[Блок 1](#_Лабораторная_работа_1.)

[Блок 2](#_Лабораторная_работа_№_1)

[Блок 3](#_Лабораторная_работа_№_2)

[Блок 4](#_Лабораторная_работа_№_3)

[Блок 5](#_Лабораторная_работа_№_4)

[Блок 6](#_Лабораторная_работа_№_6)

## 

## Лабораторная работа 1. Системы счисления

Выполнить перевод чисел

1. 11(8 c/c)→ 001001(2 c/c) → 9(16 c/c);
2. 11111001(2 c/c)→ 249(10c/c);
3. 77(10 c/c)→ 1001101(2 c/c)
4. 342(8 c/c)→ 011 100 010(2 c/c) → E2(16 c/c);
5. 11011(2 c/c)→ 33(8c/c) →1B(16c/c);
6. 1011(10 c/c)→1111110011(2 c/c);

**Лабораторная работа 2 Способы представления алгоритмов**

Даны действительные числа **x, y, z**. Вычислить **min(x + y + z, xyz) \* min(x, y, z).**

Начало

­

да

min=m

min=k

min>m

k=x + y + z, m=xyz

Ввод x, y, z

нет

­­

Min1=x

да

Min1>y

Min1=y

нет

да

Min1>z

Min1=z

нет

t=min\*Min1

Вывод t

Конец

1. Ввести числовые значения х, у, z.

2. Вычислить k, m по формулам :

k= x +y +z

m= xyz

3. Присвоить min значение переменной k.

4. Если min>m, то присвоим переменной min значение m. Иначе перейти к п. 5.

5. Присвоим переменной Min1 значение х.

6. Если условие Min1>y выполняется, то перейти к Min1=y. Иначе перейти к Min1>z. Если условие выполняется, то перейти к Min1=z.

7. Присвоим переменной t значение min\*Min1.

8. Напечатать значение t.

9. Конец вычислений.

Номер 2:

Определить есть ли среди заданных целых чисел **A**, **B**, **C**, **D** хотя бы одно чётное.

Начало

Ввод **A,B,C,D**

t = 0

нет

**A**%2 =0

да

t=t+1

t = t + 1

**B**%2=0

нет

да

**C**% 2 = 0

нет

да

t = t + 1

**D**%2=0

да

t = t + 1

нет

t ≥ 1

Вывод «**нет**»

да

Вывод «**да**»

Конец

1. Ввод A,B,C,D.
2. t=0, перейти к п. 4.
3. Если А%2=0 то перейти к t=t+1.Иначе к B%2=0.
4. Если B%2=0 то перейти к t=t+1.Иначе к C%2=0.
5. Если С%2=0 то перейти к t=t+1.Иначе к D%2=0.
6. Если D%2=0 то перейти к t=t+1.Иначе к t>=1.
7. Если t>=1 нет - вывод ‘нет’, если t>=1 да – вывод ‘да’.
8. Конец.

**Лабораторная работа 3. Основные элементы языка C++**

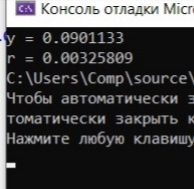
|  |  |  |
| --- | --- | --- |
| 12 |  |  |

#include <iostream>  
#include <cmath>  
using namespace std;  
int main()  
{  
double t, u, k = 6, r = 5e-7, x = 95e-3;  
t = tan(x) + r \* (1 - log(x));  
u = t / (pow(x, 3) + 1) + (1 - exp(k - 4));  
cout << "t = " << t << endl;  
cout << "u = " << u;  
}

t = 0.0952885

u = -6.29385

|  |  |  |
| --- | --- | --- |
| 15 |  |  |

#include <iostream>  
#include <cmath>  
using namespace std;  
int main()  
{  
double y, r, a = 175e-2, b = 45e-5;  
y = a\* exp(-2 \* b) - sqrt(1+a) ;  
r = log(1 + 20 \* b) / ( 1 + a);  
cout << "y = " << y;  
cout << "r = " << r;  
}

## Лабораторная работа № 4. Ввод и вывод информации

1. Нарисовать знак +, закрашенный введенным символом.

***потоковый*** способ

#include<iomanip>

#include<iostream>

Using namespace std;

Int main()

{

Setlocale(LC\_CTYPE, «Russian»);

Char c, probe setfill© << c << endl;

Cout << setw(16) << setfill(probel) << probel;

Cout << setw(2) << setfill© << c << endl;

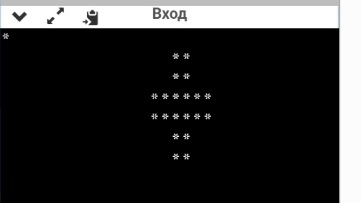
Cout << setw(14) << setfill(probel) << probel;

Cout << setw(6) << setfill© << c << endl;

Cout << setw(14) << setfill(probel) << probel;

Cout << setw(16) << setfill(probel) << probel;

Cout << setw(2) << setfill(probel) << probel;

Cout << setw(6) << setfill© << c << endl;

Cout << setw(16) << setfill(probel) << probel;

Cout << setw(2) << setfill© << c << endl;

Cout << setw(16) << setfill(probel) << probel;

Cout << setw(2) << setfill© << c << endl;

}

**Форматированный** способ

#include<iostream>

#include<stdio.h>

void main ()

{

Char s, probel;

probel = ' ';

Setlocale(LC\_CTYPE, «Russian»);

printf ("Введите символ\n");

scanf\_s("%c",&s);

printf("%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c", probel, probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel);

printf("%c%c\n",s, s);

printf("%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c", probel, probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel);

printf("%c%c\n",s, s);

printf("%c%c%c%c%c%c%c%c%c%c%c%c%c%c", probel, probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel);

printf("%c%c%c%c\n",s, s,s ,s);

printf("%c%c%c%c%c%c%c%c%c%c%c%c%c%c", probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel);

printf("%c%c\n",s, s, s,s);

printf("%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c", probel, probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel);

printf("%c%c\n",s, s);

printf("%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c", probel, probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel,probel);

printf("%c%c\n",s, s);

}

2. Сформировать число, представляющее собой реверсную (обратную в порядке следования разрядов) запись заданного трехзначного числа.

#include <iostream>

using namespace std;

int main()

{

int num, num1, num2, num3;

cout « "Number is ";

cin » num;

num1 = num / 100;

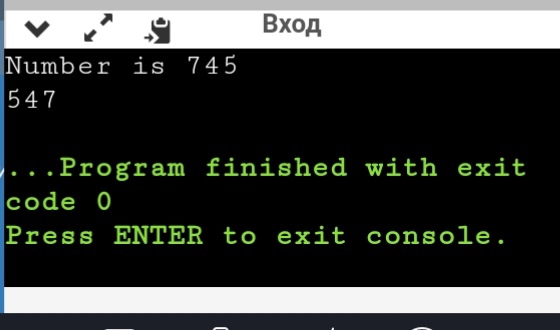
num2 = num % 100 / 10;

num3 = num % 10;

num = num3 \* 100 + num2 \* 10 + num1;

cout « num;

}



## Лабораторная работа № 5. Разветвляющиеся программы

Даны действительные числа **x, y, z**. Вычислить **min(x + y + z, xyz) \* min(x, y, z).**

#include <iostream>

void main()

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

float x, y, z, min, min1;

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

printf("Введите y "); scanf\_s("%f", &y);

printf("Введите z "); scanf\_s("%f", &z);

if ((x + y + z) < (x \* y \* z))

min = (x + y + z);

else

min = (x \* y \* z);

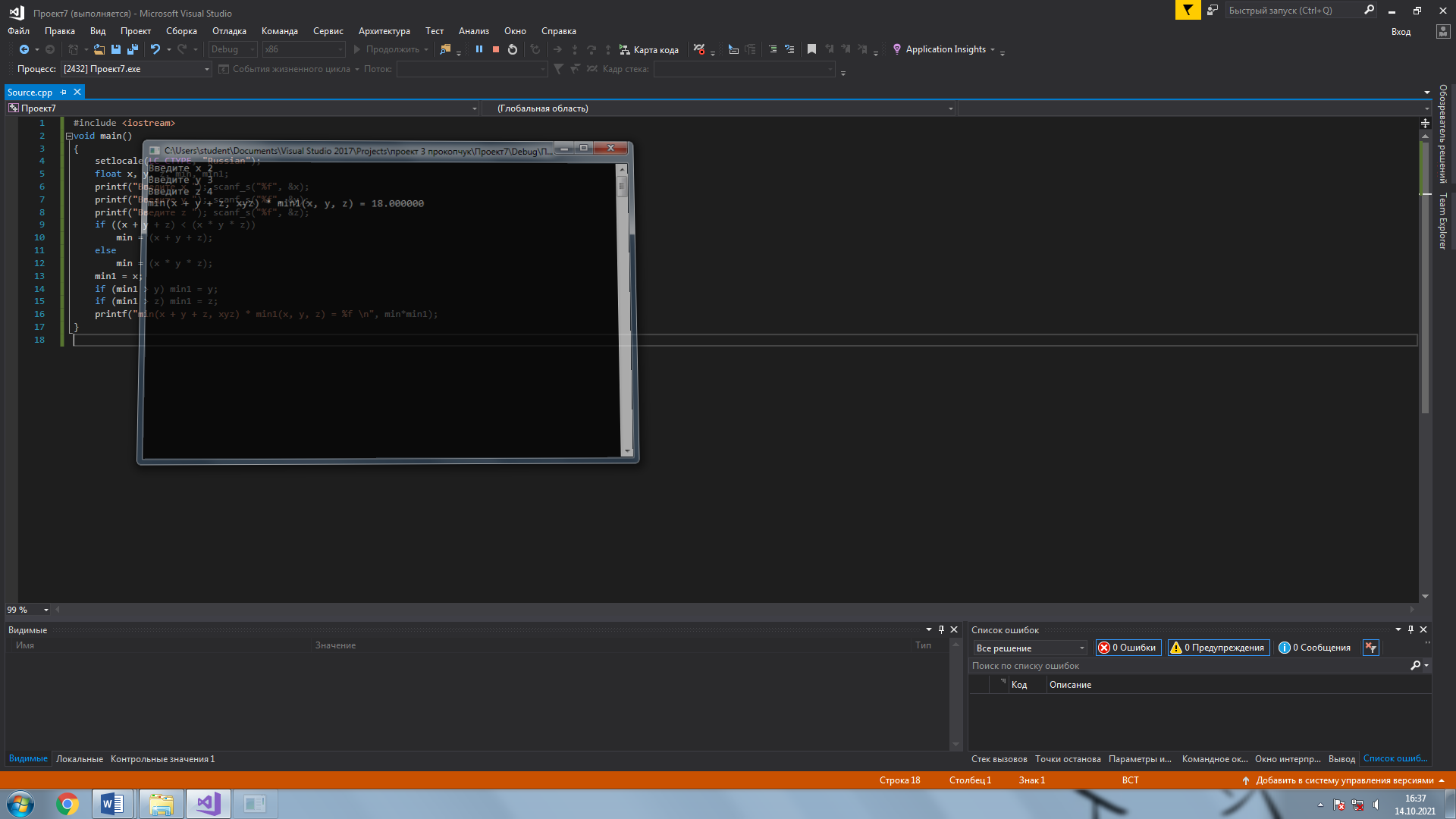
min1 = x;

