << endl;

}

}

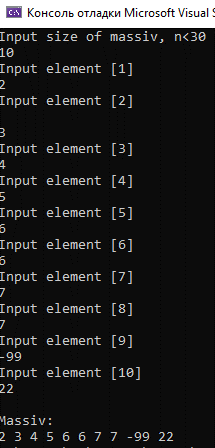
Лабораторная работа № 15. Динамическое выделение памяти

1. Изучить формирование динамического массива с использованием стандартных функций, выполнив программу на *языке* ***С***, записанную справа.

Написать пояснения к программе.

Добавить операторы вычисления максимального по модулю элемента массива.

1



#include <stdio.h>

#include <stdlib.h> // Библиотека для работы с динамической памятью (malloc, free)

void main()

{

int\* ptr, i, n; // Указатель ptr для динамического массива, переменные i (индекс) и n (размер массива)

// Ввод размера массива

printf("Input size of massiv, n<30 \n"); // Сообщение пользователю о вводе размера массива (до 30 элементов)

scanf\_s("%d", &n); // Чтение размера массива из ввода

// Выделение памяти под массив и проверка, успешно ли выделена память

if (!(ptr = (int\*)malloc(n \* sizeof(int)))) {

// malloc выделяет память для n элементов типа int

// Если malloc вернёт NULL, это значит, что памяти не хватает, и условие выполнится

puts("Not enough memory"); // Сообщение об ошибке: недостаточно памяти

return; // Завершение работы программы

}

// Ввод элементов массива

for (i = 0; i < n; i++) { // Цикл для ввода n элементов

printf("Input element [%d]\n", i + 1); // Сообщение о вводе элемента с указанием его номера

scanf\_s("%d", ptr + i); // Чтение элемента в массив (с использованием указателя)

// ptr + i указывает на адрес i-го элемента

}

// Вывод элементов массива

printf("\nMassiv: \n"); // Заголовок для вывода массива

for (i = 0; i < n; i++) { // Цикл для вывода n элементов

printf("%d ", \*(ptr + i)); // Вывод i-го элемента массива (разыменование указателя)

}

// Освобождение динамической памяти

free(ptr); // Освобождение памяти, выделенной под массив

// После вызова free память, выделенная malloc, освобождается, чтобы её мог использовать другой код

}

#include <stdio.h>

#include <stdlib.h> // Библиотека для работы с динамической памятью (malloc, free)

#include <math.h> // Для использования функции abs()

void main()

{

int\* ptr, i, n; // Указатель ptr для динамического массива, переменные i (индекс) и n (размер массива)

int maxMod = 0; // Переменная для хранения максимального по модулю элемента массива

// Ввод размера массива

printf("Input size of massiv, n<30 \n");

scanf\_s("%d", &n); // Чтение размера массива

// Выделение памяти под массив и проверка, успешно ли выделена память

if (!(ptr = (int\*)malloc(n \* sizeof(int)))) {

puts("Not enough memory"); // Сообщение об ошибке, если памяти недостаточно

return; // Завершение работы программы

}

// Ввод элементов массива

for (i = 0; i < n; i++) { // Цикл для ввода n элементов

printf("Input element [%d]\n", i + 1); // Сообщение о вводе элемента

scanf\_s("%d", ptr + i); // Чтение элемента в массив с использованием указателя

}

// Вычисление максимального по модулю элемента массива

maxMod = abs(ptr[0]); // Начинаем с первого элемента, берём его абсолютное значение

for (i = 1; i < n; i++) { // Цикл для обработки всех оставшихся элементов массива

if (abs(ptr[i]) > maxMod) { // Если модуль текущего элемента больше текущего максимума

maxMod = abs(ptr[i]); // Обновляем максимальное значение

}

}

// Вывод элементов массива

printf("\nMassiv: \n"); // Заголовок для вывода массива

for (i = 0; i < n; i++) { // Цикл для вывода всех элементов массива

printf("%d ", \*(ptr + i)); // Вывод каждого элемента (разыменование указателя)

}

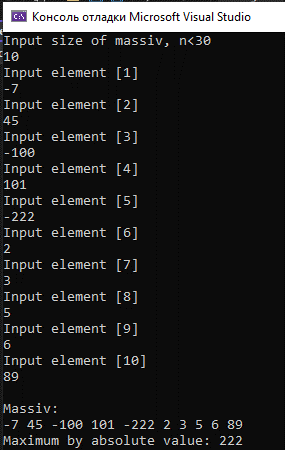
// Вывод максимального по модулю элемента

printf("\nMaximum by absolute value: %d\n", maxMod);

// Освобождение динамической памяти

free(ptr); // Освобождение памяти, выделенной под массив

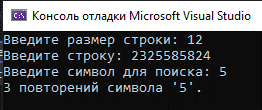
}

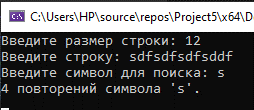


Изучить способы выделения динамической памяти для *одномерного* *массива*, выполнив программу на *языке* ***С++***, записанную в правой части.

Опробовать работу программы с разными значениями вводимых символов.

2





#include <iostream>

using namespace std;

void main() {

setlocale(LC\_CTYPE, "Russian");

char\* pStr;

char c;

int j, length, count = 0;

// Ввод размера строки

cout << "Введите размер строки: ";

cin >> length;

// Выделение памяти под строку

pStr = new char[length + 1]; // +1 для символа конца строки

// Ввод строки

cout << "Введите строку: ";

cin >> pStr; // Ввод строки

// Ввод символа для поиска

cout << "Введите символ для поиска: ";

cin >> c; // Ввод символа

// Подсчет повторений символа

for (j = 0; j < length; j++) {

if (pStr[j] == c) {

count++; // Увеличиваем счетчик при совпадении

}

}

// Вывод результата

cout << count << " повторений символа '" << c << "'." << endl;

// Освобождение памяти

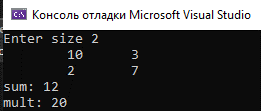
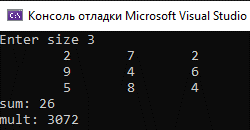
delete[] pStr;

}

В программе, записанной справа, демонстрируется использование динамической памяти при работе с *двумерным массивом*.

Выполнить программу с различными размерами исходного массива.

3

**#include <ctime>**

**#include <iostream>**

**using namespace std;**

**void main()**

**{ int size, mult = 1, sum = 0, \*\*A;**

**cout << "Enter size ";**

**cin >> size;**

**srand(time(0));**

**A = new int\*[size]; //выделение динам.памяти**

**for (int i = 0; i < size; i++)**

**A[i] = new int[size];**

**for (int j = 0; j < size; j++)**

**A[i][j] = 1 + rand() % 10;**

**for (int \*\*ptr = A; ptr != A + size; ++ptr)**

**{ for (int\* it = \*ptr; it != \*ptr + size; ++it)**

**{ cout << "\t" << \*it;**

**if (\*it % 2 == 0)**

**{ mult \*= \*it;**

**sum += \*it;**

**}**

**}**

**cout << endl;**

**}**

**cout << "sum: " << sum << endl;**

**cout << "mult: " << mult << endl;**

**for(int k=0; k < size; k++) //освобождение памяти**

**delete[] A[k];**

**delete[] A;**

**}**

Задание 4 (Вариант 5)

В одномерном массиве, состоящем из **n** вещественных элементов, вычислить количество элементов массива, равных 0, и сумму элементов массива, расположённых после минимального элемента.

