Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Отчёт по лабораторным работам

По дисциплине «Основы алгоритмизации и программирования»

Выполнил:

Студент 1 курса 5 группы

Киравницин Кирилл Олегович

Преподаватель: доц. Пустовалова Н.Н

2023, Минск

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

**Лабораторная работа 1 на тему «Системы счисления»:**

**Вариант 4.**

1. 332(8 с/с) = 0011 0011 0010 (2 с/с) = 332 (16 c/c)
2. 1011 (2 c/c) = 11 (10 с/с)
3. 11111 (10 c/с) = 11111 (2 c/c)
4. 12 (8 с/с) = 1010 (2 с/с) = A (16 c/c)
5. 111 100 001 (2 c/c) = 741 (8 c/c) = 741 (16 c/c)
6. 73 (10 c/c) = 1001001 (2c/c)

**Лабораторная работа 2 на тему «Способы представления алгоритмов»:**

**Вариант 4.**

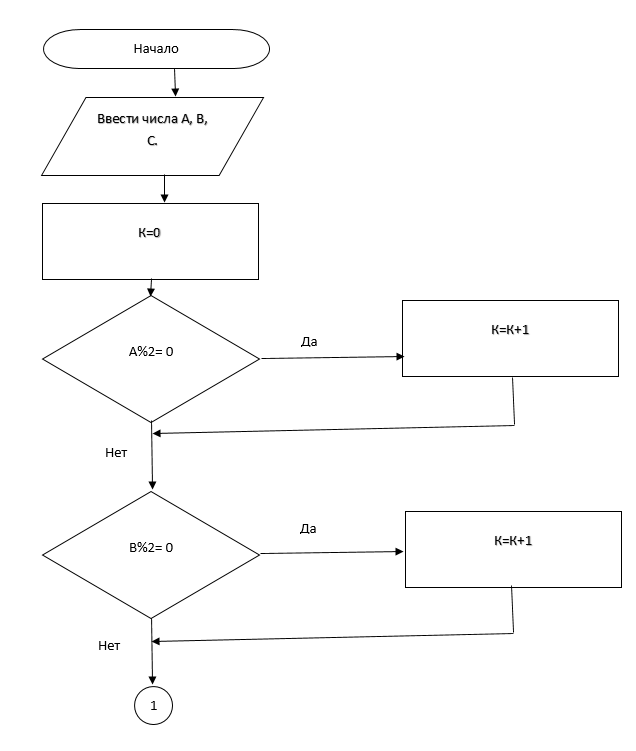
В приложении Word разработать *словесно-формульное* описание и *блок-схему* алгоритма в соответствии со своим вариантом для решения задачи из таблицы. Номер варианта определяет преподаватель.

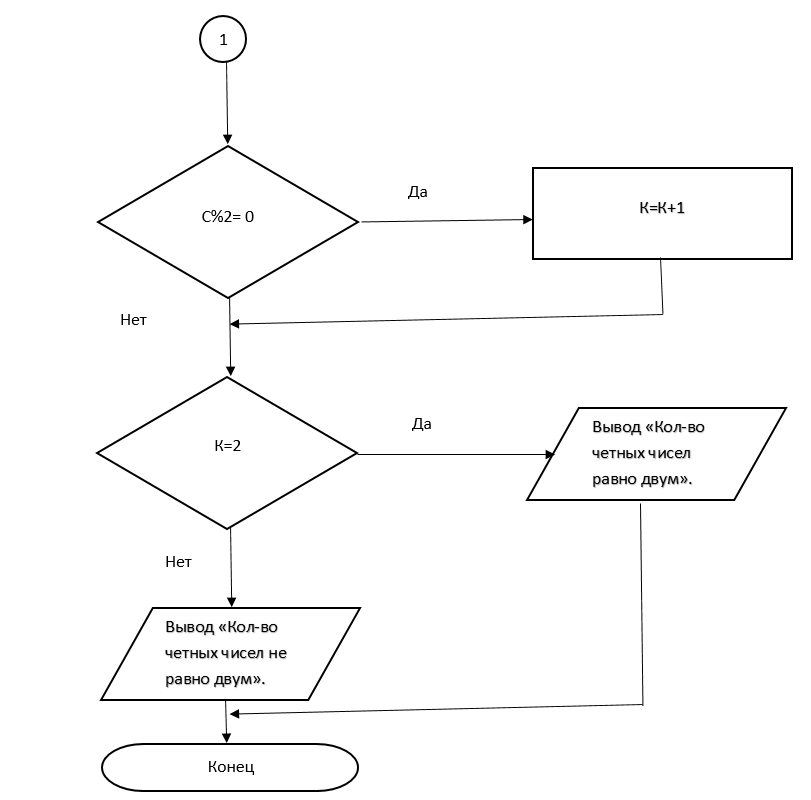
**Задание 1:** Определить, есть ли среди заданных целых чисел **A, B, C** два чётных числа.

**Словесно-формульное описание алгоритма:**

1. Ввод чисел А, B, С.
2. К=0
3. Если A%2 = 0 , то перейти к п.4, иначе – перейти к п.5.
4. K= K+1, затем перейти к п.8.
5. Если B%2 = 0 , то перейти к п.6, иначе – перейти к п.7.
6. К= К+1
7. Если С%2 = 0 , то перейти к п.8, иначе – перейти к п.9.
8. К=К+1, затем перейти к п.12
9. Если К=2, перейти к п.10, иначе перейти к п.11
10. Вывести на дисплей «Количество четных чисел равняется двум», затем перейти к п.15.
11. Вывести на дисплей «Количество четных чисел не равняется двум»
12. Конец вычислений.