if (min1 > y) min1 = y;

if (min1 > z) min1 = z;

printf("min(x + y + z, xyz) \* min1(x, y, z) = %f \n", min\*min1);

}



Определить есть ли среди заданных целых чисел **A**, **B**, **C**, **D** хотя бы одно чётное.

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "Russian");

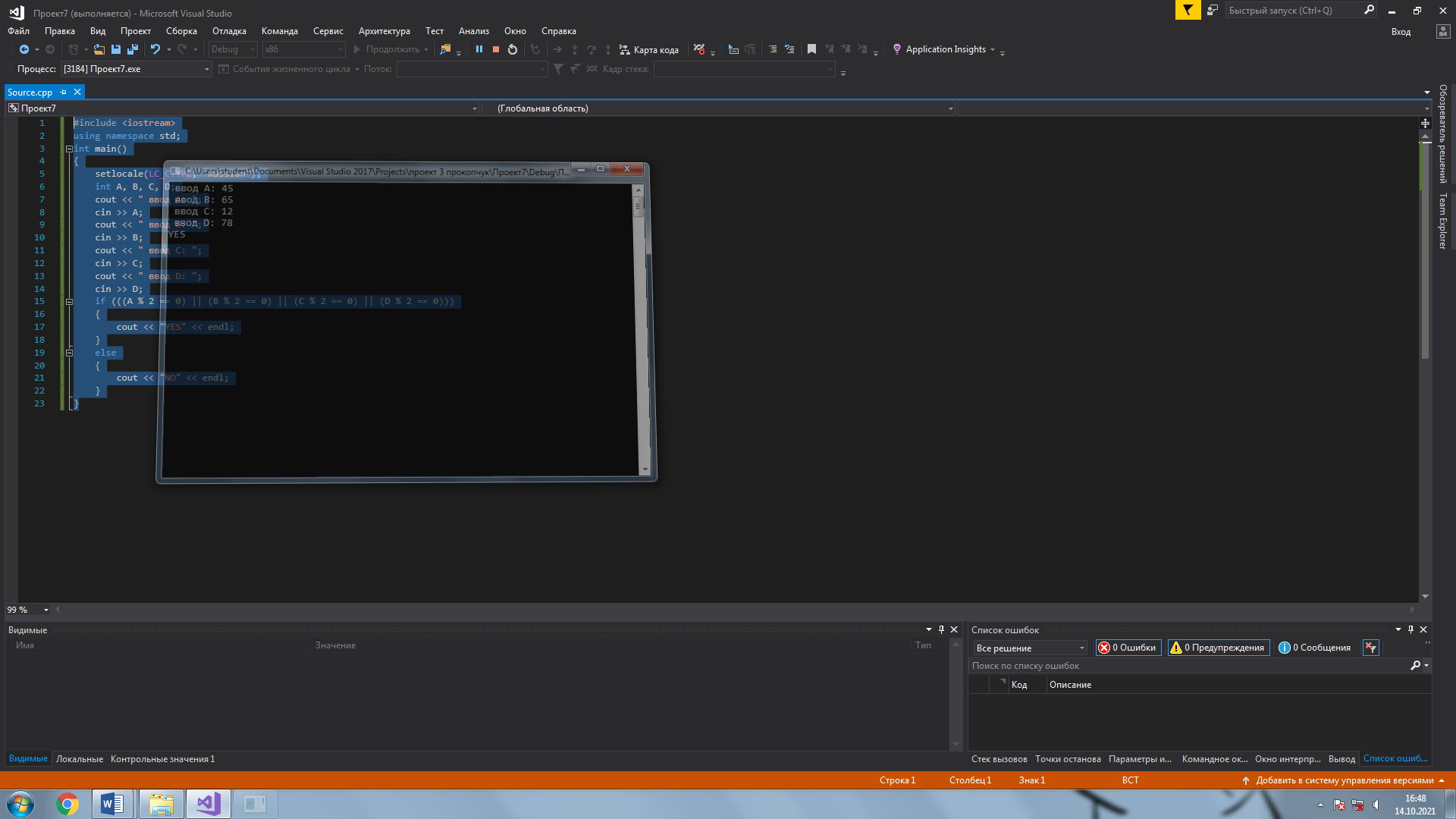
int A, B, C, D;

cout << " ввод A: ";

cin >> A;

cout << " ввод B: ";

cin >> B;

 cout << " ввод C: ";

cin >> C;

cout << " ввод D: ";

cin >> D;

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

{

cout << "YES" << endl;

}

else

{

cout << "NO" << endl;

}

}

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

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "Russian");

int q;

puts("Привет, ты сегодня занята? ( 1 - нет; 2 - да) ");

cin >> q;

switch (q)

{

case 1:

{

puts("Погуляем? ( 1 - да, давай; 2 - нет, у меня встреча с мамой) ");

cin >> q;

switch (q)

{

case 1: puts("тогда через час возле кафе"); break;

case 2: puts("напиши, когда освободишься"); break;

}

break;

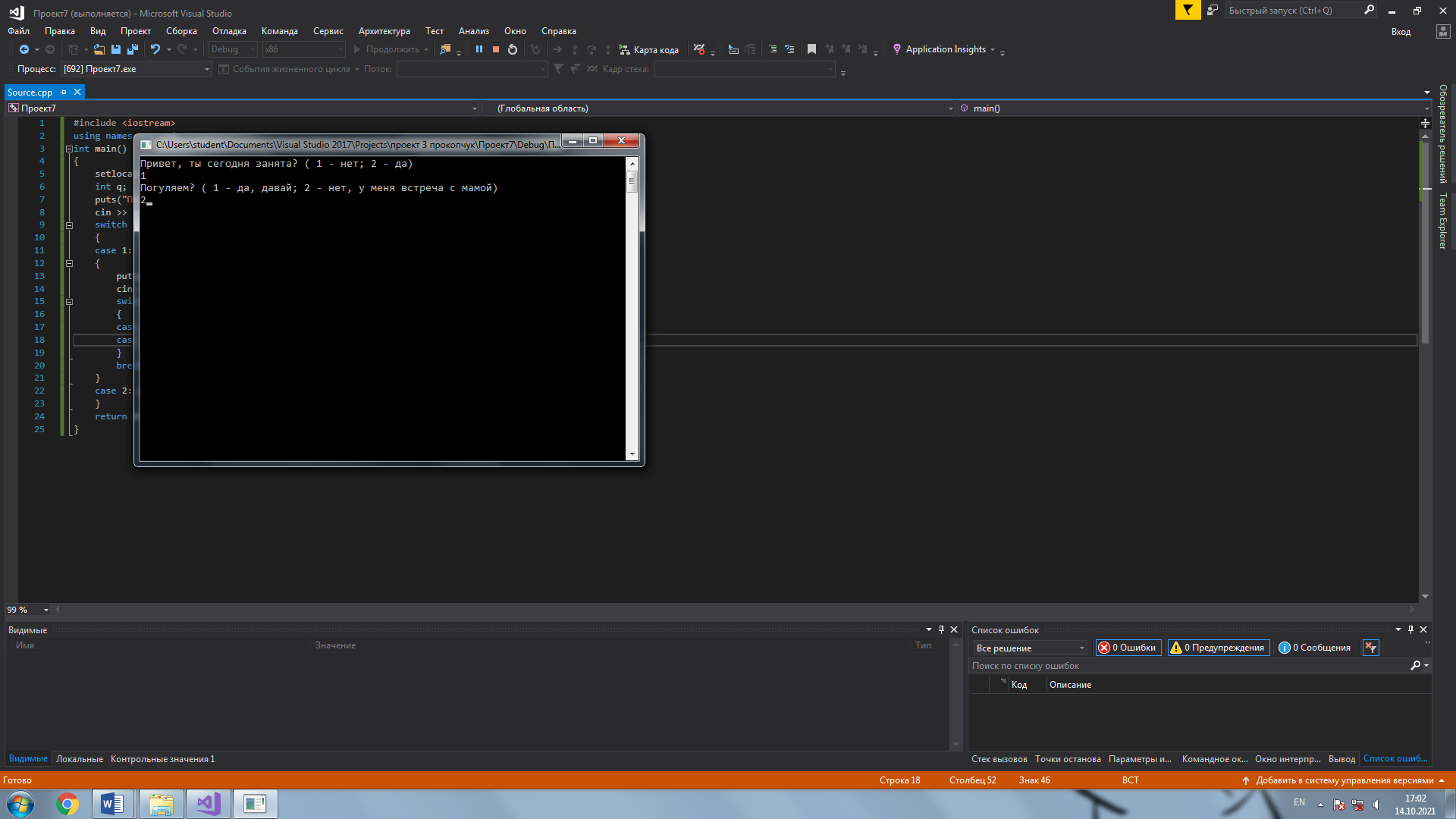
}

case 2: puts("До выходных тогда"); break;