1

#include <stdio.h>

#include <stdlib.h>

void main()

{

int\* ptr, i, n, zeroCount = 0, minIndex = 0;

double min, sumAfterMin = 0;

printf("Input size of array: ");

scanf\_s("%d", &n);

if (!(ptr = (int\*)malloc(n \* sizeof(int))))

{

puts("Not enough memory");

return;

}

for (i = 0; i < n; i++)

{

printf("Input element [%d]: ", i + 1);

scanf\_s("%d", ptr + i);

}

for (i = 0; i < n; i++)

{

if (\*(ptr + i) == 0)

{

zeroCount++;

}

}

// Find minimum element and its index

min = \*(ptr);

for (i = 1; i < n; i++)

{

if (\*(ptr + i) < min)

{

min = \*(ptr + i);

minIndex = i;

}

}

// Calculate sum of elements after the minimum element

for (i = minIndex + 1; i < n; i++)

{

sumAfterMin += \*(ptr + i);

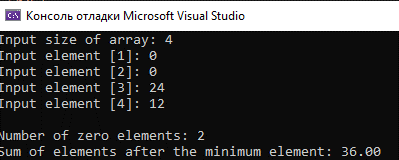
}

printf("\nNumber of zero elements: %d", zeroCount);

printf("\nSum of elements after the minimum element: %.2f", sumAfterMin);

free(ptr);

}



2. Дана целочисленная прямоугольная матрица. Определить количество столбцов, содержащих хотя бы один нулевой элемент и номер строки, в которой находится самая длинная серия одинаковых элементов.

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int rows, cols, nonZeroColumns = 0, maxSeriesRow = -1, maxSeriesLength = 0;

int\*\* matrix;

cout << "Введите количество строк: ";

cin >> rows;

cout << "Введите количество столбцов: ";

cin >> cols;

// Выделение памяти для матрицы

matrix = new int\* [rows];

for (int i = 0; i < rows; i++)

{

matrix[i] = new int[cols];

}

// Ввод элементов матрицы

cout << "Введите элементы матрицы: \n";

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

cin >> matrix[i][j];

}

}

// Подсчёт столбцов без нулевых элементов

for (int j = 0; j < cols; j++)

{

bool hasZero = false;

for (int i = 0; i < rows; i++)

{

if (matrix[i][j] == 0)

{

hasZero = true;

break;

}

}

if (!hasZero)

{

nonZeroColumns++;

}

}

// Поиск строки с самой длинной серией одинаковых элементов

for (int i = 0; i < rows; i++)

{

int currentSeriesLength = 1;

for (int j = 1; j < cols; j++)

{

if (matrix[i][j] == matrix[i][j - 1])

{

currentSeriesLength++;

}

else

{

if (currentSeriesLength > maxSeriesLength)

{

maxSeriesLength = currentSeriesLength;

maxSeriesRow = i;

}

currentSeriesLength = 1;

}

}

if (currentSeriesLength > maxSeriesLength) // Проверка последней серии в строке

{

maxSeriesLength = currentSeriesLength;

maxSeriesRow = i;

}

}

cout << "Количество столбцов, не содержащих ни одного нулевого элемента: " << nonZeroColumns << endl;

if (maxSeriesRow != -1)

{

cout << "Номер строки с самой длинной серией одинаковых элементов: " << maxSeriesRow + 1 << endl;

}

else

{

cout << "В матрице нет одинаковых элементов, составляющих серию." << endl;

}

// Освобождение памяти

for (int i = 0; i < rows; i++)

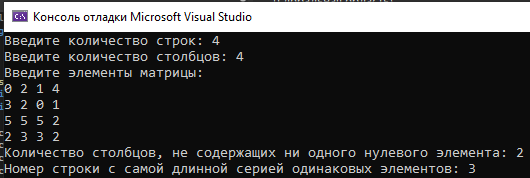
{

delete[] matrix[i];

}

delete[] matrix;

}



Дополнительные задания

1. В одномерном массиве, состоящем из **n** вещественных элементов, вычислить номер минимального элемента массива и сумму элементов массива, расположенных между первым и вторым отрицательными элементами.

-1

#include <stdio.h>

#include <stdlib.h>

void main()

{

int\* ptr, i, n, minIndex = 0, firstNegative = -1, secondNegative = -1;

double min, sumBetweenNegatives = 0;

printf("Input size of array: ");

scanf\_s("%d", &n);

if (!(ptr = (int\*)malloc(n \* sizeof(int))))

{

puts("Not enough memory");

return;

}

for (i = 0; i < n; i++)

{

printf("Input element [%d]: ", i + 1);

scanf\_s("%d", ptr + i);

}

for (i = 0; i < n; i++)

// Find minimum element and its index

min = \*(ptr);

for (i = 1; i < n; i++)

{

if (\*(ptr + i) < min)

{

min = \*(ptr + i);

minIndex = i;

}

}

// Find indices of first and second negative elements

for (i = 0; i < n; i++)

{

if (\*(ptr + i) < 0)

{

if (firstNegative == -1)

{

firstNegative = i;

}

else if (secondNegative == -1)

{

secondNegative = i;

break;

}

}

}

// Calculate sum between first and second negative elements

if (firstNegative != -1 && secondNegative != -1)

{

for (i = firstNegative + 1; i < secondNegative; i++)

{

sumBetweenNegatives += \*(ptr + i);

}

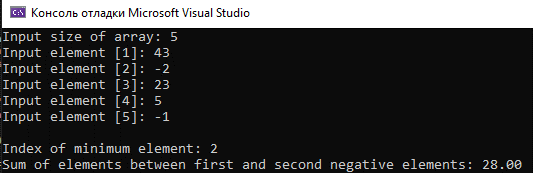
}

printf("\nIndex of minimum element: %d", minIndex+1);

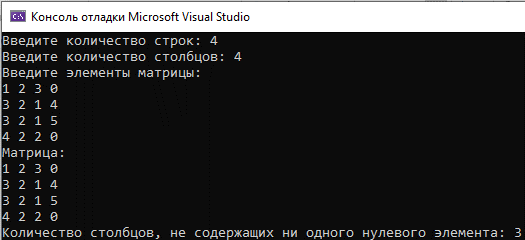
printf("\nSum of elements between first and second negative elements: %.2f", sumBetweenNegatives);

free(ptr);

}



2. Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента.



#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int rows, cols, nonZeroColumns = 0;

int\*\* matrix;

cout << "Введите количество строк: ";

cin >> rows;

cout << "Введите количество столбцов: ";

cin >> cols;

// Выделение памяти для матрицы

matrix = new int\* [rows];

for (int i = 0; i < rows; i++)

{

matrix[i] = new int[cols];

}

// Ввод элементов матрицы

cout << "Введите элементы матрицы: \n";

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

cin >> matrix[i][j];

}

}

// Подсчёт столбцов без нулевых элементов

for (int j = 0; j < cols; j++)

{

bool hasZero = false;

for (int i = 0; i < rows; i++)

{

if (matrix[i][j] == 0)

{

hasZero = true;

break;

}

}

if (!hasZero)

{

nonZeroColumns++;

}

}

// Вывод матрицы

cout << "Матрица: \n";

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

cout << matrix[i][j] << " ";

}

cout << endl;

}

cout << "Количество столбцов, не содержащих ни одного нулевого элемента: " << nonZeroColumns << endl;

// Освобождение памяти

for (int i = 0; i < rows; i++)

{

delete[] matrix[i];

}

delete[] matrix;

}

1. Задан массив **A** из **n** элементов. Подсчитать, сколько раз встречается в нем максимальное число

12

#include <stdio.h>

#include <stdlib.h>

void main()

{

int\* ptr, i, n, maxCount = 0;

double max;

printf("Input size of array: ");

scanf\_s("%d", &n);

if (!(ptr = (int\*)malloc(n \* sizeof(int))))

{

puts("Not enough memory");

return;

}

for (i = 0; i < n; i++)

{

printf("Input element [%d]: ", i + 1);

scanf\_s("%d", ptr + i);

}

for (i = 0; i < n; i++)

// Find maximum element

max = \*(ptr);

for (i = 1; i < n; i++)

{

if (\*(ptr + i) > max)