**Блок-схема алгоритма:**

****

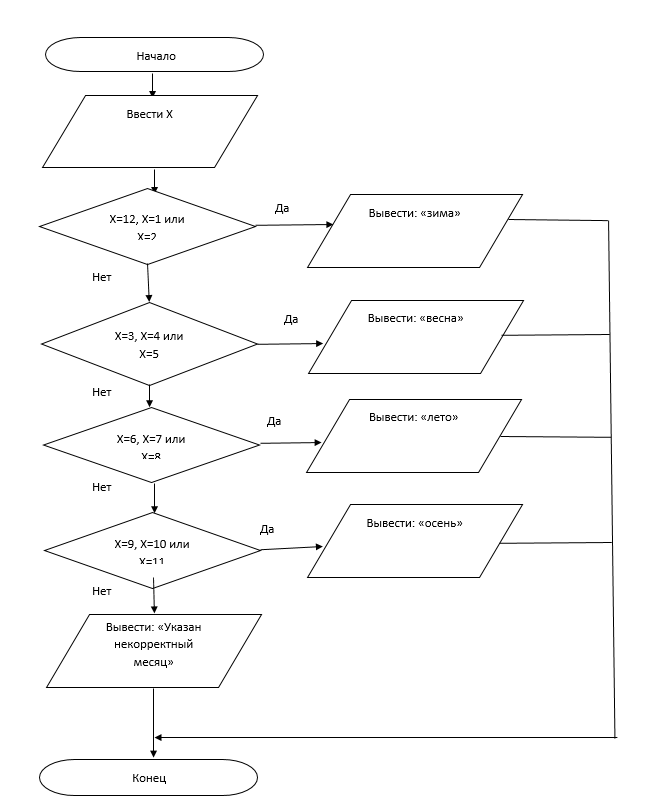
****

**Задание 2:** По номеру месяца напечатать пору года.

**Словесно-формульное описание алгоритма:**

1. Ввести номер месяца = x.
2. Если x равен: 12, 1 или 2, то перейти к п. 6
3. Если x равен: 3, 4 или 5, то перейти к п. 7
4. Если x равен: 6, 7 или 8, то перейти к п. 8
5. Если x равен: 9, 10 или 11, то перейти к п. 9
6. Вывести «зима», затем перейти к п. 10
7. Вывести «весна», затем перейти к п. 10
8. Вывести «лето», затем перейти к п. 10
9. Вывести «осень»
10. Конец вычислений.

**Блок-схема алгоритма:**

****

**Лабораторная работа 3 на тему «Основные элементы языка С++»:**

**Вариант 4.**

**Задание 1.** В таблице ниже приведены условия задач. В соответствии со своим вариантом разработать программу для решения задачи. Опробовать работу программы и проанализировать результаты.

**Исходные данные:**

**Решение:**

#include <iostream>

using namespace std;

int main()

{

int t = 6;

double y = -1.2;

int x = 4E5;

double p, q;

p = 2.6 \* t + cos(y / (3 \* x + y));

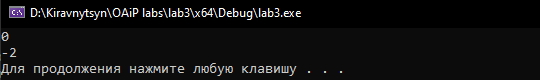
q = sin(t) / cos(t);

cout << p << endl;

cout << q << endl;

return 0;

}

**Результат: **

**Задание 2.** 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).

**Исходные данные:**



**Решение:**

#include <iostream>

using namespace std;

int main()

{

int m= 6;

double x= 1.4;

double z = 5E-7;

double y, w;

y = pow(1 + x, 1 / 2) - cos(2 / m);

w = 0.6 \* z - 2 \* exp( -2 \* y \* m);

cout << y << endl;

cout << w << endl;

return 0;

}

**Результат:**

****

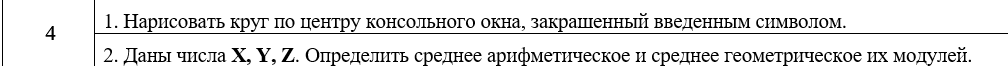
**Лабораторная работа №4 на тему «Ввод и вывод информации»**

**Вариант 4:**

**Условие:**

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

Для задачи № 1 использовать манипуляторы, в задаче № 2 ввод и вывод информации осуществить с использованием *потокового* и *форматированного* способов.



**Решение:**

**Задание 1:**

#include <iomanip>

#include <iostream>

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "Russian");

char c, probel;

probel = ' ';

cout << "Введите символ "; cin >> c;

system("cls");

cout << setw(80) << setfill(probel) << probel << endl;

cout << setw(80) << setfill(probel) << probel << endl;

cout << setw(80) << setfill(probel) << probel << endl;

cout << setw(80) << setfill(probel) << probel << endl;

cout << setw(80) << setfill(probel) << probel << endl;

cout << setw(80) << setfill(probel) << probel << endl;

cout << setw(80) << setfill(probel) << probel << endl;

cout << setw(80) << setfill(probel) << probel << endl;

cout << setw(80) << setfill(probel) << probel << endl;

cout << setw(80) << setfill(probel) << probel << endl;

cout << setw(50) << setfill(probel) << probel;

cout << setw(20) << setfill(c) << c << endl;

cout << setw(47) << setfill(probel) << probel;

cout << setw(26) << setfill(c) << c << endl;

cout << setw(46) << setfill(probel) << probel;

cout << setw(28) << setfill(c) << c << endl;

cout << setw(46) << setfill(probel) << probel;

cout << setw(28) << setfill(c) << c << endl;

cout << setw(46) << setfill(probel) << probel;

cout << setw(28) << setfill(c) << c << endl;

cout << setw(46) << setfill(probel) << probel;

cout << setw(28) << setfill(c) << c << endl;

cout << setw(46) << setfill(probel) << probel;

cout << setw(28) << setfill(c) << c << endl;

cout << setw(46) << setfill(probel) << probel;

cout << setw(28) << setfill(c) << c << endl;

cout << setw(46) << setfill(probel) << probel;

cout << setw(28) << setfill(c) << c << endl;

cout << setw(46) << setfill(probel) << probel;

cout << setw(28) << setfill(c) << c << endl;

cout << setw(46) << setfill(probel) << probel;

cout << setw(28) << setfill(c) << c << endl;

cout << setw(47) << setfill(probel) << probel;

cout << setw(26) << setfill(c) << c << endl;

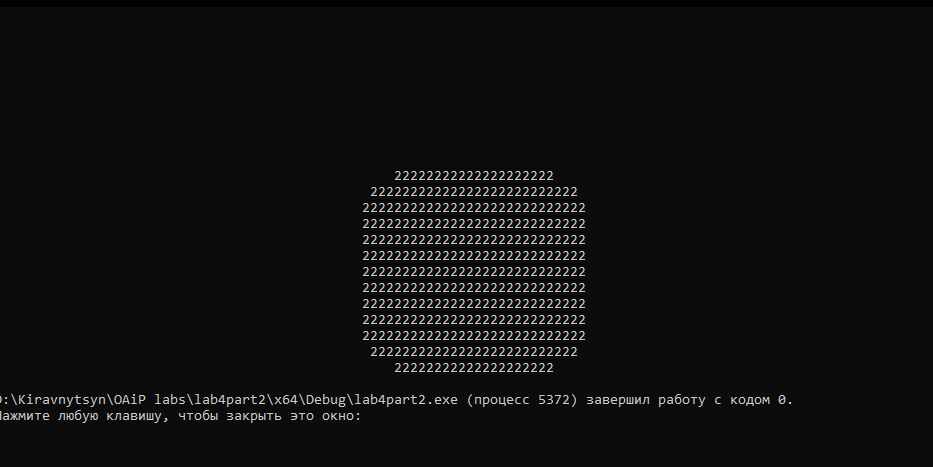
cout << setw(50) << setfill(probel) << probel;

cout << setw(20) << setfill(c) << c << endl;

return 0;

}

**Результат:**

****

**Задание 2:**

**Решение:**

#include <iostream>;

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

double x, y, z;

double m, n;

cout << "Введите x " << endl;

cin >> x;

cout << "Введите y " << endl;

cin >> y;

cout << "Введите z" << endl;

cin >> z;

m = (abs(x) + abs(y) + abs(z)) / 3;

n = pow(abs(x\*y\*z), 1.0/3.0);