}

return 0;

}



## Лабораторная работа № 6. Циклические программы

В таблице приведены формулы и три варианта исходных данных, по которым надо разработать три ***блок-схемы*** и три ***циклические*** программы с одними и теми же расчетными формулами.

|  |  |  |  |
| --- | --- | --- | --- |
| Формулы для вычислений | For | While | To while |
|  |  | j = 2(0,5)3 | j = 2(0,5)3,  m={0,4; −1; 1,9} |

1. FOR

Начало

a=24000, b=5e-1, m=8, j[3] = { 1.7, 2.9, 8 }

t=0,3



Конец

Вывод y, z

#include <iostream>

int main()

{

setlocale(LC\_CTYPE, "Russian");

float a=24000, b=5e-1, m=8, y, z, j[3] = { 1.7, 2.9, 8 };

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

{

printf("Введите j ");

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

y = ( m - b) /(sin(a)-exp(a));

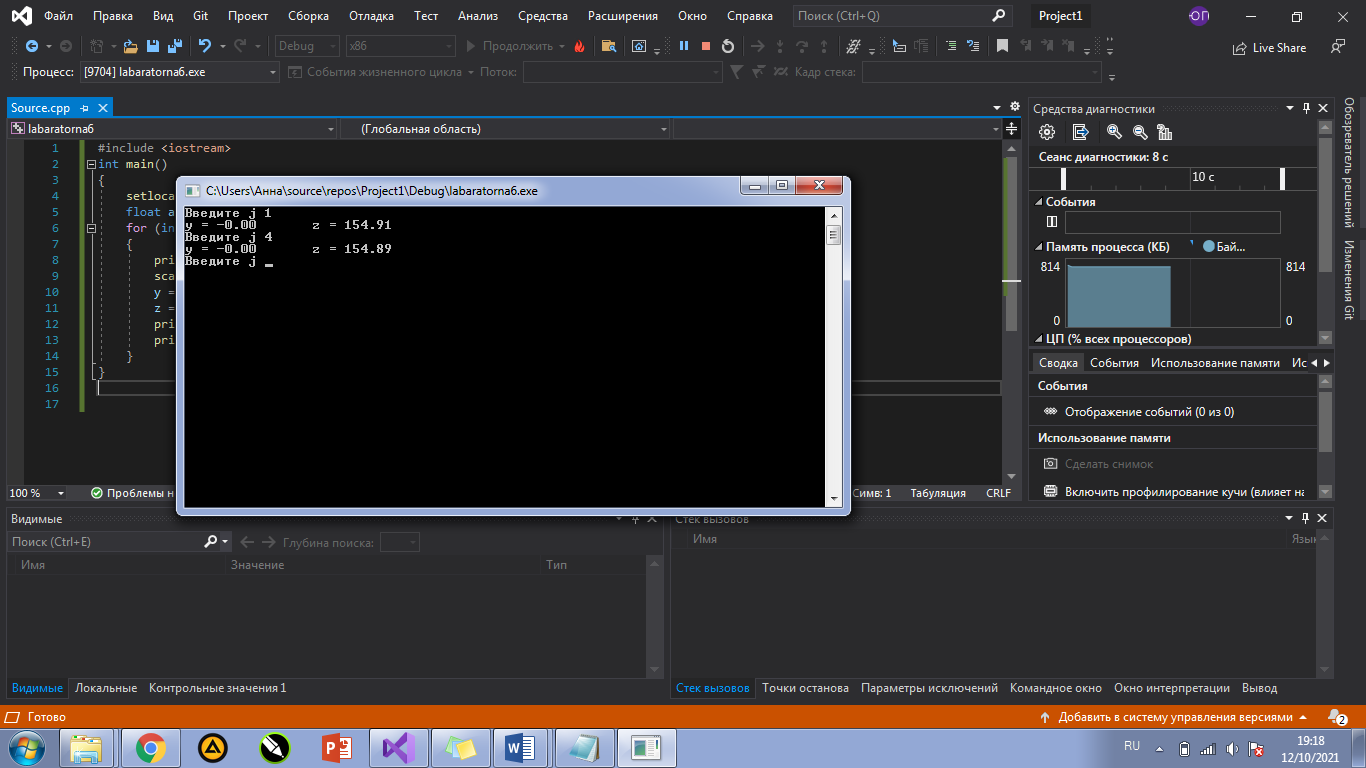
z =3\*y + sqrt( a-4\*j\*b);

printf("y = %5.2f\t", y);

printf("z = %5.2f\n", z);

}

}



1. WHILE

#include <iostream>

using namespace std;

int main()

{

float z, y, j = 2, a=24000, b=5e-1, m=8;

while (j <= 3)

{ y = ( m - b) /(sin(a)-exp(a));

z =3\*y + sqrt( a-4\*j\*b);

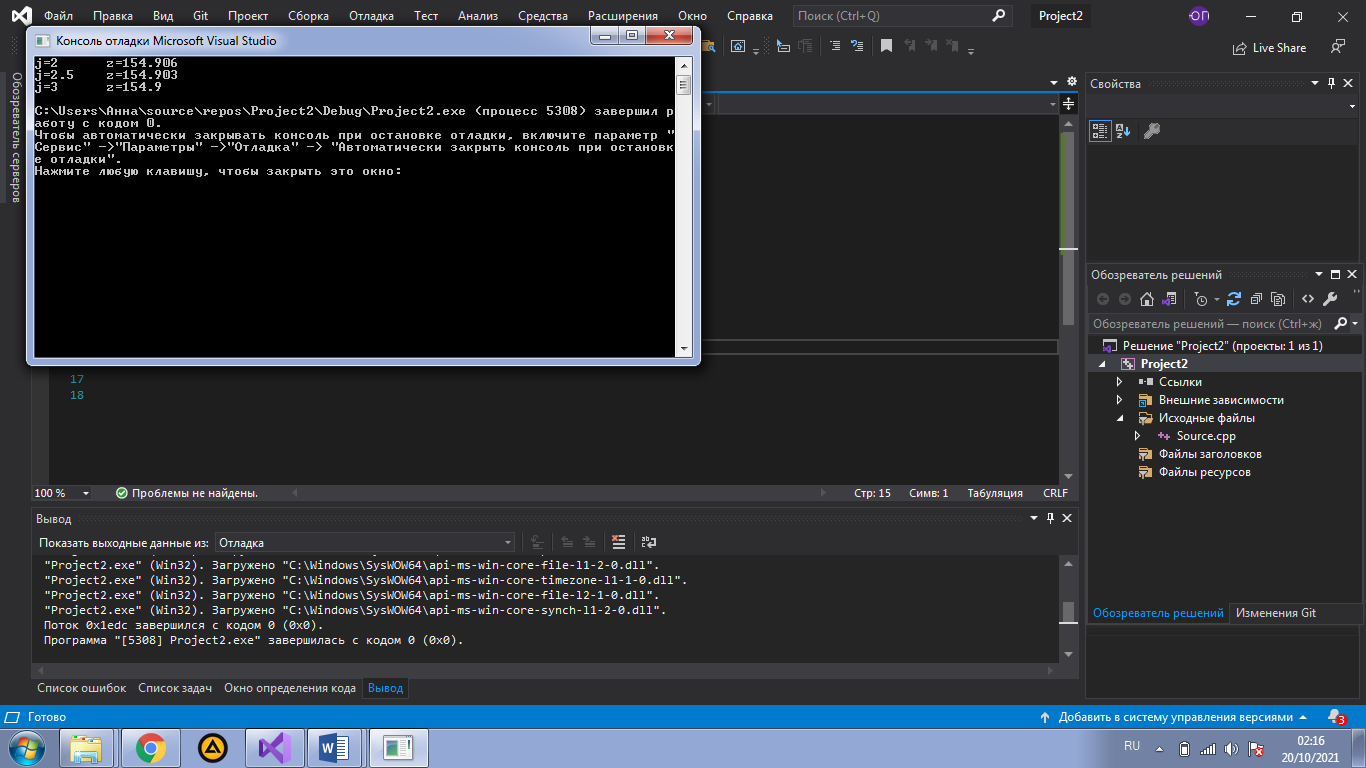
cout << "j=" << j << "\t";

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

j = j + 0.5;

}

}



Начало



Вывод y, z

Конец

j=j+0.5

j<=3

a=24000, b=5e-1, m=8, j=2

3) TO WHILE

#include <iostream>

using namespace std;

int main()

{

double z, y, j, a = 24000, b = 5, m[3] = { 0.4, -1, 1.9 };

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

{

j = 2;

while (j <= 3)

{

y = (m[i] - b) / (sin(a) - exp(a));

z = 3 \* y + sqrt(a - 4 \* j \* b);