{

max = \*(ptr + i);

}

}

// Count occurrences of the maximum element

for (i = 0; i < n; i++)

{

if (\*(ptr + i) == max)

{

maxCount++;

}

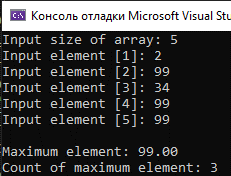
}

printf("\nMaximum element: %.2f", max);

printf("\nCount of maximum element: %d", maxCount);

free(ptr);

}



#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int rows, cols, positiveRow = -1;

int\*\* matrix;

cout << "Введите количество строк: ";

cin >> rows;

cout << "Введите количество столбцов: ";

cin >> cols;

// Выделение памяти для матрицы

matrix = new int\* [rows];

for (int i = 0; i < rows; i++)

{

matrix[i] = new int[cols];

}

// Ввод элементов матрицы

cout << "Введите элементы матрицы: \n";

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

cin >> matrix[i][j];

}

}

// Проверка на наличие строки с положительным элементом

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

if (matrix[i][j] > 0)

{

positiveRow = i;

break;

}

}

if (positiveRow != -1)

{

break;

}

}

// Изменение знаков элементов предыдущей строки

if (positiveRow > 0)

{

for (int j = 0; j < cols; j++)

{

matrix[positiveRow - 1][j] = -matrix[positiveRow - 1][j];

}

}

// Вывод матрицы

cout << "Матрица после изменений: \n";

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

cout << matrix[i][j] << " ";

}

cout << endl;

}

if (positiveRow != -1)

{

cout << "Номер первой строки, содержащей положительный элемент: " << positiveRow + 1 << endl;

}

else

{

cout << "В матрице нет строки с положительными элементами." << endl;

}

// Освобождение памяти

for (int i = 0; i < rows; i++)

{

delete[] matrix[i];

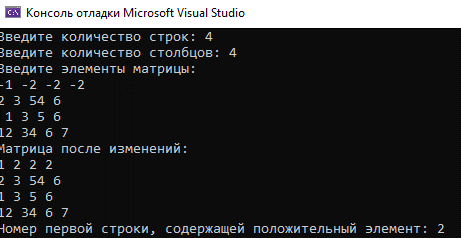
}

delete[] matrix;

}

3

2. Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент, и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные.



1. Задан массив **A** из **n** элементов. Найти количество элементов этого массива, больших среднего арифметического всех его элементов.

7

#include <stdio.h>

#include <iostream>

#include <stdlib.h>

void main()

{

setlocale(LC\_CTYPE, "Russian");

int\* ptr, i, k, sr, sum = 0, el = 0;

printf("Input size of massiv: ");

scanf\_s("%d", &k);

if (!(ptr = (int\*)malloc(k \* sizeof(int))))

{

puts("Not enough memory");

return;

}

for (i = 0; i < k; i++)

{

printf("Input element [%d]: ", i + 1);

scanf\_s("%d", ptr + i);

}

printf("\nMassiv: \n");

for (i = 0; i < k; i++)

{

sum += \*(ptr + i);

printf("%d ", \*(ptr + i));

}

sr = sum / i;

for (i = 0; i < k; i++)

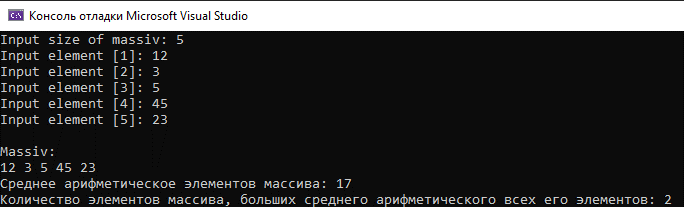
if (\*(ptr + i) > sr)

el++;

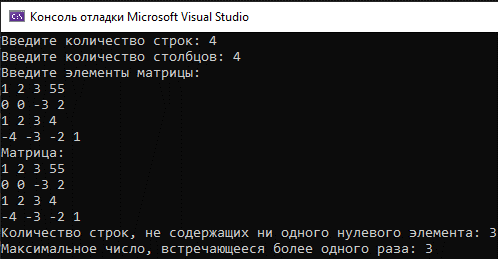
printf("\nСреднее арифметическое элементов массива: %d\nКоличество элементов массива, больших среднего арифметического всех его элементов: %d", sr, el);

free(ptr);

}



2. Дана целочисленная прямоугольная матрица. Определить количество строк, не содержащих ни одного нулевого элемента и максимальное из чисел, встречающихся в заданной матрице более одного раза



­ #include <iostream>

#include <map>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int rows, cols, nonZeroRows = 0, maxRepeated = INT\_MIN;

int\*\* matrix;

map<int, int> frequency;

cout << "Введите количество строк: ";

cin >> rows;

cout << "Введите количество столбцов: ";

cin >> cols;

// Выделение памяти для матрицы

matrix = new int\* [rows];

for (int i = 0; i < rows; i++)

{

matrix[i] = new int[cols];

}

// Ввод элементов матрицы

cout << "Введите элементы матрицы: \n";

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

cin >> matrix[i][j];

frequency[matrix[i][j]]++;

}

}

// Подсчет строк без нулевых элементов

for (int i = 0; i < rows; i++)

{

bool hasZero = false;

for (int j = 0; j < cols; j++)

{

if (matrix[i][j] == 0)

{

hasZero = true;

break;

}

}

if (!hasZero)

{

nonZeroRows++;

}

}

// Нахождение максимального числа, встречающегося более одного раза

for (const auto& pair : frequency)

{

if (pair.second > 1 && pair.first > maxRepeated)

{

maxRepeated = pair.first;

}

}

// Вывод результатов

cout << "Матрица: \n";

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

cout << matrix[i][j] << " ";

}

cout << endl;

}

cout << "Количество строк, не содержащих ни одного нулевого элемента: " << nonZeroRows << endl;

if (maxRepeated != INT\_MIN)

{

cout << "Максимальное число, встречающееся более одного раза: " << maxRepeated << endl;

}

else

{

cout << "В матрице нет чисел, встречающихся более одного раза." << endl;

}

// Освобождение памяти

for (int i = 0; i < rows; i++)

{

delete[] matrix[i];

}

delete[] matrix;

}

Изучить способы передачи значений *переменных* в функцию, выполнив программы, записанные в правой части.

Записать условие задачи.

**Лабораторная работа № 16. Функции пользователя**

1

Условие задачи: функцию square, которая принимает значение числа y в качестве аргумента и возвращает его квадрат. Используйте эту функцию, вы вводите квадраты чисел от 1 до 10 последовательно.

#include <iostream>

using namespace std;

int square(int);

void main()

{

for (int x = 1; x <= 10; x++)

cout << square(x) << ' ';

}

int square(int y)

{

return y \* y;

}





#include <iostream>

using namespace std;

int square(int\*);

void main()

{

for (int x = 1; x <= 10; x++)

cout << square(&x) << ' ';

}

int square(int\* y)

{

return \*y \* \*y;

}



#include <iostream>

using namespace std;

int square(int&);

void main()

{

for (int x = 1; x <= 10; x++)

cout << square(x) << ' ';

}

int square(int& y)

{

return y \* y;

}

Изучить использование *указателей* и *ссылок* как *формальных параметров* функции при работе с *одномерным* *массивом данных*, опробовав работу программ в правой части.

Написать условие задачи.

2



#include <iostream>

using namespace std;

int sum(int x[]) // определение функции

{

int res = 0;

for (int i = 0; i < 5; i++)

res += x[i];

return res;

}

int main()

{

int mas[5], i;

for (i = 0; i < 5; i++)

{

mas[i] = i \* i;

cout << mas[i] << ' ';

}

cout << endl << "result=" << sum(mas) << endl;

}

Условие задачи:  
Создаёт массив из 5 элементов, элементы которых равны квадратам их индекса  
Выводит элементы массива на экране через пробел  
функцию sum, принимает массив в качестве аргумента, вычисляет сумму всех его элементов и получает результат.