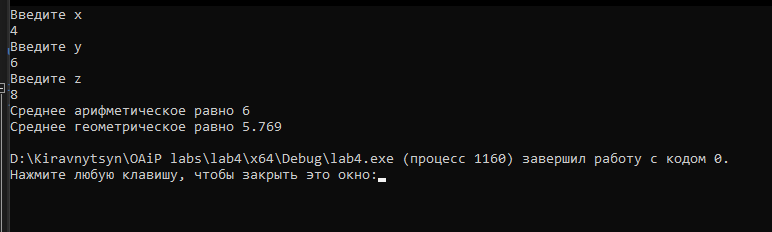
cout << "Среднее арифметическое равно " << m << endl;

cout << "Среднее геометрическое равно " << n << endl;

return 0;

}

**Результат:**

****

**Лабораторная работа №5 на тему «Разветвляющиеся программы»**

**Условие:**

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

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



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

**Задание 1:**

**Решение:**

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int a,b,c,k;

k = 0;

cout << "Введите число а" << endl;

cin >> a;

cout << "Введите число b" << endl;

cin >> b;

cout << "Введите число с" << endl;

cin >> c;

if (a % 2 == 0)

k = k + 1;

if (b % 2 == 0)

k = k + 1;

if (c % 2 == 0)

k = k + 1;

if (k == 2)

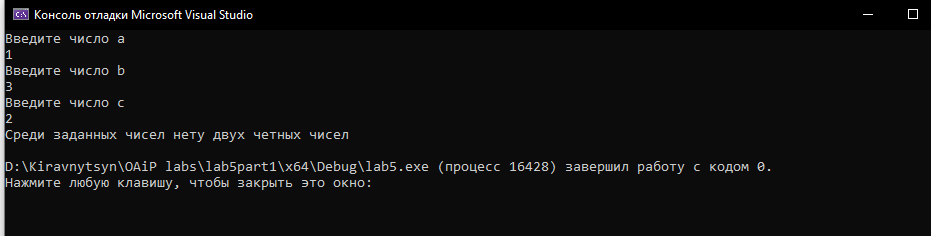
cout << "Среди заданных чисел есть два четных числа" << endl;

else

cout << "Среди заданных чисел нету двух четных чисел" << endl;

return 0;

**Результат:**

****

**Задание 2:**

**Решение:**

#include <iostream>

using namespace std;

int main(){

setlocale(LC\_ALL, "rus");

int x;

cout << "Введите номер месяца" << endl;

cin >> x;

switch (x) {

case 1: case 2: case 12: cout << "Зима"; break;

case 3: case 4: case 5: cout << "Весна"; break;

case 6: case 7: case 8: cout << "Лето"; break;

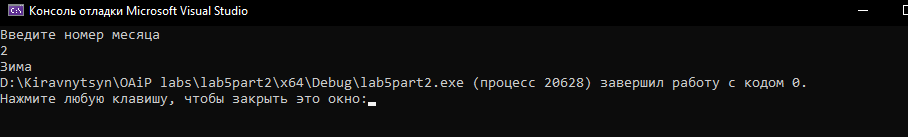
case 9: case 10: case 11: cout << "Осень"; break;

}

return 0;

}

**Результат:**

****

**Задание 3:**

**Решение:**

#include <iostream>

using namespace std;

int main() {

setlocale(0, "");

int k;

cout << "Нравится ли вам учится в БГТУ? (1 - Да, 2 - Нет)" << endl;

cin >> k;

switch (k) {

case 1: cout << "Поздравляем"; break;

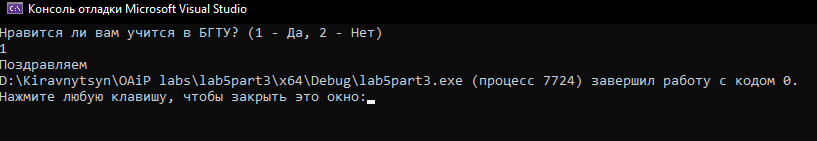
case 2: cout << "Неправильный ответ"; break;

default: cout << "Введена некорректная цифра ответа"; break;

}

}

**Результат:**

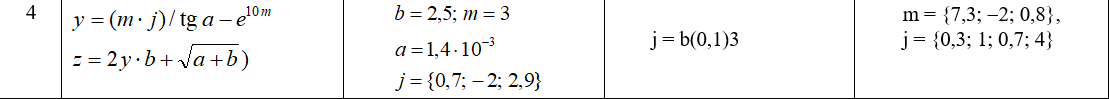
****

**Лабораторная работа №6 на тему «Циклические программы»:**

**Условие:**

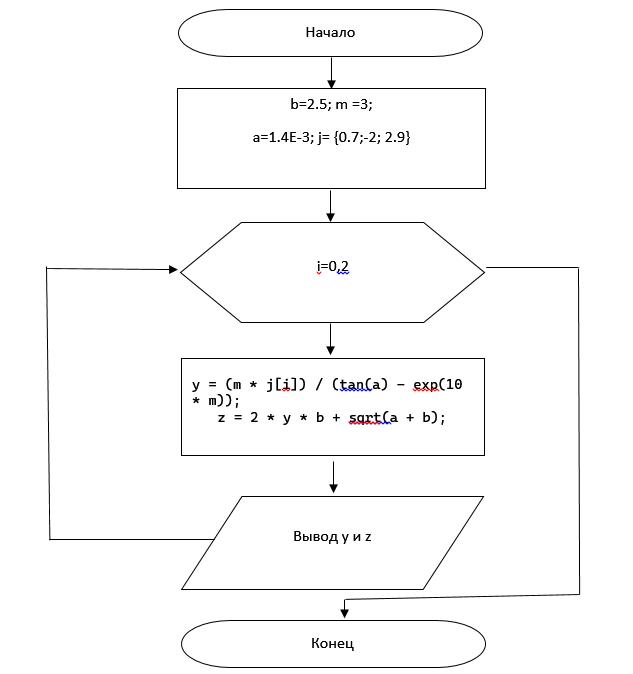
. В таблице приведены формулы и три варианта исходных данных, по которым надо разработать три ***блок-схемы*** и три ***циклические*** программы с одними и теми же расчетными формулами.

При наличии ошибок из-за некорректных исходных данных выполнить вычисления с другими числами.