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

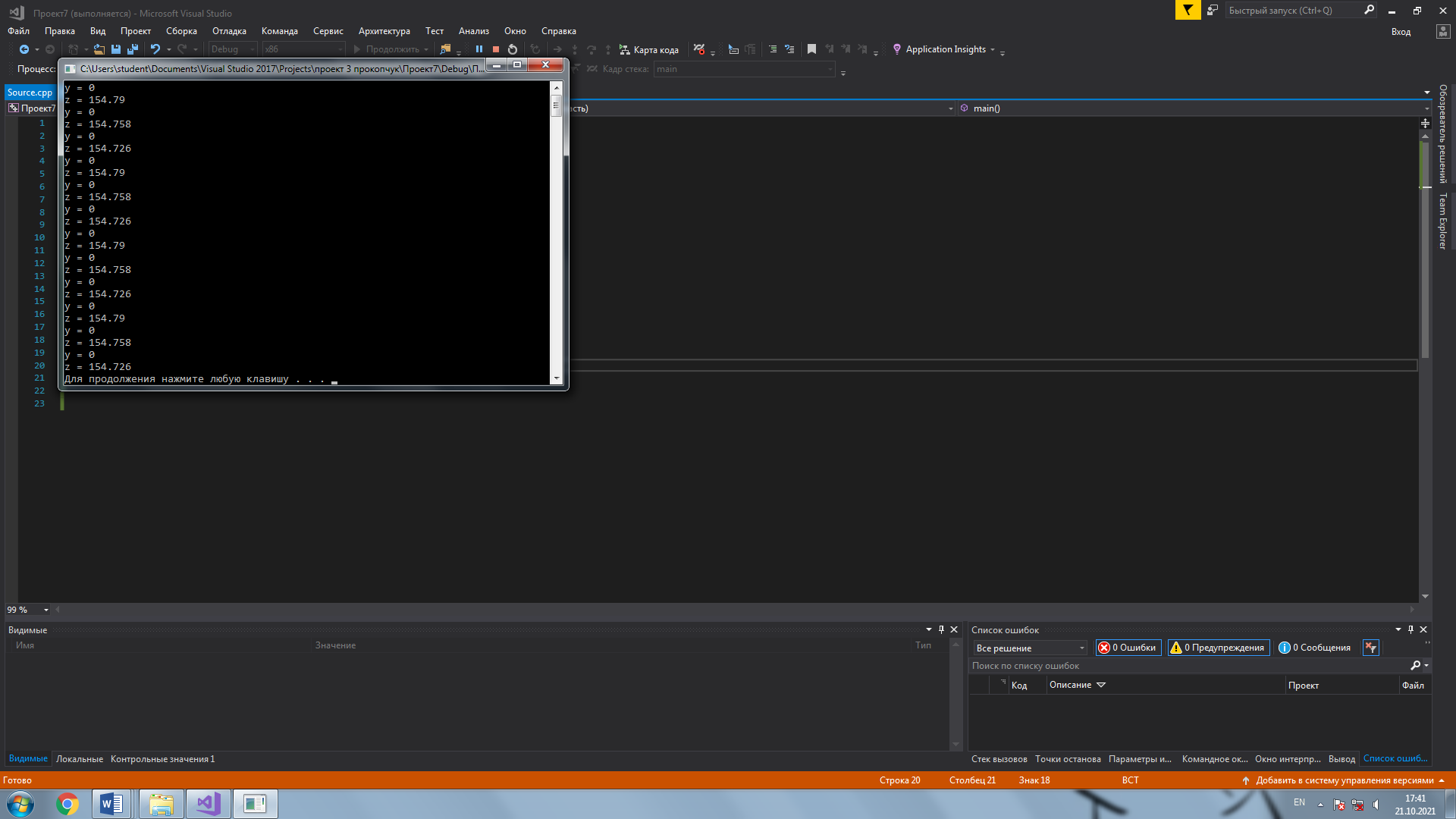
cout << "z = " << z << endl;

j = j + 0.5;

}

}

}



j, a = 24000, b = 5, m[3] = { 0.4, -1, 1.9 }

Начало



Вывод y, z

Конец

j=j+0.5

j<=3

i=0,3

да

j=2

нет

## Лабораторная работа № 7. Отладка программ

В соответствии со своим вариантом написать программу и отладить ее для задачи, представленной в таблице ниже. Опробовать средства ***интерактивной*** отладки **Шаг с обходом**. Вызвать окно **Контрольное значение** **1**, поместить туда имена переменных и проконтролировать их значения в процессе выполнения программы.

В отчете представить результаты в окне **Отладчика** (окно **Видимые** иокно **Контрольное значение** **1**).

Для той же программы представить результаты в окнах **Отладчика** для интерактивной отладки **Шаг с заходом**.

|  |  |
| --- | --- |
|  |  |

#include<iostream>

using namespace std;

int main()

{

double a = 4, c = 3.7, k = 1, d = 0.519, x, y;

while (c <= 5);

{

x = tan(a \* a - 1) / (d + 1);

if (3 \* x <= a \* c)

y = a \* k + d;

else if (3 \* x > a \* c)

y = cos(a \* k);

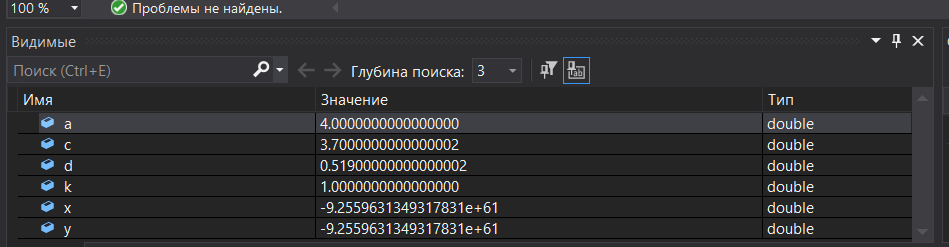
cout << "x = " << x << endl;

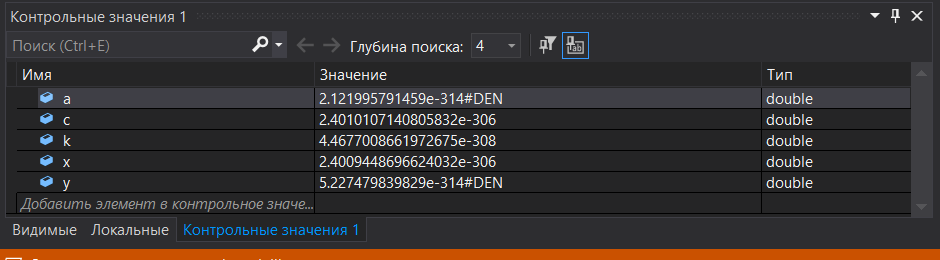
cout << "y = " << y << endl;

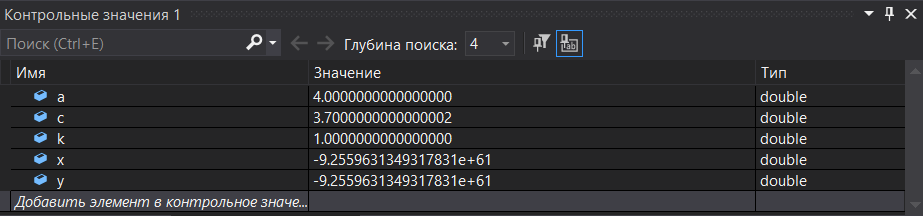
c = c + 0.1;

}

}

****

****

****

7. В соответствии со своим вариантом написать программу и отладить ее для задачи, представленной в таблице ниже. Опробовать средства ***планируемой*** отладки.

В отчете представить результаты в окне **Отладчика** (окно **Локальные** и окно **Контрольное значение 1**).

Определить количество трехзначных натуральных чисел, сумма цифр которых равна **n**.

#include<iostream>

using namespace std;

int main()

{

int n, a, b, c, t = 0;

setlocale(LC\_ALL, "RUS");

cout << "Введите число n ";

cin >> n;

if (n > 27)

cout << "Ваше число слишком большое";

else

for (a = 1; a <= 9; a++)

{

for (b = 0; b <= 9; b++)

{

for (c = 0; c <= 9; c++)

{

if (n == a + b + c)

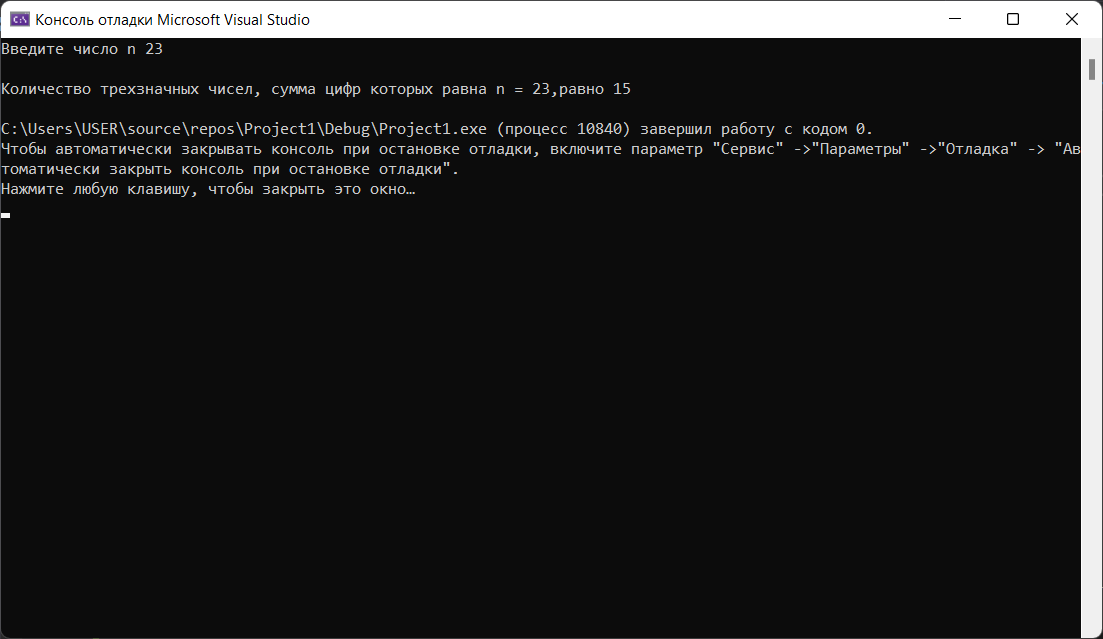
++t;