#include <iostream>

using namespace std;

int sum(int& x)

{

int\* p = &x, res = 0;

for (int i = 0; i < 5; i++)

res += p[i];

return res;

}

int main()

{

int mas[5], i;

for (i = 0; i < 5; i++)

{

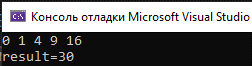
mas[i] = i \* i;

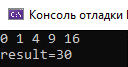
cout << mas[i] << ' ';

}

cout << endl << "result=" << sum(\*mas) << endl;

}





#include <iostream>

using namespace std;

int sum(int\* x) // определение функции

{

int res = 0;

for (int i = 0; i < 5; i++)

res += x[i];

return res;

}

int main()

{

int mas[5], i;

for (i = 0; i < 5; i++)

{

mas[i] = i \* i;

cout << mas[i] << ' ';

}

cout << endl << "result=" << sum(mas) << endl;

}

В программе, записанной справа, используется меню, разработанное с помощью оператора **switch**, которое позволяет делать выбор между двумя функциями.

Написать условие задачи и комментарии к программе.

3

#include <iostream>

using namespace std;

// Прототип функции для обработки одномерного массива

int massiv(int\* M, int);

// Прототип функции для обработки матрицы

void matrix();

int main(void)

{

setlocale(LC\_CTYPE, "Russian");

int choice; // Переменная для выбора действия

do

{

cout << "Выберите вариант работы" << endl;

cout << "1 - с одномерным массивом" << endl;

cout << "2 - с матрицей" << endl;

cout << "3 - выход" << endl;

cin >> choice; // Пользовательский выбор

switch (choice)

{

case 1: // Работа с одномерным массивом

int\* M, size, i, av;

cout << "Введите размер массива ";

cin >> size;

// Выделение памяти для массива

M = new int[size];

// Ввод элементов массива

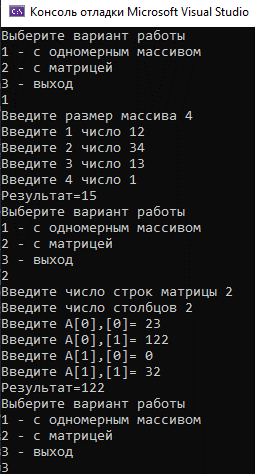
for (i = 0; i < size; i++)

{

cout << "Введите " << i + 1 << " число ";

cin >> \*(M + i);

}



// Вычисление среднего значения элементов массива

av = massiv(M, size);

cout << "Результат=" << av << endl;

// Освобождение памяти

delete[] M;

break;

case 2: // Работа с матрицей

matrix();

break;

case 3: // Выход из программы

break;

}

} while (choice != 3); // Цикл продолжается, пока не выбран выход

}

// Функция для работы с одномерным массивом

int massiv(int\* M, int size)

{

int sum = 0, avar;

// Суммирование элементов массива

for (int i = 0; i < size; i++)

sum += \*(M + i);

// Вычисление среднего значения

avar = sum / size;

return avar;

}

// Функция для работы с матрицей

void matrix()

{

int\*\* A, row, col, i, j, m;

// Ввод размеров матрицы

cout << "Введите число строк матрицы ";

cin >> row;

cout << "Введите чиcло столбцов ";

cin >> col;

// Выделение памяти для строк матрицы

A = new int\* [row];

for (i = 0; i < row; i++)

A[i] = new int[col]; // Выделение памяти для столбцов

// Ввод элементов матрицы

for (i = 0; i < row; i++)

for (j = 0; j < col; j++)

{

cout << "Введите A[" << i << "],[" << j << "]= ";

cin >> \*(\*(A + i) + j);

}

// Нахождение максимального элемента матрицы

m = A[0][0];

for (i = 0; i < row; i++)

for (j = 0; j < col; j++)

if (\*(\*(A + i) + j) > m)

m = \*(\*(A + i) + j);

cout << "Результат=" << m << endl;

// Освобождение памяти

for (i = 0; i < row; i++)

delete[] A[i]; // Освобождение памяти для строк

delete[] A; // Освобождение памяти для массива указателей

}

ВАРИАНТ 5

1. Дан двумерный массив.  В каждой строке выбрать минимальный элемент и среди них максимальный.  Напечатать номер строки, в которой расположен этот элемент.  
2. Написать программу, которая вводит несколько строк текста и символ поиска и определяет суммарное число символов в тексте.

4

#include <iostream>

#include <string>

using namespace std;

void task1(); // Прототип функции для задачи 1

void task2(); // Прототип функции для задачи 2

int main()

{

setlocale(LC\_CTYPE, "Russian");

int choice;

do

{

cout << "Выберите выполняемую задачу" << endl;

cout << "1 - В каждой строке массива выбрать минимальный элемент и среди них найти максимальный" << endl;

cout << "2 - Подсчет вхождений символа в текст" << endl;

cout << "3 - Выход" << endl;

cin >> choice;

switch (choice)

{

case 1:

task1();

break;

case 2:

task2();

break;

case 3:

break;

default:

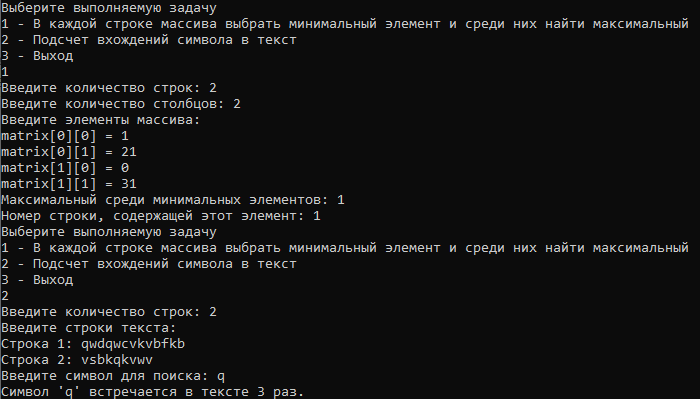
cout << "Некорректный выбор! Попробуйте снова." << endl;

}

} while (choice != 3);

return 0;

}



// Реализация задачи 2

void task2()

{

int numLines;

cout << "Введите количество строк: ";

cin >> numLines;

cin.ignore(); // Очистка буфера ввода

string\* lines = new string[numLines];

cout << "Введите строки текста:" << endl;

for (int i = 0; i < numLines; i++)

{

cout << "Строка " << i + 1 << ": ";

getline(cin, lines[i]);

}

char searchChar;

cout << "Введите символ для поиска: ";

cin >> searchChar;

int totalOccurrences = 0;

for (int i = 0; i < numLines; i++)

{

for (char ch : lines[i])

{

if (ch == searchChar)

totalOccurrences++;

}

}

cout << "Символ '" << searchChar << "' встречается в тексте " << totalOccurrences << " раз." << endl;

delete[] lines;

}

// Реализация задачи 1

void task1()

{

int rows, cols;

cout << "Введите количество строк: ";

cin >> rows;

cout << "Введите количество столбцов: ";

cin >> cols;

int\*\* matrix = new int\* [rows];

for (int i = 0; i < rows; i++)

matrix[i] = new int[cols];

cout << "Введите элементы массива:" << endl;

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

cout << "matrix[" << i << "][" << j << "] = ";

cin >> matrix[i][j];

}

}

int maxOfMin = INT\_MIN;

int rowIndex = -1;

for (int i = 0; i < rows; i++)

{

int minInRow = matrix[i][0];

for (int j = 1; j < cols; j++)

{

if (matrix[i][j] < minInRow)

minInRow = matrix[i][j];

}

if (minInRow > maxOfMin)

{

maxOfMin = minInRow;

rowIndex = i;

}

}

cout << "Максимальный среди минимальных элементов: " << maxOfMin << endl;

cout << "Номер строки, содержащей этот элемент: " << rowIndex + 1 << endl;

for (int i = 0; i < rows; i++)

delete[] matrix[i];

delete[] matrix;

}

**Дополнительные задания**

1. Дан двумерный массив, состоящий из **N** строк и **М** столбцов. Найти количество отрицательных, положительных и нулевых элементов массива.    
2. Написать программу, которая вводит несколько слов текста и определяет общее количество слов. Слова разделены пробелами.