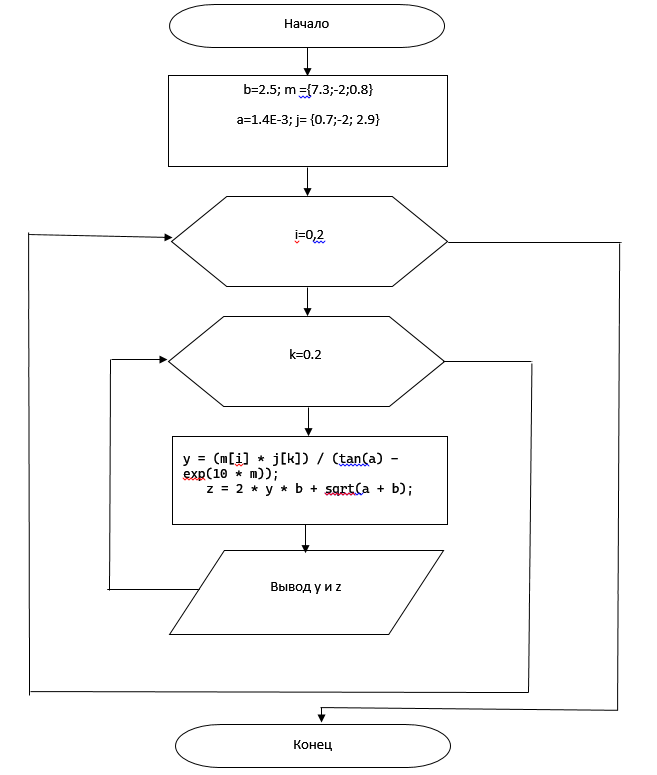
****

**Задание 1 и 2 (Цикл for и двойной цикл):**

**Блок-схема цикла for:**

****

**Блок-схема двойного цикла:**

****

**Решение:**

#include <iostream>

using namespace std;

void second(double a, double b, double z) {

double y;

double m[] = { 7.3,-2,0.8 }, j[] = { 0.3,1,0.7,4 };

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

for (int k=0; k < 4; k++) {

y = (m[i] \* j[k]) / (tan(a) - exp(10 \* m[i]));

cout << y << endl;

z = 2 \* y \* b + sqrt(a + b);

cout << z << endl;

}

}

}

int main() {

double y, z, b, m, a;

b = 2.5;

m = 3;

a = 1.4E-3;

double j[] = { 0.7,-2,2.9 };

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

y = (m \* j[i]) / (tan(a) - exp(10 \* m));

z = 2 \* y \* b + sqrt(a + b);

cout << y << " ";

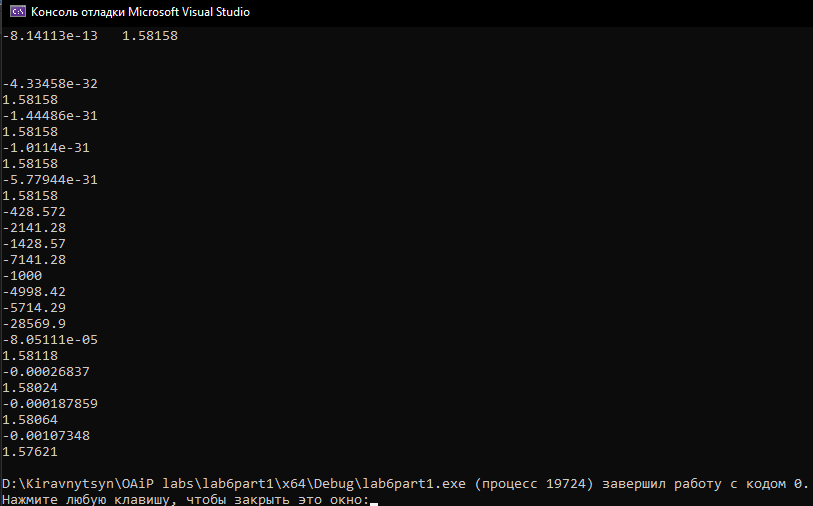
cout << z << endl;

}

cout << endl << endl;

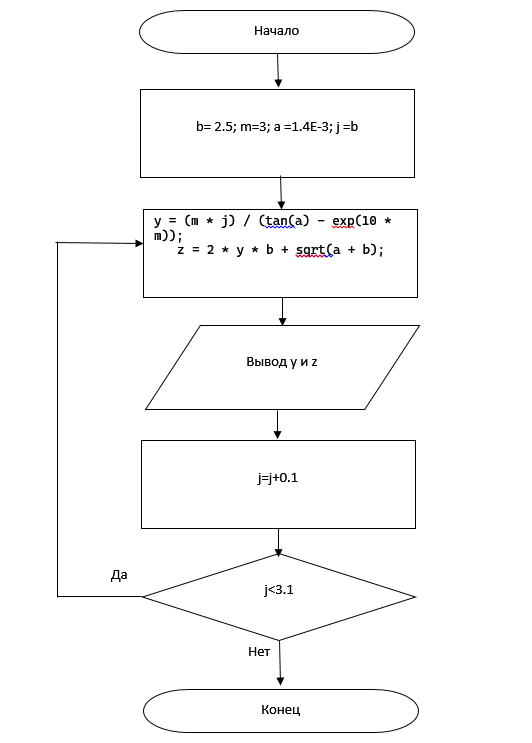
second(a, b, z);

**Результат:**

****

**Задание 2 (цикл while):**

**Блок-схема:**

****

**Решение:**

#include <iostream>

using namespace std;

int main() {

double y,z,b,a,j,m;

m = 3;

b = 2.5;

a = 1.4E-3;

j = 2.5;

while (j < 3.1)

{

y = (m \* j) / (tan(a) - exp(10 \* m));

z = (2\*y\*b) + sqrt(a + b);

cout << y << endl;

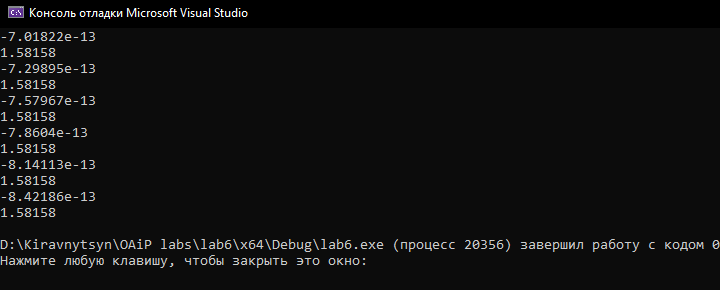
cout << z << endl;

j = j + 0.1;

}

}

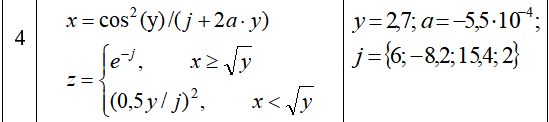
**Результат:**

****

**Лабораторная работа №7 на тему «Отладка программ»**

**Задание:** 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**).

**Условие: **

**Решение:**

#include <iostream>

using namespace std;

int main() {

double x, z, y, a;

double j[4] = { 6,-8.2,15.4,2 };

y = 2.7;

a = -5.5E-4;

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

x = pow(cos(y), 2) / (j[i] +2\*a\*y);

if (x >= sqrt(y))

z = exp(-j[i]);

if (x < sqrt(y))