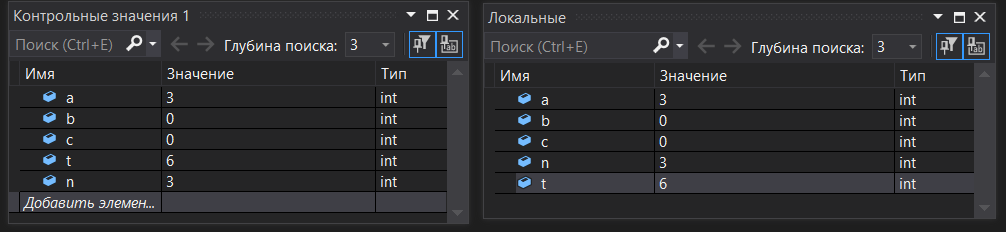
}

}

}

cout << endl << "Количество трехзначных чисел, сумма цифр которых равна n = " << n << ",равно " << t << endl;

}



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

В соответствии со своим вариантом разработать программы по условиям, приведенным в таблице ниже. Исходные данные ввести с клавиатуры. Произвести *отладку* программы.

|  |  |
| --- | --- |
|  |  |

#include<iostream>

using namespace std;

int main()

{

int i = 0, m = 5;

double y = 0, z, x[5] = { 6,2,0.9,0.1,5 };

while (i < m)

{

y += (x[i] - 2) \* (x[i] - 2);

https://studfile.net/html/65066/740/html__SsLNFSodM.ka18/htmlconvd-pPsCcv_html_dcc5d06956578a59.png

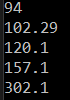
i++;

}

z = 8 \* x[2] + y;

cout << z << endl;

}



|  |  |
| --- | --- |
|  |  |

#include<iostream>

using namespace std;

int main()

{

int i = 0;

double a = 1, p, y = 0, x[5] = { 9,2.7,4.1,6,12 };

while(i<5)

{

if (a < x[i])

{

a = x[i];

}

i++;

}

i = 0;

while (i < 5)

{

y += (x[i] \* x[i]) + 1;

i++;

}

p = y + a;

cout << p << endl;

}

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

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.

|  |  |
| --- | --- |
| **Функция f(x)** | **Границы отрезков** |
| ex + 6 | a = 5, b = 11 |

**Способ трапеций**

#include<iostream>

using namespace std;

int main()

{

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

h = (b - a) / n;

x = a;

for (x; x <= (b - h); x += h)

{

s += h \* ( exp(x) + 6 + exp(x+h) + 6) / 2;

}

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

}

**Способ парабол**

#include<iostream>

using namespace std;

int main()

{

double a = 5, b = 11, n = 200, h, s1 = 0, s2 = 0, x, z;

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

x = a + 2 \* h;

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

{

s2 += exp(x) + 6;

x += h;

s1 += exp (x) + 6;

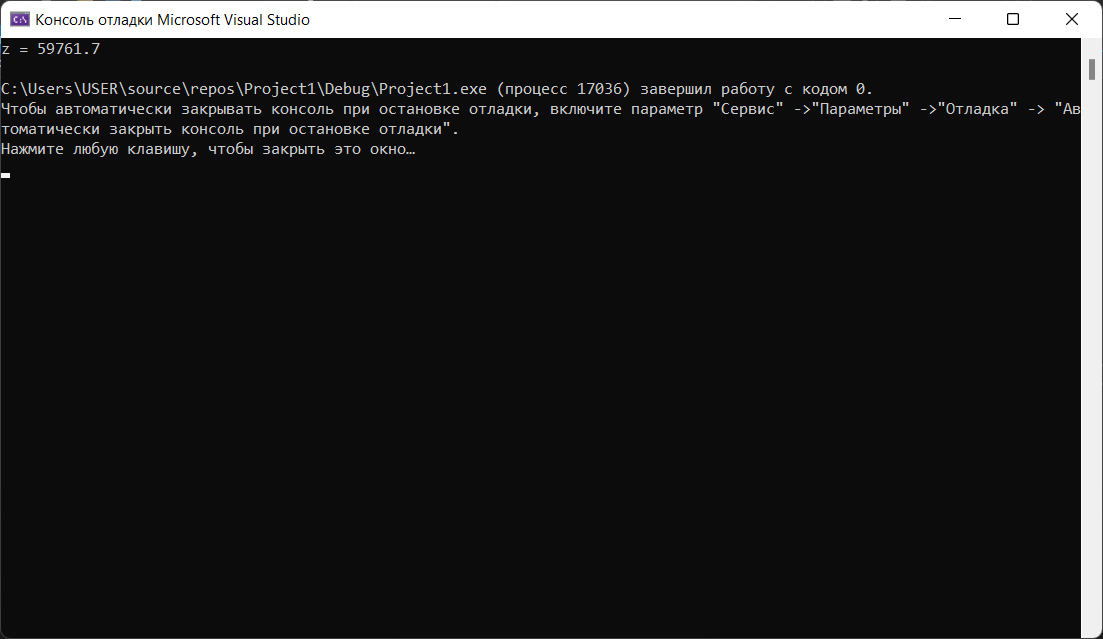
x += h;

}

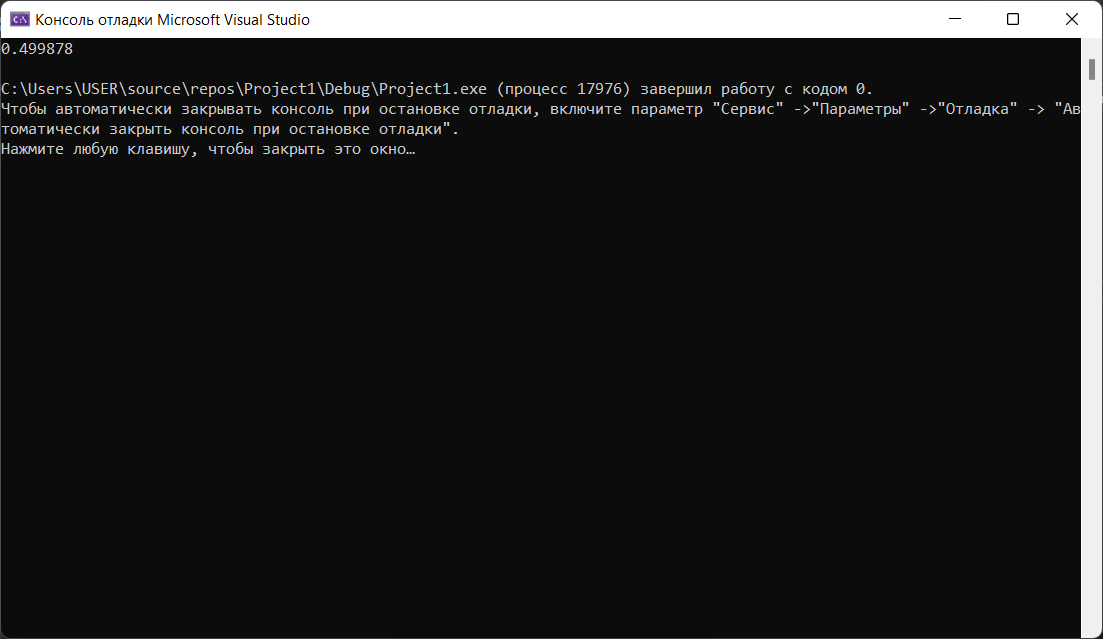
z = (h / 3) \* ( exp(a) + 6 + 4 \* (exp(a + h) + 6) + 4 \* s1 + 2 \* s2 + + exp(b) + 6);

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

}

Результаты отличаются на 4.5, но так как площадь большая, то это незначительная разница.

**Функция x3 + 2x – 4**

**Метод дихотомии**

#include <iostream>

using namespace std;

int main()

{

double a = 0, b = 0.5, e = 0.0001, x;

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

{

x = (a + b) / 2;

if ((pow(a, 3) + 2 \* a - 4) \* (pow(x, 3) + 2 \* x - 4) <= 0)

b = x;

else

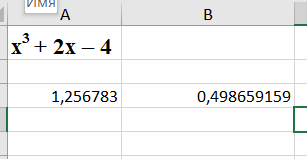
a = x;

}

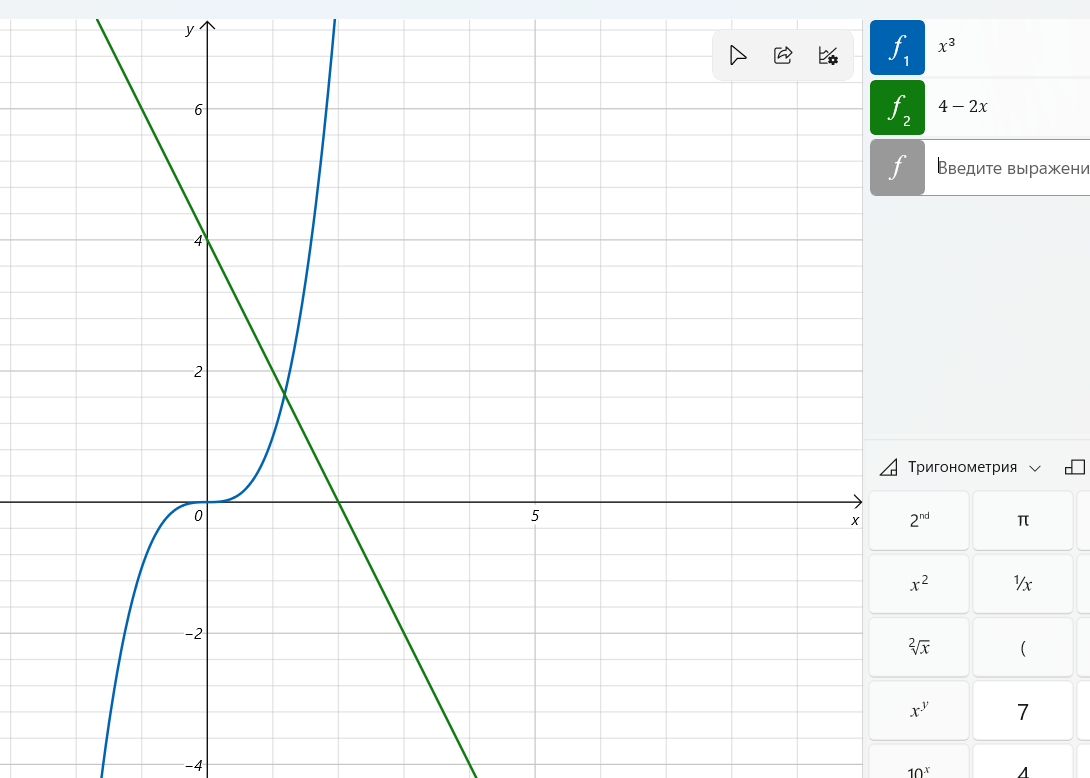
cout << x << endl;

}

**Метод вычисления с помощью Excel**

****

**Графический метод**



## Лабораторная работа № 10. Одномерные массивы

1.Найти минимальный элемент массива и заменить им четные по номеру элементы.