8

#include <iostream>

#include <string>

using namespace std;

void countArrayElements();

void countWords();

int main(void)

{

setlocale(LC\_CTYPE, "Russian");

int choice;

do

{

cout << "Выберите выполняемую задачу" << endl;

cout << "1 - Подсчет отрицательных, положительных и нулевых элементов в массиве" << endl;

cout << "2 - Подсчет количества слов в тексте" << endl;

cout << "3 - Выход" << endl;

cin >> choice;

switch (choice)

{

case 1: countArrayElements();

break;

case 2: countWords();

break;

case 3: break;

default:

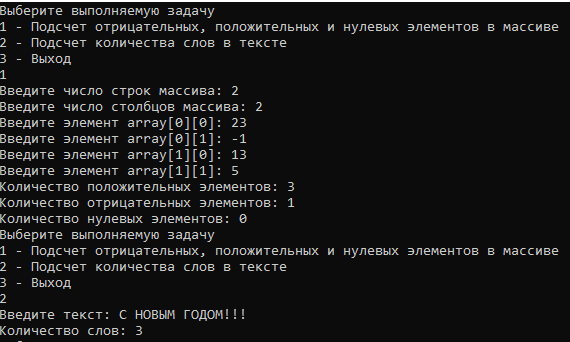
cout << "Некорректный выбор. Попробуйте снова." << endl;

}

} while (choice != 3);

return 0;

}



void countWords()

{

cin.ignore(); // Очистка буфера

string text;

cout << "Введите текст: ";

getline(cin, text);

int wordCount = 0;

bool inWord = false;

for (char c : text)

{

if (!isspace(c))

{

if (!inWord)

{

inWord = true;

wordCount++;

}

}

else

{

inWord = false;

}

}

cout << "Количество слов: " << wordCount << endl;

}

void countArrayElements()

{

int N, M, positive = 0, negative = 0, zero = 0;

cout << "Введите число строк массива: ";

cin >> N;

cout << "Введите число столбцов массива: ";

cin >> M;

int\*\* array = new int\* [N];

for (int i = 0; i < N; i++)

{

array[i] = new int[M];

for (int j = 0; j < M; j++)

{

cout << "Введите элемент array[" << i << "][" << j << "]: ";

cin >> array[i][j];

if (array[i][j] > 0)

positive++;

else if (array[i][j] < 0)

negative++;

else

zero++;

}

}

cout << "Количество положительных элементов: " << positive << endl;

cout << "Количество отрицательных элементов: " << negative << endl;

cout << "Количество нулевых элементов: " << zero << endl;

for (int i = 0; i < N; i++)

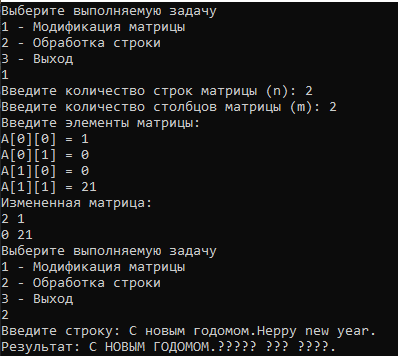
delete[] array[i];

delete[] array;

}

1. Ввести вещественные числа в массив **А**[**n**, **m**] (**n**, **m** > 0) и ко всем числам каждой строки прибавить единицу. Если первый элемент строки нулевой, ничего не делать.  
**2.** Дана строка. Все русские буквы привести к верхнему регистру, латинские − заменить символом «**?**». Вывести результат на экран.

2



#include <iostream>

#include <string>

#include <cctype>

#include <Windows.h>

using namespace std;

void modifyMatrix();

void processString();

int main(void)

{

setlocale(LC\_CTYPE, "Russian");

int choice;

do

{

cout << "Выберите выполняемую задачу" << endl;

cout << "1 - Модификация матрицы" << endl;

cout << "2 - Обработка строки" << endl;

cout << "3 - Выход" << endl;

cin >> choice;

switch (choice)

{

case 1: modifyMatrix();

break;

case 2: processString();

break;

case 3: break;

default:

cout << "Некорректный выбор. Попробуйте снова." << endl;

}

} while (choice != 3);

return 0;

}

void processString()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cin.ignore(); // Очистка буфера

string text;

cout << "Введите строку: ";

getline(cin, text);

for (char& c : text)

{

if (c >= 'а' && c <= 'я') // Проверка русских букв в нижнем регистре

{

c -= 32; // Приведение к верхнему регистру

}

else if (c >= 'А' && c <= 'Я')

{

// Уже в верхнем регистре, ничего не делаем

}

else if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z'))

{

c = '?'; // Замена латинских букв на '?'

}

}

cout << "Результат: " << text << endl;

}

void modifyMatrix()

{

int n, m;

cout << "Введите количество строк матрицы (n): ";

cin >> n;

cout << "Введите количество столбцов матрицы (m): ";

cin >> m;

if (n <= 0 || m <= 0)

{

cout << "Размеры матрицы должны быть больше нуля." << endl;

return;

}

double\*\* A = new double\* [n];

for (int i = 0; i < n; i++)

A[i] = new double[m];

cout << "Введите элементы матрицы:" << endl;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

cout << "A[" << i << "][" << j << "] = ";

cin >> A[i][j];

}

}

for (int i = 0; i < n; i++)

{

if (A[i][0] != 0)

{

for (int j = 0; j < m; j++)

A[i][j] += 1;

}

}

cout << "Измененная матрица:" << endl;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

cout << A[i][j] << " ";

}

cout << endl;

}

for (int i = 0; i < n; i++)

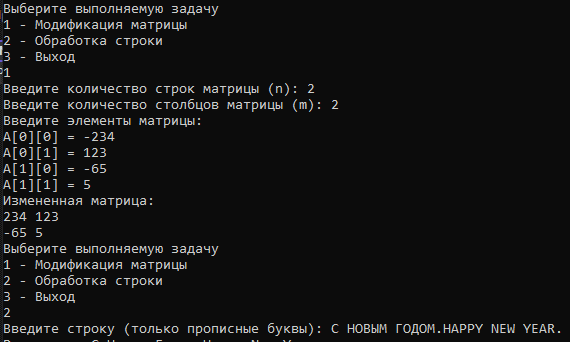
delete[] A[i];

delete[] A;

}

1. Ввести целые числа в массив **A**[**n**, **m**] (**n**, **m** > 0) и все отрицательные элементы в нечетных строках заменить на такие же положительные.  
2. Написать программу, преобразующую строку, состоящую только из прописных букв, в строку, состоящую из прописных и строчных букв, при этом первая буква после точки и пробела − прописная, остальные − строчные.