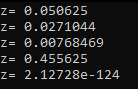
z = pow((0.5 \* y) / j[i], 2);

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

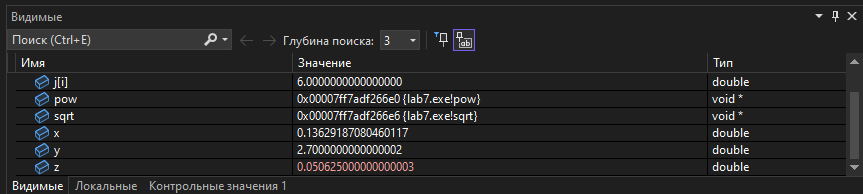
}

}

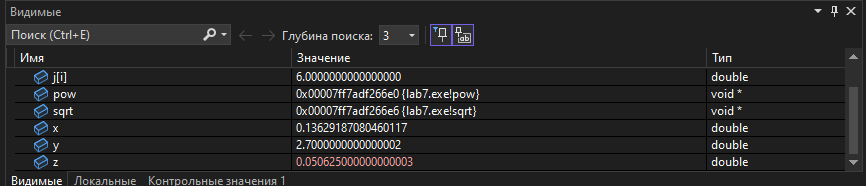
**Результат:**



**Отладка(шаг с обходом):**

****

**Отладка ( шаг с заходом):**

****

**Задание 2:** 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**).

**Условие: **

**Решение:**

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

int n, a, b;

cout << "Введите число а:" << endl;

cin >> a;

cout << "Введите число b:" << endl;

cin >> b;

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

if (a % i == 0 && b % i == 0) {

cout << "НОД равен " << i << endl;

break;

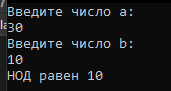
}

}

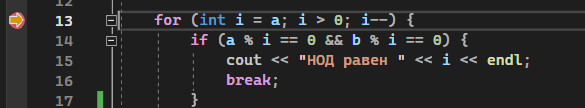
return 0;

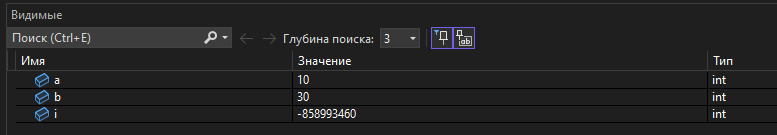
}

**Результат:**



**Отладка (планируемая):**

****

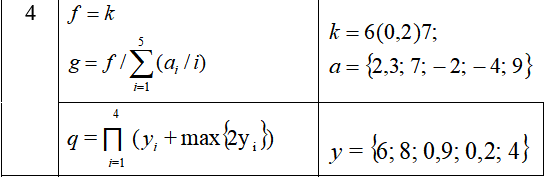
****

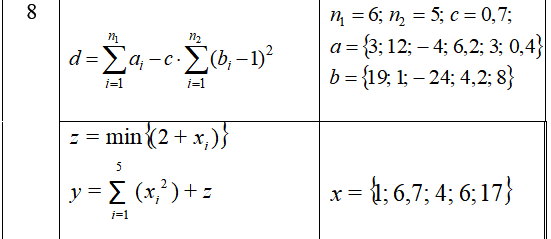
**Лабораторная работа №8 на тему «Вычисление сумм, произведений, экстремумов»**

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

Произвести *отладку* программы.

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

**Условия: **

****

**Решение:**

#include <iostream>

using namespace std;

int main() {

setlocale(0, "");

double g, sum =0;

double k = 6;

double a[] = { 2.3,7,-2,-4,9 };

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

for (int h = 1; h < 6;h++) {

while (k < 7.2) {

sum += a[i] / h;

g = k / sum;

cout << g << endl;

k = k + 0.2;

}

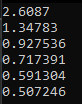
}

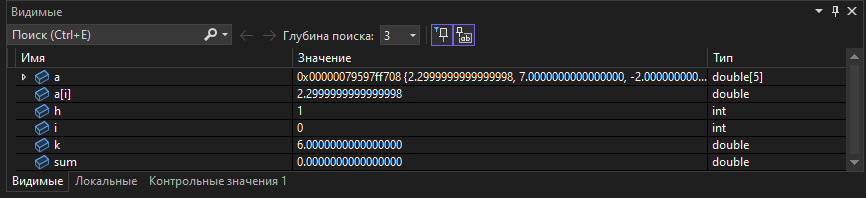
}

return 0;

}

**Результат:**

****

**Отладка: **

**Решение:**

#include <iostream>

using namespace std;

void third() {

double x[] = { 1,6.7,4,6,17 }, y = 0, min = x[0];

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

if (x[i] < min) {

x[i] = min;

}

}

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

y += pow(x[i], 2);

}

cout << "y= " << y+min+2 << endl;

}

void second() {

double d,c = 0.7,sum1=0,sum2=0;

double a[] = {3, 12, -4, 6, 2, 3, 0.4}, b[] = { 19,1,-24,4,2,8 };

for (int i = 0; i < 7; i++) {

for (int k = 0; k < 6; k++) {

sum1 += a[i] - c;

sum2 += pow((b[k] - 1), 2);

}

}

d = sum1 - sum2;

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

}

int main() {

double q=1,y[] = { 6,8,0.9,0.2,4 };

double max = y[0];

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

if (y[i] > max) {

y[i] = max;

}

}

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

q\*=(y[i] + 2\*max);

}

cout << q << endl;

second();

third();

}

**Результат:**

****

**Лабораторная работа №9 на тему «Решение инженерных задач на основе циклических программ»**

**Задание1:** 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.

**Задание2:** Написать программу вычисления корня уравнения методом ***дихотомии***. Точность вычислений принять равной **e** = 0,0001 для всех вариантов.

**Условие1:**

****

**Условие2:**

****

**Решение:** #include <iostream>

using namespace std;

void third() // метод Дихотомии

{

using namespace std;

setlocale(LC\_ALL, "RUS");

double e = 0.0001, x = 0, a = 1, b = 2;

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

x = (a + b) / 2;

if ((exp(x) - 3 - 1 / x) \* (exp(a) - 3 - 1 / a) <= 0) {

b = x;

}

else {

a = x;

}

}

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

}

void second() { // Метод параболл

setlocale(LC\_ALL, "RUS");

int n = 200, i = 1;

double h, x, s1 = 0, s2 = 0, z, a = 2, b = 3;

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

x = a + 2 \* h;

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

{

s2 += (exp(x)+1/x);

x += h;

s1 += (exp(x)+ 1/x);

x += h;

}

z = (h / 3) \* ((exp(a) - 1/a) + (4 \* (exp(a+h)-1/(a+h)) + (4 \* s1) + (2 \* s2) + (exp(b)-1/b)));

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

}

int main() { //Метод трапеций

setlocale(LC\_ALL, "rus");

double h, x, s = 0, a = 2, b = 3, n = 200;

h = (b - a) / n;

x = a;