#include<iostream>

#include<locale>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

const int a = 100;

int arr[a]; int i, n, kmin = 0;

cout << "Введите размер массива (n < 100) ";

cin >> n;

if ((n <= 99) && (n - (int)n == 0))

srand((unsigned)time(NULL));

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

{

arr[i] = rand() / n;

cout << arr[i] << " ";

}

cout << endl;

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

if (arr[i] < arr[kmin])

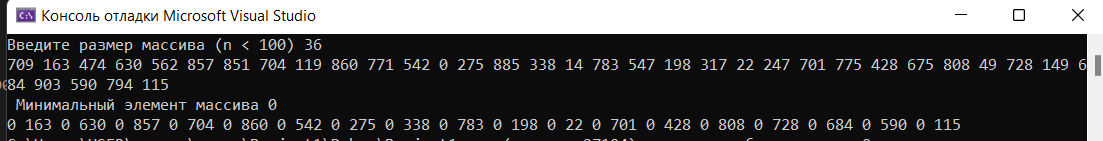
kmin = i;

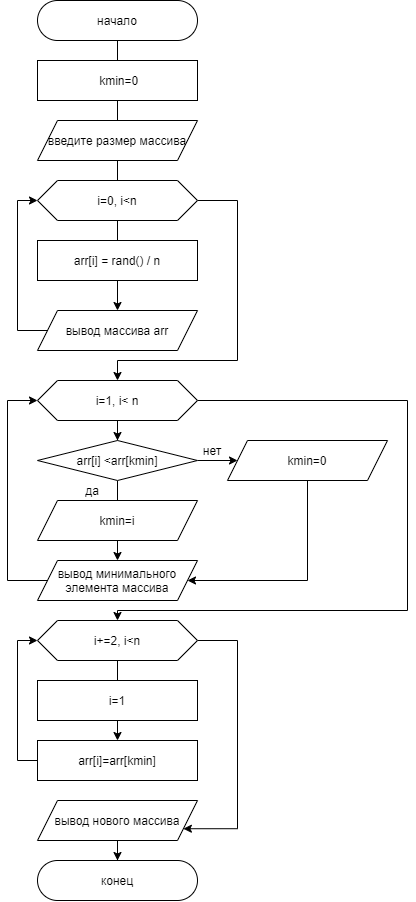
cout << " Минимальный элемент массива " << arr[kmin] << endl;

for (int i = 0; i < n; i += 2) arr[i] = arr[kmin];

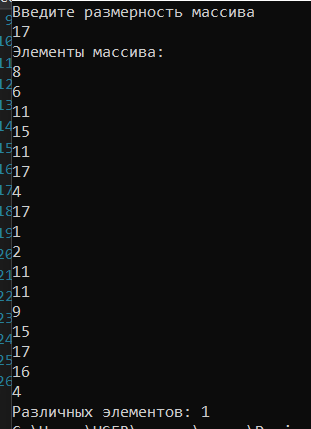
for (int i = 0; i < n; i++) cout << arr[i] << " ";

}





1. Задан массив из k символов. Определить количество различных элементов в массиве.

#include <iostream>

int main()

{

using namespace std;

setlocale(LC\_CTYPE, "Russian");

int i, k, t[100];

cout << "Введите размерность массива" << endl;

cin >> k;

char\* a;

a = new char[k];

cout << "Элементы массива: " << endl;

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

{

t[i] = rand() % k + 1;

cout << t[i] << endl;

}

int count = 0;

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

int j;

for (j = i - 1; j >= 0; j--)

if (a[j] == a[i]) break;

if (j < 0) count++;

}

cout << "Различных элементов: " << count;

}

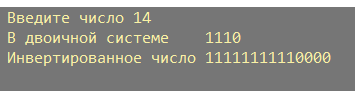
**Лабораторная работа № 11. Битовые операции**

В соответствии со своим вариантом разработать программы, использующие ***битовые*** операции для решения задач, представленных в таблице. Результаты одной из программ представить в **Отладчике**

1. Ввести целое число **A**. Инвертировать все биты с 2 по 14, включая эти биты. Вывести полученное число.

#include <iostream>

using namespace std;

int main()

{

system("color 8E");

int a, maska;

char tmp[33];

setlocale(LC\_CTYPE, "Russian");

cout << " Введите число ";

cin >> a;

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

cout << " В двоичной системе " << tmp << endl;

maska = 1;

maska = (maska << 13);

maska = maska - 1;

maska = maska << 1;

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

a = a ^ maska;

maska = 1 << 15;

a = a & (~maska);

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

cout << " Инвертированное число " << tmp << endl;

}

1. Извлечь 3 бита числа **А**, начиная с позиции **n**, и вставить в число **В**, начиная с позиции **m**

#include <iostream>

using namespace std;

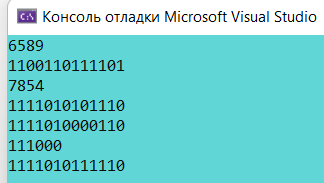
void main()

{

system("color B0");

char tmp[30];

int a, b, mask1 = 28, mask2 = ~mask1 << 1;

 cin >> a;

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

cout << tmp << endl;

cin >> b;

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

cout << tmp << endl;

\_itoa\_s((b & mask2), tmp, 2);

cout << tmp << endl;

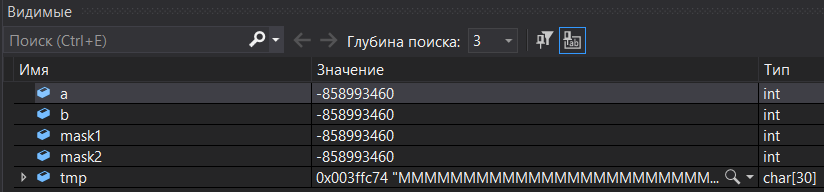
\_itoa\_s((a & mask1) << 1, tmp, 2);

cout << tmp << endl;

\_itoa\_s(((a & mask1) << 1) | (b & mask2), tmp, 2);

cout << tmp << endl;

}



**Лабораторная работа № 12. Использование указателей**

В соответствии со своим вариантом написать программы для условий задач из таблицы, представленной ниже, с использованием ***указателей***для доступа к элементам массива. Проанализировать содержимое памяти компьютера.

1. Заданы два массива **A** и **B**, каждый из **n** элементов. Подсчитать количество таких чисел **k**, для которых: **A[k] = B[k], A[k] > B[k]** и **A[k] < B[k]**.

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

int n, s1 = 0, s2 = 0, s3 = 0, i = 0, j = 0;

cout << "Введите количество элементов массива ";

cin >> n;

float\* a = new float[n];

float\* b = new float[n];

float\* pa, \* pb, k;

cout << "Заполните массив a" << endl;

for (pa = a; pa < a + n; pa++)

{

cout << "a[" << i << "] = "; cin >> \*(pa + 1);

i++;

}

cout << "Заполните массив b" << endl;

i = 0;

for (pb = b; pb < b + n; pb++)

{

cout << "b[" << i << "] = "; cin >> \*(pb + 1);

i++;

}

i = 0;

for (pa = a; pa < a + n; pa++)

{

j = 0;

for (pb = b; pb < b + n; pb++)

{

if (i == j)

{

if (\*(pa + 1) < (\*(pb + 1))) s1++;

if (\*(pa + 1) > (\*(pb + 1))) s2++;

if (\*(pa + 1) == (\*(pb + 1))) s3++;

}

j++;

}

i++;

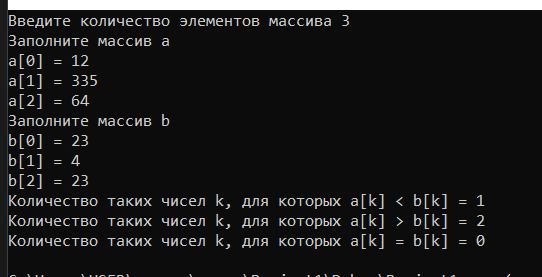
}

cout << "Количество таких чисел k, для которых a[k] < b[k] = " << s1 << endl;

cout << "Количество таких чисел k, для которых a[k] > b[k] = " << s2 << endl;

cout << "Количество таких чисел k, для которых a[k] = b[k] = " << s3 << endl;

}



2.Даны два массива **x** и **y**. Найти количество одинаковых элементов в этих массивах, т. е. количество пар **x[i]** **= y[j]** для некоторых **i** и **j**.