1



#include <iostream>

#include <string>

#include <windows.h>

using namespace std;

void modifyMatrix();

void processString();

int main(void)

{

setlocale(LC\_CTYPE, "Russian");

int choice;

do

{

cout << "Выберите выполняемую задачу" << endl;

cout << "1 - Модификация матрицы" << endl;

cout << "2 - Обработка строки" << endl;

cout << "3 - Выход" << endl;

cin >> choice;

switch (choice)

{

case 1: modifyMatrix();

break;

case 2: processString();

break;

case 3: break;

default:

cout << "Некорректный выбор. Попробуйте снова." << endl;

}

} while (choice != 3);

return 0;

}

void processString()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cin.ignore(); // Очистка буфера

string text;

cout << "Введите строку (только прописные буквы): ";

getline(cin, text);

bool capitalizeNext = true;

for (char& c : text)

{

if (capitalizeNext && (c >= 'А' && c <= 'Я' || c >= 'A' && c <= 'Z'))

{

// Оставляем букву прописной

capitalizeNext = false;

}

else if (c >= 'А' && c <= 'Я'|| c >= 'A' && c <= 'Z')

{

c = c + ('а' - 'А');

}

if (c == '.' || c == ' ')

{

capitalizeNext = true;

}

}

cout << "Результат: " << text << endl;

}

void modifyMatrix()

{

int n, m;

cout << "Введите количество строк матрицы (n): ";

cin >> n;

cout << "Введите количество столбцов матрицы (m): ";

cin >> m;

if (n <= 0 || m <= 0)

{

cout << "Размеры матрицы должны быть больше нуля." << endl;

return;

}

int\*\* A = new int\* [n];

for (int i = 0; i < n; i++)

A[i] = new int[m];

cout << "Введите элементы матрицы:" << endl;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

cout << "A[" << i << "][" << j << "] = ";

cin >> A[i][j];

}

}

for (int i = 0; i < n; i += 2)

{

for (int j = 0; j < m; j++)

{

if (A[i][j] < 0)

A[i][j] = -A[i][j];

}

}

cout << "Измененная матрица:" << endl;

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

cout << A[i][j] << " ";

}

cout << endl;

}

for (int i = 0; i < n; i++)

delete[] A[i];

delete[] A;

}

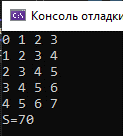
**Лабораторная работа № 17. Указатели и ссылки при работе с функциями**

1. Выполнив программу, записанную в правой части, изучить один из способов передачи *динамической* *матрицы* в функцию пользователя.

Записать условие задачи.

1

Условие задачи: программа динамически создает двумерный массив целых чисел, инициализирует его элементами, зависящими от индексов, и вычисляет сумму всех элементов в матрице.



#include <iostream>

using namespace std;

int sum(int\*\*, int, int);

int main()

{

int i, j;

int\*\* matr = new int\* [5];

for (i = 0; i < 5; i++)

matr[i] = new int[4];

for (i = 0; i < 5; i++)

{

for (j = 0; j < 4; j++)

{

matr[i][j] = i + j;

cout << matr[i][j] << ' ';

}

cout << endl;

}

cout << "S=" << sum(matr, 5, 4) << endl;

for (int i = 0; i < 5; i++)

delete matr[i];

delete[] matr;

}

int sum(int\*\* x, int n, int m)

{

int res = 0;

for (int i = 0; i < n; i++)

for (int j = 0; j < m; j++)

res += x[i][j];

return res;

}

2. Выполнив программу, записанную в правой части, изучить один из способов передачи *одномерного* *массива* в функцию, а также использование *указателя как результата* выполнения функции.

2

**#include <iostream>**

**int \*pfmin(int \*p, int n);**

**void main()**

**{ int masB[5] = { 4, 8, 2, 6, 4 };**

**(\*pfmin(masB, 5))++;**

**for (int i = 0; i < 5; i++)**

**std::cout<<masB[i]<<' ';**

**}**

**int \*pfmin(int \*p, int n)**

**{ int \*pmin;**

**for (pmin = p; n > 0; p++, n--)**

**if (\*p < \*pmin)**

**pmin = p;**

**return pmin;**

**}**



3. Изучить использование *ссылки* как *результата* работы функции, выполнив программу, записанную в правой части. Опробовать работу программы с разными значениями массива **А**.

3

#include <iostream>

using namespace std;

double& dmin(double A[], int size);

void main()

{

double s;

const int size = 5;

double A[] = { 5, 4.1, 3, 0.2, 11 };

s = dmin(A, size);

cout << s << endl;

for (int i = 0; i < size; i++)

cout << " " << A[i];

cout << endl;

dmin(A, size) = 1.0; // изменение минимума на значение 1.0

for (int i = 0; i < size; i++)

cout << " " << A[i];

}

double& dmin(double A[], int size)

{

int i, j = 0;

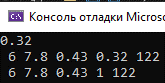
for (i = 1; i < size; i++)

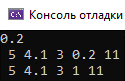
if (A[j] > A[i])

j = i;

return A[j];

}





4. Ознакомиться с использованием функций с результатом *логического* типа, опробовав работу программы в правой части

4





#include <iostream>

using namespace std;

bool is\_elem(int\* pA, int n, int iV);

void main()

{

setlocale(LC\_CTYPE, "Rus");

bool t; int k;

int A[] = { 5, 4, 3, 2, 11 };

cout << "Введите число ";

cin >> k;

t = is\_elem(A, sizeof(A) / sizeof(int), k);

if (t == true)

cout << "Число есть в массиве ";

else

cout << "Числа нет в массиве ";

}

bool is\_elem(int\* pA, int n, int iV)

{

bool bf = false;

for (int i = 0; i < n; i++)

if (pA[i] == iV)

{

bf = true;

break;

}

return bf;

}

1. В одномерном массиве, содержащем 15 элементов, посчитать и вывести сумму всех четных элементов массива.  
2. Даны две целочисленные матрицы размером **n** x **m** и **m** x **n**. Если в матрице отсутствуют нулевые элементы, то определить произведение этих матриц.

5

#include <iostream>

using namespace std;

int\* sumEvenElements(int\* arr, int size, int& sum);

int\*\* multiplyMatrices(int\*\* A, int n, int m, int\*\* B, int p, int q);

int main(void)

{

setlocale(LC\_CTYPE, "Russian");

int choice;

do

{

cout << "Выберите выполняемую задачу" << endl;

cout << "1 - Сумма четных элементов массива" << endl;

cout << "2 - Умножение матриц" << endl;

cout << "3 - Выход" << endl;

cin >> choice;

switch (choice)

{

case 1: {

int arr[15], sum = 0;

cout << "Введите 15 элементов массива:" << endl;

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

cin >> arr[i];

int\* result = sumEvenElements(arr, 15, sum);

cout << "Сумма четных элементов массива: " << \*result << endl;

break;

}

case 2: {

int n, m, p, q;

cout << "Введите размеры первой матрицы (n x m): ";

cin >> n >> m;

cout << "Введите размеры второй матрицы (p x q): ";

cin >> p >> q;

if (m != p) {

cout << "Умножение невозможно, число столбцов первой матрицы должно быть равно числу строк второй матрицы." << endl;

break;

}

int\*\* A = new int\* [n];

int\*\* B = new int\* [p];

cout << "Введите элементы первой матрицы:" << endl;

for (int i = 0; i < n; i++) {

A[i] = new int[m];

for (int j = 0; j < m; j++) {

cin >> A[i][j];

}

}

cout << "Введите элементы второй матрицы:" << endl;

for (int i = 0; i < p; i++) {

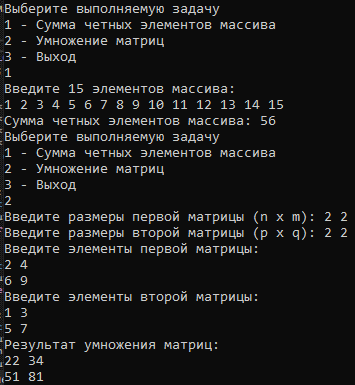
B[i] = new int[q];

for (int j = 0; j < q; j++) {

cin >> B[i][j];

}

}



else {

cout << "В одной из матриц присутствуют нулевые элементы." << endl;

}

for (int i = 0; i < n; i++)

delete[] A[i];

delete[] A;

for (int i = 0; i < p; i++)

delete[] B[i];

delete[] B;

break;