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

{

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

}

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

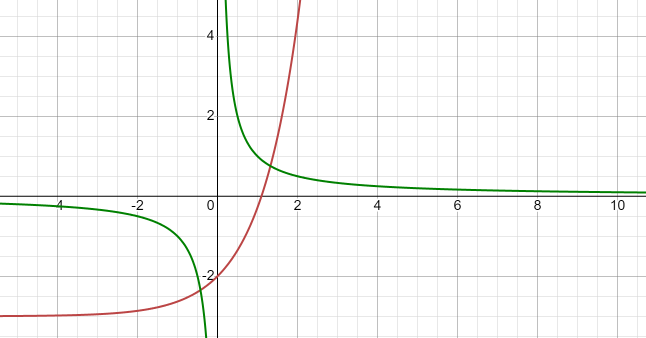
second();

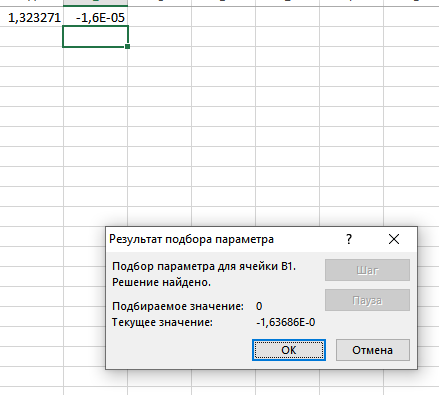
third();

}

**Результат:**

****

**Проверка: Из представленного ниже графика видим, что x=1.32 **

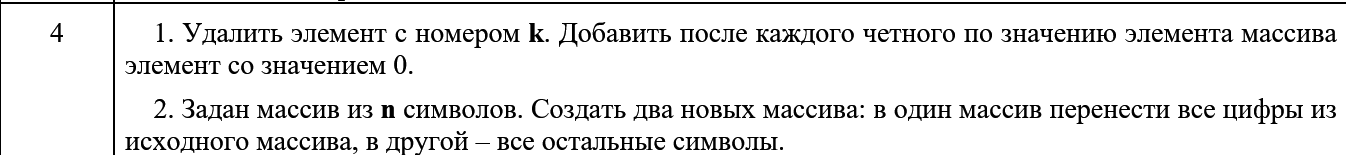
**Excel:** ****

**Лабораторная работа №10 на тему «Одномерные массивы»**

**Задание:** В соответствии со своим вариантом написать программы по условиям, представленным в таблице ниже. Использовать одномерный массив целых случайных чисел (диапазон от 0 до 99). Размер массива ввести с клавиатуры.

Представить результаты в окне **Отладчика**.

Для одной из программ составить ***блок-схему*** *а*лгоритма.

**Условие: **

**Решение:**

#include <iostream>

#include <locale>

using namespace std;

int second(){

char arr[] = "abc123def456";

char digits[sizeof(arr)], nonDigits[sizeof(arr)];

int dIndex = 0, ndIndex = 0;

for (int i = 0; arr[i] != '\0'; i++) {

if (arr[i] >= '0' && arr[i] <= '9') {

digits[dIndex++] = arr[i];

}

else {

nonDigits[ndIndex++] = arr[i];

}

}

digits[dIndex] = '\0';

nonDigits[ndIndex] = '\0';

cout << "Цифры: " << digits << endl;

cout << "Другие символы: " << nonDigits << endl;

return 0;

}

int main() {

setlocale(LC\_ALL, "rus");

const int maxSize = 99;

int n, k;

int a[maxSize];

cout << "Введите размер массива (<99)" << endl;

cin >> n;

if (n > 99) {

cout << "Неверное значение" << endl;

return 0;

}

srand((unsigned)time(NULL));

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

a[i] = rand() % 99;

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

}

cout << endl << "Введите номер элемента, который надо удалить" << endl;

cin >> k;

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

a[i] = a[i + 1];

}

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

if (a[i] % 2 == 0) {

for (int j = n; j > i + 1; j--) {

a[j] = a[j - 1];

}

a[i + 1] = 0;

i++; // Пропустить добавленный 0

n++; // Расширить размер массива

}

}

cout << "Массив после удаления элемента и добавления нулей к четным элементам: ";

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

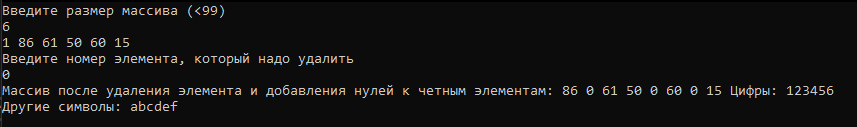
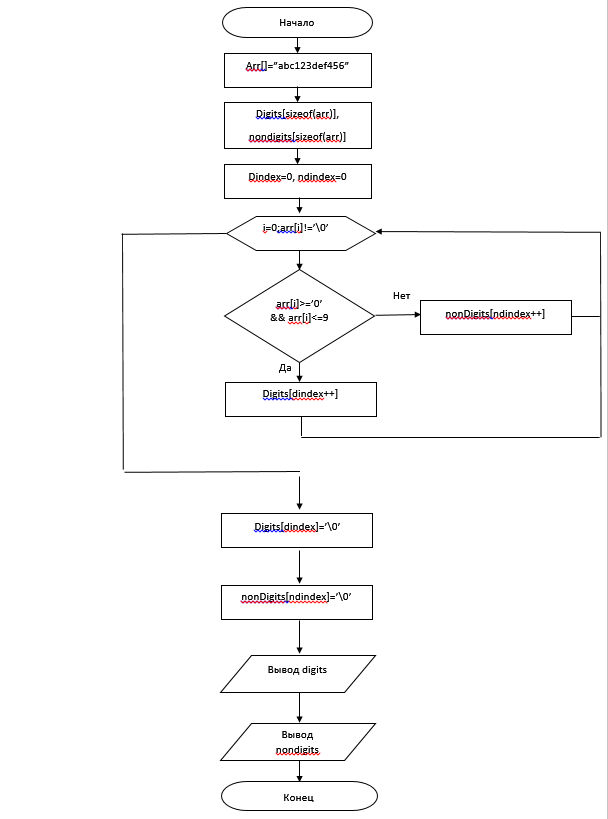
cout << a[i] << " ";

}

second();

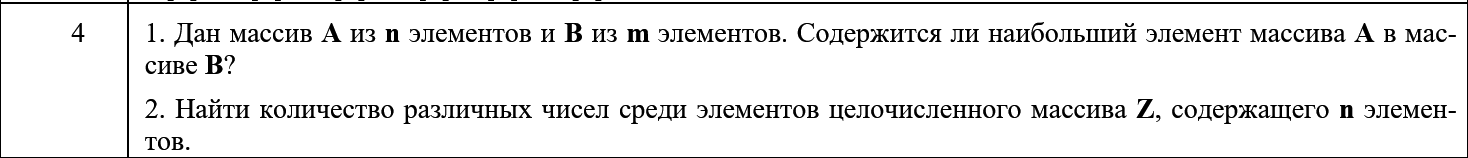
return 0;