#include <iostream>

#include <ctime>

#include <locale>

using namespace std;

void main()

{

int i, j, n = 0;

double x[100], y[90];

srand((unsigned)time(NULL));

for (i = 0; i < 100; i++)

{

\*(x+i) = rand() % 200;

}

for (j = 0; j < 90; j++)

{

\*(y+j) = rand() % 100;

}

for (i = 0; i < 100; i++)

{

j = 0;

for (j; j < 90; j++)

{

if (\*(x+i) == \*(y+j))

{

n++;

 break;

}

}

}

cout << n << endl;

}

## Лабораторная работа № 13. Обработка символьной информации

Зашифровать введенную с клавиатуры строку, поменяв местами первый символ со вторым, третий с четвертым и т. д.

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

char s[100], tmp;

cin.getline(s, 100, '\n');

for (int i = 1; \*(s + i); i += 2)

{

tmp = s[i - 1];

s[i - 1] = s[i];

s[i] = tmp;

}

cout << s << endl;

}



1. Дана строка символов, состоящая из букв, цифр, запятых, точек, знаков «+» и «–». Найти количество запятых и точек

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

int dots = 0, comma = 0;

char s[100], tmp;

cin.getline(s, 100, '\n');

for (int i = 0; \*(s + i); i++) {

if (\*(s + i) == '.') dots++;

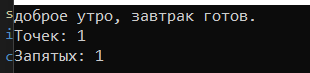
if (\*(s + i) == ',') comma++;

}

cout << "Точек: " << dots << endl;

cout << "Запятых: " << comma << endl;

}



**Лабораторная работа № 14. Многомерные массивы**

1 . Дана целочисленная матрица **A(n, m)**. Вычислить сумму и произведение нечётных отрицательных элементов матрицы, удовлетворяющих условию **| ai j| < i**.

#include <iostream>

#include <locale>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

const int n = 10, m = 15;

int a[n][m], summ = 0, prod = 1;

srand((unsigned)time(NULL));

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

for (int j = 0; j < m; j++)

{

a[i][j] = rand() % 30 - 15;

cout << a[i][j] << "\t";

}

cout << endl;

}

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

if (a[i][j] < 0 && a[i][j] % 2 != 0 && abs(a[i][j]) < i) {

summ += a[i][j];

prod \*= a[i][j];

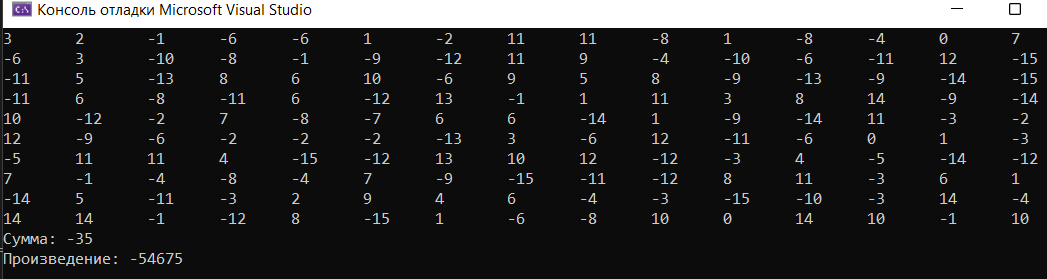
}

}

cout << "Сумма: " << summ << endl;

cout << "Произведение: " << prod << endl;

}



2.Проверить, все ли строки матрицы содержат хотя бы один положительный элемент. Если да, то изменить знаки всех элементов матрицы на обратные.

#include <iostream>

#include <locale>

using namespace std;

int main()

{

const int n = 10, m = 15;

int a[n][m], count = 0, icount = 0;

srand((unsigned)time(NULL));

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

for (int j = 0; j < m; j++) {

\*(a+i+j) = rand() % 30 - 15;

cout << a[i][j] << "\t";

}

cout << endl;

}

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

for (int j = 0; j < m; j++) if (a[i][j] > 0) icount++;

if (icount > 0) count++;

}

if (count == n) {

cout << "В матрице все строки содержат хотя бы один положительный элемент. Измененная матрица:" << endl;

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

for (int j = 0; j < m; j++) {

\*(\*(a+i)+j) \*= -1;

cout << \*(\*(a+i)+j) << "\t";

}

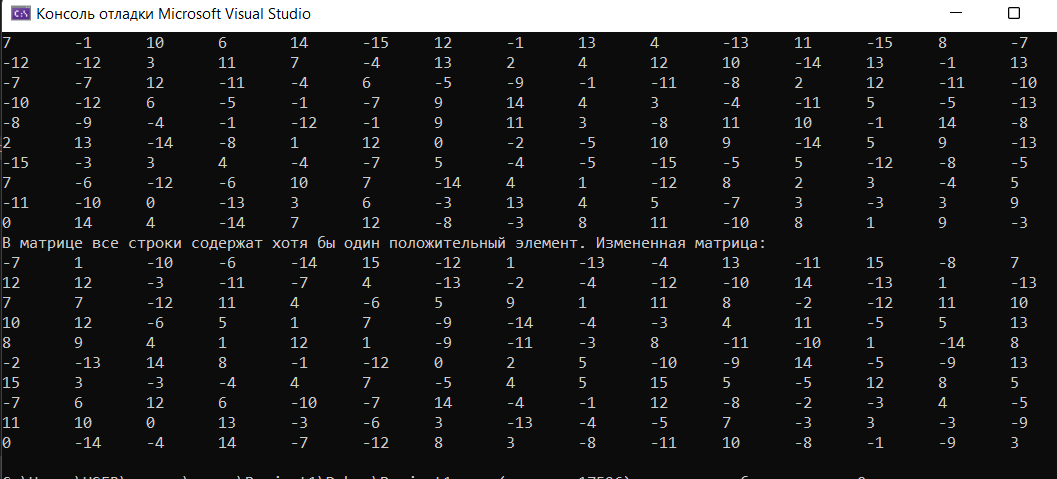
cout << endl;

}

}

else cout << "В матрице не все строки содержат положительный элемент.";

}



**Лабораторная работа № 15. Динамическое выделение памяти**

В одномерном массиве, состоящем из n вещественных элементов, вычислить количество элементов массива, меньших некоторого заданного числа с, и сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.

|  |
| --- |
| #include <iostream> #include <ctime> using namespace std; int main() {     const int rmin = -99, rmax = 99;     int n, count = 0;     float \*a, c, s = 0;     cout << "Введите кол-во элементов массива: "; cin >> n;     cout << "Введите число с: "; cin >> c;     if(!(a = (float\*)malloc(n\*sizeof(int)))) {          puts("Не хватает памяти.");          return 0;      }     srand((unsigned) time(NULL));     cout << "Сгенерированный массив: ";     for (int i = 0; i < n; i++) {         \*(a + i) = (((double) rand() / (double) RAND\_MAX) \* (rmax - rmin) + rmin);         printf("%f  ", \*(a + i));     }     for (int i = 0; i < n; i++) {         if (\*(a + i) < c) count++;         s += \*(a + i);         if (\*(a + i) < 0) s = 0;     }     printf("\nКол-во элементов массива, меньших числа С: %d\nСумма элементов после последнего отрицательного: %f", count, s);  Free(a); } |

## https://lh5.googleusercontent.com/zg52xfB7K2FRVYbG3nXcV3SUkQXjnXUPhKOMPIYwzzR-LbWBQNSQQHS9GR6nkoJodrP5nh5hxish16uCDoZDSWVHn65lwPVTm3M3SsVtCKKy1LNmSrpEAP4hagaaJ9vyDEBlZiNI

Проверить, все ли строки матрицы содержат хотя бы один нулевой элемент. Если нет, то заменить значения всех отрицательных элементов матрицы на нулевые.

|  |
| --- |
| #include <iostream> #include <ctime> using namespace std; int main() {     const int rmin = -5, rmax = 5;     int n, \*\*a; bool b = true;     cout << "Введите размерность матрицы: "; cin >> n;     a = new int\*[n];     srand((unsigned) time(NULL));     cout << "Сгенерированная матрица: " << endl;     for (int i = 0; i < n; i++) {         a[i] = new int[n];         for (int j = 0; j < n; j++) {             a[i][j] = (int)(((double) rand() / (double) RAND\_MAX) \* (rmax - rmin) + rmin);             printf("%d\t", a[i][j]);         }         cout << endl;     }     for (int i = 0; i < n; i++) {         for (int j = 0; j < n; j++) {             if (a[i][j] == 0) break;             if (j == n - 1) {                 b = false; break;             }         }     }     if (!b) {         cout << "Измененная матрица:" << endl;         for (int i = 0; i < n; i++) {             for (int j = 0; j < n; j++) {                 if (a[i][j] < 0) a[i][j] = 0;                 printf("%d\t", a[i][j]);             }             cout << endl;         }     } else cout << "В каждой строке матрицы есть хотя бы один нуль."; delete[] a;  } |

## https://lh6.googleusercontent.com/sl2CK3ZKtYW0QnD_0qPjgVF7EO0sXEN-3N8nmd2l_rgbdpV0VkX9RvHFamNd5sKbwmt7vfi0VSXX3MazSm7SP7cLFO2FXZhkp9Ua-D-HG49rz300BndbsRp04nlulHDvTUSAwHxHhttps://lh6.googleusercontent.com/OxMYKxdnD70LTxY8WEMZNU1sgKrDMsWN4v9F1AazGRt3w1QaK9N0rpSwFY8OixI2siutr53BJDXr3tgfK4Dnq2udRLPs3jlKiKn2LZrBbS_r7_FveOKOHMAy6ED3lvqxp58GhQ7a

## Лабораторная работа № 16. Функции пользователя

1. Найти сумму элементов, лежащих ниже главной диагонали в целочисленном массиве A[n, n].

2. Написать программу, которая вводит несколько строк текста и символ «m» и использует функцию, чтобы определить суммарное число вхождений символа в текст.