}

case 3:

break;

default:

cout << "Некорректный выбор. Попробуйте снова." << endl;

}

} while (choice != 3);

return 0;

}

int\* sumEvenElements(int\* arr, int size, int& sum) {

sum = 0;

for (int i = 0; i < size; i++) {

if (arr[i] % 2 == 0) {

sum += arr[i];

}

}

return &sum;

}

int\*\* multiplyMatrices(int\*\* A, int n, int m, int\*\* B, int p, int q) {

int\*\* result = new int\* [n];

for (int i = 0; i < n; i++) {

result[i] = new int[q];

for (int j = 0; j < q; j++) {

result[i][j] = 0;

for (int k = 0; k < m; k++) {

result[i][j] += A[i][k] \* B[k][j];

}

}

}

return result;

}

int\*\* A = new int\* [n];

int\*\* B = new int\* [p];

cout << "Введите элементы первой матрицы:" << endl;

for (int i = 0; i < n; i++) {

A[i] = new int[m];

for (int j = 0; j < m; j++) {

cin >> A[i][j];

}

}

cout << "Введите элементы второй матрицы:" << endl;

for (int i = 0; i < p; i++) {

B[i] = new int[q];

for (int j = 0; j < q; j++) {

cin >> B[i][j];

}

}

bool isValid = true;

for (int i = 0; i < n && isValid; i++) {

for (int j = 0; j < m; j++) {

if (A[i][j] == 0) {

isValid = false;

break;

}

}

}

for (int i = 0; i < p && isValid; i++) {

for (int j = 0; j < q; j++) {

if (B[i][j] == 0) {

isValid = false;

break;

}

}

}

if (isValid) {

int\*\* result = multiplyMatrices(A, n, m, B, p, q);

if (result) {

cout << "Результат умножения матриц:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < q; j++) {

cout << result[i][j] << " ";

}

cout << endl;

}

for (int i = 0; i < n; i++)

delete[] result[i];

delete[] result;

}

}

else {

cout << "В одной из матриц присутствуют нулевые элементы." << endl;

}

Дополнительные задания

**1.**

1. Определить сумму элементов целочисленного массива, расположенных между первым максимальным и последним ми­нимальным элементами.

2. Даны две квадратные целочисленные матрицы. Если все числа положительны, то определить произведение этих матриц.

#include <iostream>

using namespace std;

int sumBetweenMaxAndMin(int\* arr, int size);

int\*\* multiplyMatrices(int\*\* A, int n, int\*\* B);

int main(void) {

setlocale(LC\_CTYPE, "Russian");

int choice;

do {

cout << "Выберите выполняемую задачу" << endl;

cout << "1 - Сумма элементов между первым максимальным и последним минимальным" << endl;

cout << "2 - Умножение двух матриц" << endl;

cout << "3 - Выход" << endl;

cin >> choice;

switch (choice) {

case 1: {

int size;

cout << "Введите размер массива: ";

cin >> size;

if (size <= 0) {

cout << "Размер массива должен быть положительным." << endl;

break;

}

int\* arr = new int[size];

cout << "Введите элементы массива:" << endl;

for (int i = 0; i < size; i++) {

cin >> arr[i];

}

int result = sumBetweenMaxAndMin(arr, size);

if (result == -1) {

cout << "Невозможно вычислить сумму: первый максимальный элемент не перед последним минимальным." << endl;

}

else {

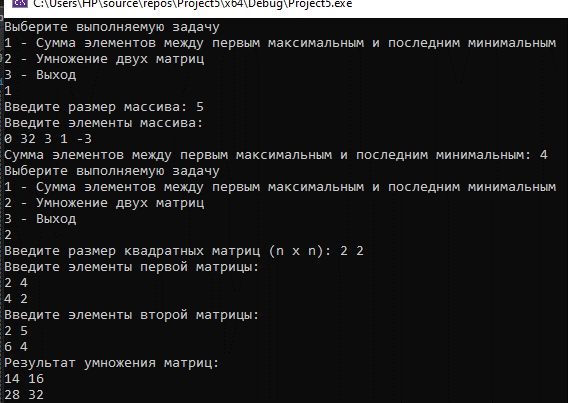
cout << "Сумма элементов между первым максимальным и последним минимальным: " << result << endl;

}

delete[] arr;

break;

}



for (int i = 0; i < n; i++) {

delete[] A[i];

delete[] B[i];

}

delete[] A;

delete[] B;

break;

}

case 3:

break;

}

} while (choice != 3);

return 0;

}

int sumBetweenMaxAndMin(int\* arr, int size) {

int firstMaxIndex = -1, lastMinIndex = -1;

int max = arr[0], min = arr[0];

for (int i = 0; i < size; i++) {

if (arr[i] > max) {

max = arr[i];

firstMaxIndex = i;

}

if (arr[i] <= min) {

min = arr[i];

lastMinIndex = i;

}

}

if (firstMaxIndex == -1 || lastMinIndex == -1 || firstMaxIndex >= lastMinIndex) {

return -1; // Сумма невозможна

}

int sum = 0;

for (int i = firstMaxIndex + 1; i < lastMinIndex; i++) {

sum += arr[i];

}

return sum;

}

int\*\* multiplyMatrices(int\*\* A, int n, int\*\* B) {

int\*\* result = new int\* [n];

for (int i = 0; i < n; i++) {

result[i] = new int[n];

for (int j = 0; j < n; j++) {

result[i][j] = 0;

for (int k = 0; k < n; k++) {

result[i][j] += A[i][k] \* B[k][j];

}

}

}

return result;

}

case 2: {

int n;

cout << "Введите размер квадратных матриц (n x n): ";

cin >> n;

if (n <= 0) {

cout << "Размер матриц должен быть положительным." << endl;

break;

}

int\*\* A = new int\* [n];

int\*\* B = new int\* [n];

cout << "Введите элементы первой матрицы:" << endl;

for (int i = 0; i < n; i++) {

A[i] = new int[n];

for (int j = 0; j < n; j++) {

cin >> A[i][j];

}

}

cout << "Введите элементы второй матрицы:" << endl;

for (int i = 0; i < n; i++) {

B[i] = new int[n];

for (int j = 0; j < n; j++) {

cin >> B[i][j];

}

}

bool isValid = true;

for (int i = 0; i < n && isValid; i++) {

for (int j = 0; j < n; j++) {

if (A[i][j] <= 0 || B[i][j] <= 0) {

isValid = false;

break;

}

}

}  
 if (isValid) {

int\*\* result = multiplyMatrices(A, n, B);

cout << "Результат умножения матриц:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

cout << result[i][j] << " ";

}

cout << endl;

}

for (int i = 0; i < n; i++) {

delete[] result[i];

}

delete[] result;

}

else {

cout << "В одной из матриц есть неположительные элементы." << endl;

}

for (int i = 0; i < n; i++) {

delete[] A[i];

delete[] B[i];

}

delete[] A;

delete[] B;

break;

}

case 3:

break;

}

} while (choice != 3);

return 0;

}

1. В массиве из целых чисел определить сумму элементов, расположенных между первым и последним нулевыми элементами

2. Если есть в матрице строка, все элементы которой положительны, то найти сумму этих элементов. Уменьшить все элементы матрицы на эту сумму.