}

**Результат: Блок-схема: **

**Лабораторная работа №11 на тему «Указатели»**

**Задание:** 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.

**Условие:** 

**Решение:**

#include <iostream>

using namespace std;

void second() {

cout << endl << endl << "Задание №2" << endl;

int n, result = 0;

cout << "Введите размер массива Z (< 100): ";

cin >> n;

if (n > 100 || n <= 0) {

cout << "Неверное значение";

return;

}

int\* Z = new int[n];

int\* pZ = Z;

srand(time(NULL));

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

\*(Z+i) = rand() % 5;

}

bool\* search = new bool[100];

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

search[i] = false;

}

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

if (!search[\*pZ]) {

search[\*pZ] = true;

result++;

}

pZ++;

}

cout << "Массив Z: ";

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

cout << Z[i] << " ";

}

cout << endl << "Количество различных чисел в массиве Z: " << result << endl;

delete[] Z;

delete[] search;

}

void main() {

setlocale(LC\_ALL, "rus");

cout << "Задание №1" << endl;

int n, m, max = 0;

cout << "Введите размер массива A (< 100): ";

cin >> n;

if (n > 100 || n <= 0) {

cout << "Неверное значение";

return;

}

cout << "Введите размер массив B (< 100): ";

cin >> m;

if (m > 100 || m <= 0) {

cout << "Неверное значение";

return;

}

int\* A = new int[n];

int\* B = new int[m];

int\* pA = A;

int\* pB = B;

srand(time(NULL));

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

\*(A+i) = rand() % 100;

}

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

\*(B+i) = rand() % 100;

}

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

if (pA[i] > max) max = pA[i];

}

bool cont = false;

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

if (max == \*(B+i)) {

cont = true;

break;

}

}

cout << "Элементы массива A: ";

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

cout << pA[i] << " ";

}

cout << endl << "Элементы массива B: ";

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

cout << pB[i] << " ";

}

cout << endl;

if (cont) {

cout << "Максимальный элемент массива A (" << max << ") содержится в массиве B.";

}

else {

cout << "Максимальный элемент массива A (" << max << ") не содержится в массиве B.";

}

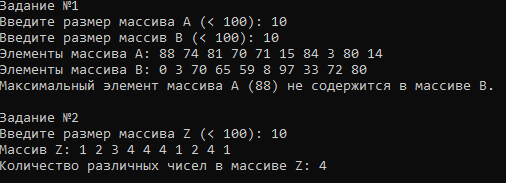
delete[] A;

delete[] B;

second();

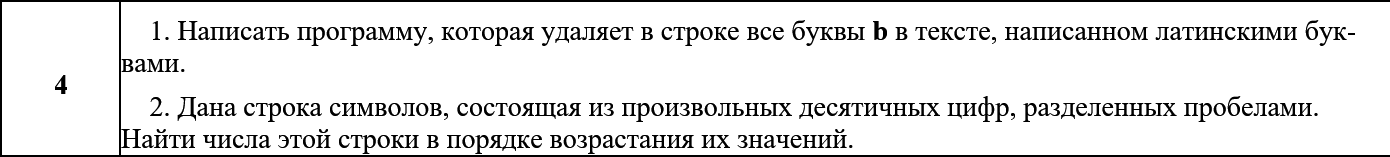
}

**Результат:**

****

**Лабораторная работа №12 на тему «Обработка символьной информации»**

**Задание:** Выполнить задания из таблицы ниже, используя ***индексы*** для доступа к элементам массивов в первой программе и ***указатели*** − во второй программе.

**Условие: **

**Решение:** #include <iostream>

using namespace std;

void second(){

char str[100];

cin >> str;

int count = strlen(str);

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

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

if (\*(str+i) > \*(str+j)) {

swap(\*(str+i),\*(str+j));

}

}

}

cout << str;

}

int main() {

setlocale(LC\_ALL, "rus");

char s='b', t[20];

int i, j, n;

cout << "Input string ";

gets\_s(t);

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];

second();

}

**Результат:**

****

**Лабораторная работа №13 на тему «Битовые операции»**

**Задание:** В соответствии со своим вариантом разработать программы, использующие ***битовые*** операции для решения задач, представленных в таблице.

**Условие:**

1. Используя битовые операции проверить, кратно ли четырем число **А**.
2. Установить в 1 в числе **А n** битов влево от позиции **p**.

**Решение:**

#include <iostream>

using namespace std;

void second() {

// Задание 2

int A;

int p = 2;

int n = 3;

cout << "Введите число ";

cin >> A;

A = A | ((1 << n) - 1) << p;

char buffer[33];

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

std::cout << "Число после установки битов: " << buffer << std::endl;

}

void main() {

// Задание 1

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;

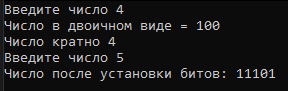
else

cout << "Число не кратно 4" << "\n" << endl;

second();

}

**Результат:**

****

**Лабораторная работа №14 на тему «Многомерные массивы»**

**Задание:** Выполнить задания из таблицы ниже, используя ***индексы*** для доступа к элементам массивов в первой программе и ***указатели*** − во второй программе.

**Условие:**

1. Дана матрица **A(n, m)**. Найти количество элементов этой матрицы, больших среднего арифметического всех её элементов.

2. Задана числовая матрица размером **n**x**m**.Определить количество таких элементов матрицы, у которых в строке слева от элемента находятся числа, меньшие его, а справа – большие.

**Решение:**

#include <iostream>

#include <ctime>

using namespace std;

void main() {

setlocale(0, "");

const int n = 3; const int m = 3;

int a[n][m],countOfElements=0;

double count = n \* m, sum=0;

double sr;

srand(time(NULL));

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

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

a[i][j] = rand() % 10;

cout << a[i][j] << ' ';

sum += a[i][j];

}

cout << endl;

}

// 1 Задание:

sr = sum / count;

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

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

if (a[i][j] > sr) countOfElements++;

}

}

cout << "\n Количество элементов больших, чем среднее арифметическое: " << countOfElements << endl;

//2 Задание:

bool status;

int countLine=0;

int countElem;

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

status = true;

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

for (int k = j; k >= 0; k--) {

if (\*(\*(a+i)+k) > \*(\*(a+i)+j)) status = false;

}

for (int k = j; k <= j; k++) {

if (a[i][k] < a[i][j]) status = false;

}

}

if (status) {

countLine++;

cout << "\n Строка " << i + 1 << " подходит под условие задачи" << endl;

}

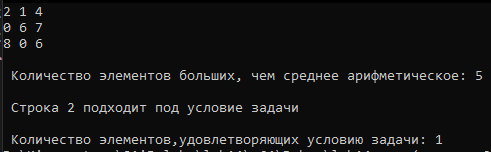
}

countElem = countLine \* (m - 2);

cout << "\n Количество элементов,удовлетворяющих условию задачи: " << countElem;

}

**Результат:**

****

**Лабораторная работа №15 на тему «Динамическое выделение памяти»**

**Задание:** В соответствии со своим вариантом написать программы с использованием ***динамических*** массивов для условий задач из таблицы. Начальные размерности массивов ввести с клавиатуры.