#include <iostream>

using namespace std;

void Fill(int\*\* matrix, int sz, int max\_value)

{

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

{

for (int j = 0; j < sz; j++)

{

matrix[i][j] = rand() % (max\_value + 1);

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

void getstring(char\*\* string\_array, int x)

{

string\_array[x] = new char[200];

cin.getline(string\_array[x], 200);

}

void define\_amount(char\*\* string\_array, int x)

{

int amount = 0;

for (int i = 0; string\_array[x][i] != '\0'; i++)

{

if (string\_array[x][i] == 'm') amount++;

}

cout << "Количество вхождений символа m в строку " << x + 1 << ": " << amount << endl;

}

void First()

{

int sz, sum = 0;

cout << "Размер матрицы: "; cin >> sz;

cout << "\n\n";

int\*\* matrix = new int\* [sz];

for (int i = 0; i < sz; i++) matrix[i] = new int[sz];

Fill(matrix, sz, 100);

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

{

for (int j = 0; j < sz; j++)

{

if (j != i)

{

sum += matrix[i][j];

}

else break;

}

}

cout << "\n\nСумма элементов под главной диагональю: " << sum << "\n\n";

// Free space

for (int i = 0; i < sz; i++) delete[] matrix[i];

delete[] matrix;

}

void Second()

{

int strings\_amount;

cout << "Количество строк: "; cin >> strings\_amount;

char\*\* string\_array = new char\* [strings\_amount];

cout << endl;

for (int i = 0; i < strings\_amount; i++)

{

cout << "\nСтрока " << i + 1 << ":\n";

if (i == 0) cin.ignore();

getstring(string\_array, i);

}

cout << "\n\n";

for (int i = 0; i < strings\_amount; i++)

{

define\_amount(string\_array, i);

}

for (int i = 0; i < strings\_amount; i++) delete[] string\_array[i];

delete[] string\_array;

}

int main()

{

setlocale(LC\_ALL, "");

int x;

cout << "Задание 1:\nНайти сумму элементов, лежащих ниже главной диагонали в целочисленном массиве A[n, n].\n\nЗадание 2:\n";

cout << "Написать программу, которая вводит несколько строк текста и символ «m» и использует функцию, чтобы определить суммарное число вхождений символа в текст.\n\n";

cout << "Выполнить задание #"; cin >> x;

cout << "\n";

switch (x)

{

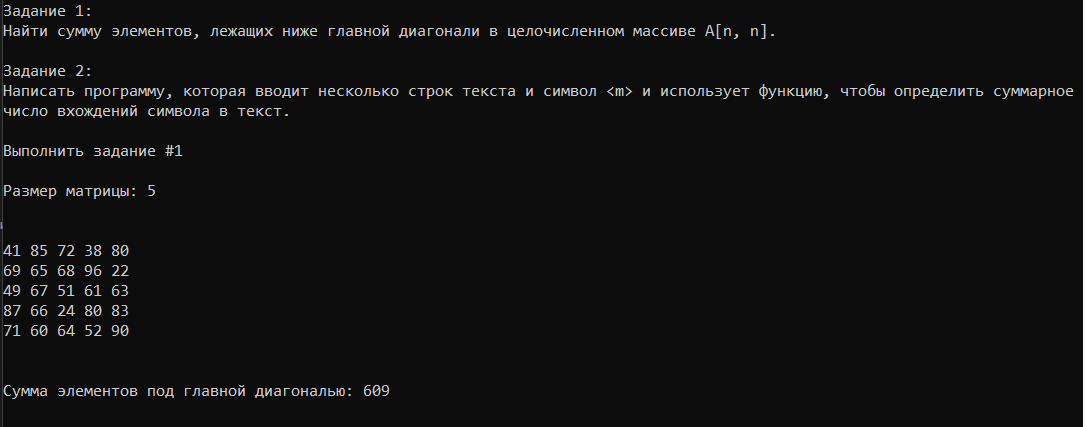
case 1: First(); break;

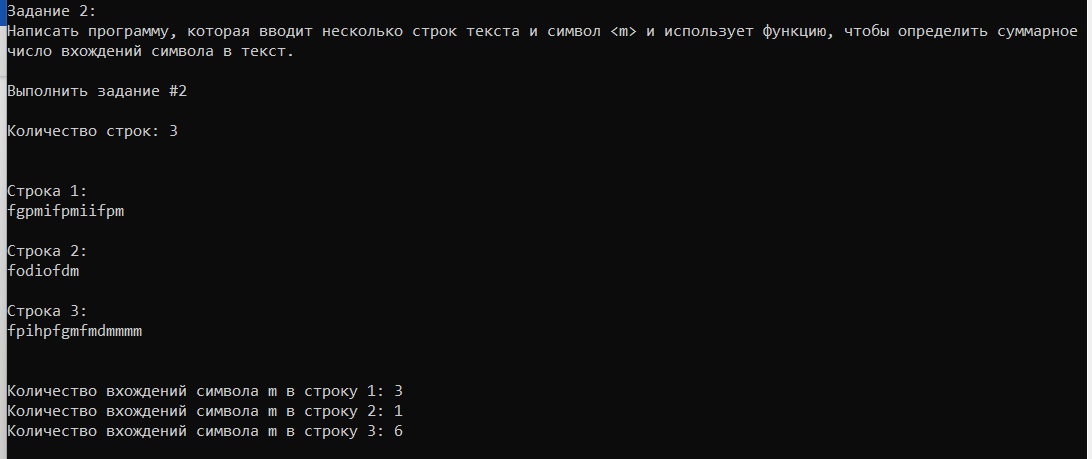
case 2: Second(); break;

default: cout << "Некорректный номер!\n\n"; exit(1);

}

}





## Лабораторная работа № 17. Указатели и ссылки при работе с функциями

1. В массиве из целых чисел определить сумму элементов, расположенных между первым и последним нулевыми элементами

#include <iostream>

using namespace std;

void fill(int\*, int\*, int); //прототип

int define(int\*, int\*);

void writer(int\*, int\*);

int main()

{

setlocale(LC\_ALL, "");

srand(time(0));

int sz, \* arr;

cout << "Размер массива: ", cin >> sz;

arr = new int[sz];

fill(arr, &sz, 10);

writer(arr, &sz);

cout << endl << define(arr, &sz) << endl;

delete[]arr;

}

void fill(int\* arr, int\* size, int max\_value)

{

for (int i = 0; i < \*size; i++) arr[i] = rand() % (max\_value);

}

int define(int\* arr, int\* size)

{

int sum = 0, first = -1, last = -1;

for (int i = 0; i < \*size; i++) if (arr[i] == 0)

{

first = i;

break;

}

for (int i = \*size; i > first; i--) if (arr[i] == 0)

{

last = i;

break;

}

if (first != -1 && last != -1)

{

for (int i = first; i < last; i++) sum += arr[i];

return sum;

}

else return 0;

}

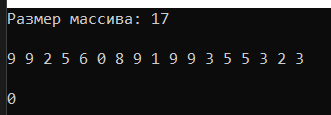
void writer(int\* arr, int\* size)

{

cout << endl;

for (int i = 0; i < \*size; i++) cout << arr[i] << " ";

cout << endl;

}

2. Если есть в матрице строка, все элементы которой положительны, то найти сумму этих элементов. Уменьшить все элементы матрицы на эту сумму.

#include <iostream>

#include <iomanip>

using namespace std;

void write(int\*\*, int\*, int\*);

void fill(int\*\*, int\*, int\*, int);

int find\_positive\_string(int\*\*, int\*, int\*);

int sum(int\*\*, int, int\*);

void reduce(int\*\*, int\*, int\*, int);

int main()

{

setlocale(LC\_ALL, "");

int\*\* matrix;

int strings, columns;

cout << "Количество строк матрицы: "; cin >> strings;

cout << "Количество столбцов матрицы: "; cin >> columns;

matrix = new int\* [strings];

for (int i = 0; i < strings; i++) matrix[i] = new int[columns];

fill(matrix, &strings, &columns, 100);

write(matrix, &strings, &columns);

if (find\_positive\_string(matrix, &strings, &columns) != -1)

{

reduce(matrix, &strings, &columns, sum(matrix, find\_positive\_string(matrix, &strings, &columns), &columns));

cout << "Обновленная матрица: ";

write(matrix, &strings, &columns);

}

for (int i = 0; i < strings; i++) delete[] matrix[i];

delete[] matrix;

}

void write(int\*\* matrix, int\* str, int\* col)

{

cout << endl;

for (int i = 0; i < \*str; i++)

{

for (int j = 0; j < \*col; j++)

{

cout << setw(4) << matrix[i][j] << " ";

}

cout << endl;

}

cout << endl;

}

void fill(int\*\* matrix, int\* str, int\* col, int max\_value)

{

srand(time(0));

for (int i = 0; i < \*str; i++)

{

for (int j = 0; j < \*col; j++)

{

matrix[i][j] = (rand() - (RAND\_MAX / 4)) % max\_value;

}

}

}

int find\_positive\_string(int\*\* matrix, int\* str, int\* col)

{

bool cond;

for (int i = 0; i < \*str; i++)

{

cond = true;

for (int j = 0; j < \*col; j++)

{

if (matrix[i][j] < 0) cond = false;

}

if (cond) return i;

}

return -1;

}

int sum(int\*\* matrix, int str, int\* col)

{

int sum = 0;

for (int i = 0; i < \*col; i++)

{

sum += matrix[str][i];

}

return sum;

}

void reduce(int\*\* matrix, int\* str, int\* col, int sum)

{

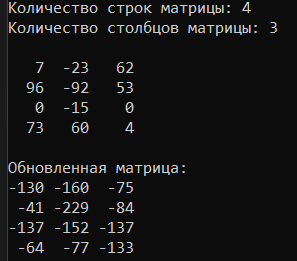
for (int i = 0; i < \*str; i++)

{

for (int j = 0; j < \*col; j++)

{

matrix[i][j] -= sum;

 }

}

}