3

#include <iostream>

using namespace std;

int sumBetweenZeros(int\* arr, int size);

int modifyMatrixIfPositiveRowExists(int\*\* matrix, int n, int m);

int main(void) {

setlocale(LC\_CTYPE, "Russian");

int choice;

do {

cout << "Выберите выполняемую задачу" << endl;

cout << "1 - Сумма элементов между первым и последним нулевыми элементами" << endl;

cout << "2 - Работа с матрицей (сумма строки и уменьшение элементов)" << endl;

cout << "3 - Выход" << endl;

cin >> choice;

switch (choice) {

case 1: {

int size;

cout << "Введите размер массива: ";

cin >> size;

if (size <= 0) {

cout << "Размер массива должен быть положительным." << endl;

break;

}

int\* arr = new int[size];

cout << "Введите элементы массива:" << endl;

for (int i = 0; i < size; i++) {

cin >> arr[i];

}

int result = sumBetweenZeros(arr, size);

if (result == -1) {

cout << "Сумма невозможна: отсутствуют два нулевых элемента." << endl;

}

else {

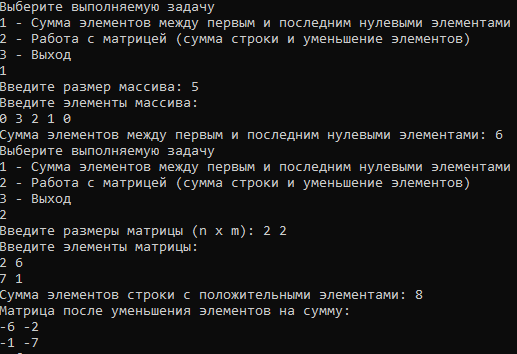
cout << "Сумма элементов между первым и последним нулевыми элементами: " << result << endl;

}

delete[] arr;

break;

}



if (firstZeroIndex == -1 || lastZeroIndex == -1 || firstZeroIndex == lastZeroIndex) {

return -1; // Сумма невозможна

}

int sum = 0;

for (int i = firstZeroIndex + 1; i < lastZeroIndex; i++) {

sum += arr[i];

}

return sum;

}

int modifyMatrixIfPositiveRowExists(int\*\* matrix, int n, int m) {

int sum = 0;

bool positiveRowFound = false;

for (int i = 0; i < n; i++) {

bool allPositive = true;

int rowSum = 0;

for (int j = 0; j < m; j++) {

if (matrix[i][j] <= 0) {

allPositive = false;

break;

}

rowSum += matrix[i][j];

}

if (allPositive) {

sum = rowSum;

positiveRowFound = true;

break;

}

}

if (positiveRowFound) {

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

matrix[i][j] -= sum;

}

}

}

return positiveRowFound ? sum : 0;

}

case 2: {

int n, m;

cout << "Введите размеры матрицы (n x m): ";

cin >> n >> m;

if (n <= 0 || m <= 0) {

cout << "Размеры матрицы должны быть положительными." << endl;

break;

}

int\*\* matrix = new int\* [n];

cout << "Введите элементы матрицы:" << endl;

for (int i = 0; i < n; i++) {

matrix[i] = new int[m];

for (int j = 0; j < m; j++) {

cin >> matrix[i][j];

}

}

int sum = modifyMatrixIfPositiveRowExists(matrix, n, m);

if (sum > 0) {

cout << "Сумма элементов строки с положительными элементами: " << sum << endl;

cout << "Матрица после уменьшения элементов на сумму:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

else {

cout << "В матрице нет строки, все элементы которой положительны." << endl;

}

for (int i = 0; i < n; i++) {

delete[] matrix[i];

}

delete[] matrix;

break;

}

case 3:

break;

default:

cout << "Некорректный выбор. Попробуйте снова." << endl;

}

} while (choice != 3);

return 0;

}

int sumBetweenZeros(int\* arr, int size) {

int firstZeroIndex = -1, lastZeroIndex = -1;

for (int i = 0; i < size; i++) {

if (arr[i] == 0) {

if (firstZeroIndex == -1) {

firstZeroIndex = i;

}

lastZeroIndex = i;

}

}

if (firstZeroIndex == -1 || lastZeroIndex == -1 || firstZeroIndex == lastZeroIndex) {

return -1; // Сумма невозможна

}

1. Найти количество отрицательных элементов, стоящих на чётных местах в одномерном массиве.

2. Проверить, есть ли в матрице хотя бы одна строка, содержащая отрицательный элемент, и найти ее номер. Все элементы столбца с таким же номером уменьшить вдвое.

12

#include <iostream>

using namespace std;

int countNegativeOnEvenIndices(int\* arr, int size);

int modifyMatrixIfNegativeRowExists(int\*\* matrix, int n, int m);

int main(void) {

setlocale(LC\_CTYPE, "Russian");

int choice;

do {

cout << "Выберите выполняемую задачу" << endl;

cout << "1 - Количество отрицательных элементов на чётных местах" << endl;

cout << "2 - Работа с матрицей (проверка строки и уменьшение столбца)" << endl;

cout << "3 - Выход" << endl;

cin >> choice;

switch (choice) {

case 1: {

int size;

cout << "Введите размер массива: ";

cin >> size;

if (size <= 0) {

cout << "Размер массива должен быть положительным." << endl;

break;

}

int\* arr = new int[size];

cout << "Введите элементы массива:" << endl;

for (int i = 0; i < size; i++) {

cin >> arr[i];

}

int result = countNegativeOnEvenIndices(arr, size);

cout << "Количество отрицательных элементов на чётных местах: " << result << endl;

delete[] arr;

break;

}

case 2: {

int n, m;

cout << "Введите размеры матрицы (n x m): ";

cin >> n >> m;

if (n <= 0 || m <= 0) {

cout << "Размеры матрицы должны быть положительными." << endl;

break;

}

int\*\* matrix = new int\* [n];

cout << "Введите элементы матрицы:" << endl;

for (int i = 0; i < n; i++) {

matrix[i] = new int[m];

for (int j = 0; j < m; j++) {

cin >> matrix[i][j];

}

}

int result = modifyMatrixIfNegativeRowExists(matrix, n, m);

if (result != -1) {

cout << "Номер строки с отрицательным элементом: " << result + 1 << endl;

cout << "Матрица после уменьшения элементов столбца вдвое:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

else {

cout << "В матрице нет строк с отрицательными элементами." << endl;

}

for (int i = 0; i < n; i++) {

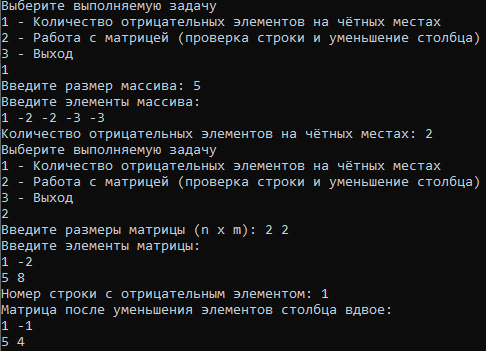
delete[] matrix[i];

}

delete[] matrix;

break;

}



case 3:

break;

default:

cout << "Некорректный выбор. Попробуйте снова." << endl;

}

} while (choice != 3);

return 0;

}

int countNegativeOnEvenIndices(int\* arr, int size) {

int count = 0;

for (int i = 1; i < size; i += 2) {

if (arr[i] < 0) {

count++;

}

}

return count;

}

int modifyMatrixIfNegativeRowExists(int\*\* matrix, int n, int m) {

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

if (matrix[i][j] < 0) {

for (int k = 0; k < n; k++) {

matrix[k][j] /= 2;

}

return i; // Номер строки с отрицательным элементом

}

}

}

return -1; // Отрицательных элементов нет

}

int\*\* matrix = new int\* [n];

cout << "Введите элементы матрицы:" << endl;

for (int i = 0; i < n; i++) {

matrix[i] = new int[m];

for (int j = 0; j < m; j++) {

cin >> matrix[i][j];

}

}

int result = modifyMatrixIfNegativeRowExists(matrix, n, m);

if (result != -1) {

cout << "Номер строки с отрицательным элементом: " << result + 1 << endl;

cout << "Матрица после уменьшения элементов столбца вдвое:" << endl;

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

else {

cout << "В матрице нет строк с отрицательными элементами." << endl;

}

for (int i = 0; i < n; i++) {

delete[] matrix[i];

}

delete[] matrix;

break;

}