Первое задание выполнить с использованием ***функций*** для выделения динамической памяти на языке С, второе − с использованием операторов **new** и **delete**  на языке **С++**.

**Условие:**

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.

2. Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом – все остальные.

**Решение:**

#include <iostream>

using namespace std;

void second() {

int size,count=0;

cout << "Задание 2" << endl;

cout << "\nВведите размер массива : ";

cin >> size;

double\* array = new double[size];

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

cout << "Введите элемент " << i + 1 << ": ";

cin >> array[i];

}

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

if (array[i] == 0) {

swap(array[i], array[count]);

count++;

}

}

cout << "Массив после преобразования:" << endl;

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

cout << array[i] << ' ';

}

delete[] array;

}

void main() {

setlocale(0, "");

int size, firstNegative = -1, secondNegative = -1, min=0;

double sum = 0;

cout << "Задание 1" << endl;

cout << "Введите размер массива: ";

cin >> size;

double\* array = (double\*)malloc(size \* sizeof(double));

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

cout << "Введите элемент " << i + 1 << ": ";

cin >> array[i];

}

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

cout << array[i] << ' ';

}

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

if (array[min] > array[i]) {

min = i;

}

}

cout << "Индекс минимального элемента массива: " << min << endl;

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

if (array[i] < 0) {

if (firstNegative == -1) {

firstNegative = i;

}

else if (secondNegative == -1) {

secondNegative = i;

break;

}

}

}

if (firstNegative != -1 && secondNegative != -1) {

for (int i = firstNegative + 1; i < secondNegative; i++) {

sum += array[i];

}

cout << "Сумма элементов между двумя первыми отрицательными элементами: " << sum << endl;

}

else {

cout << "В массиве нет двух отрицательных чисел." << endl;

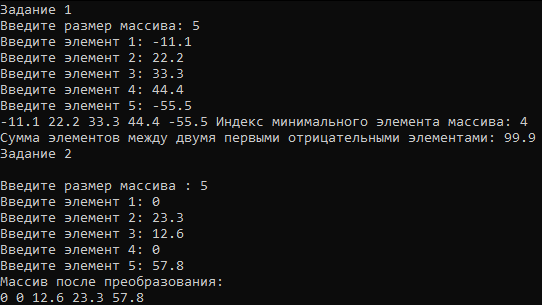
}

free(array);

second();

}

**Результат:**

****

**Лабораторная работа №16 на тему «Двумерные динамические массивы»**

**Задание:** В соответствии со своим вариантом написать программы на языке **С++** с использованием двумерных ***динамических*** массивов для условий задач из таблицы. Начальные размерности массивов ввести с клавиатуры.

**Условие:** В целочисленной прямоугольной матрице определить количество столбцов, не содержащих ни одного нулевого элемента.

**Решение:**

#include <iostream>

#include <ctime>

using namespace std;

int main()

{

setlocale(0, "");

srand(time(0));

int rows, cols;

cout << "Введите количество строк: ";

cin >> rows;

cout << "Введите количество столбцов: ";

cin >> cols;

int\*\* array = new int\* [rows];

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

array[i] = new int[cols];

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

{

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

{

array[i][j] = rand() % 100;

cout << array[i][j] << ' ';

}

cout << endl;

}

int nonZeroCols = 0;

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

{

bool isNonZero = false;

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

{

if (array[i][j] != 0)

{

isNonZero = true;

break;

}

}

if (isNonZero)

nonZeroCols++;

}

cout << "Количество ненулевых столбцов: " << nonZeroCols << endl;

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

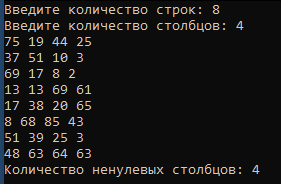
delete[] array[i];

delete[] array;

return 0;

}

**Результат:**

****

**Лабораторная работа №17 на тему «Функции пользователя»**

**Задание:** В соответствии со своим вариантом написать программы с использованием ***динамических*** массивов и ***функций* *пользователя*** для условий задач из таблицы.

Объединить написанные программы, разработав интерфейс с помощью оператора **switch**. Начальные размерности и значения элементов массивов ввести с клавиатуры в главной функции.

Массивы передать из главной функции в функции пользователя как ***параметры***.

**Условие:** 1. Ввести целочисленный массив **A**[**n**, **n**] и вычесть из каждой строки предыдущую. Из первой строки вычесть последнюю строку. Получившийся массив вывести.

2. Дана строка. Слова в предложении разделены одним или несколькими пробелами. Слова могут состоять только из цифр или букв. Найти сумму чисел, входящих в строку.

**Решение:**

#include <iostream>

#include <string>

using namespace std;

int stringsearch(int size,string array) {

int sum = 0;

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

if (array[i] >= '0' && array[i] <= '9') sum += array[i] - '0';

}

return sum;

}

void subtractRows(int\*\* A, int n) {

int\*\* B = new int\* [n];

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

B[i] = new int[n];

}

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

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

if (i == 0)

B[i][j] = A[i][j] - A[n - 1][j];

else

B[i][j] = A[i][j] - A[i - 1][j];

}

}

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

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

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

}

cout << endl;

}

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

delete[] B[i];

delete[] B;

}

int main() {

setlocale(0, "");

int x;

int\*\* A;

int n;

cout << "Выберите операцию (1 - Вычитание строк массива, 2 - Сумма чисел строки)" << endl;

cin >> x;

switch (x) {

case 1:

cout << "Введите размер массива: ";

cin >> n;

A = new int\* [n];

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

A[i] = new int[n];

}

cout << "Введите элементы массива:\n";

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

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

cin >> A[i][j];

}

}

subtractRows(A, n);

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

delete[] A[i];

delete[] A;

return 0;

break;

case 2:

int size;

string array;

cout << "Введите строку" << endl;

cin >> array;

getline(cin, array);

size = array.length();

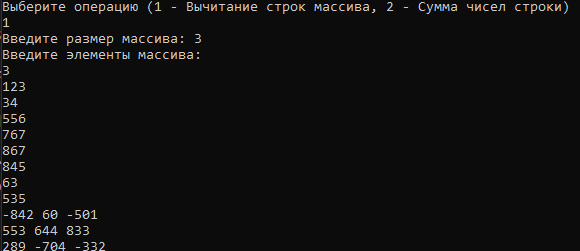
cout << "Сумма цифр в строке: " << stringsearch(size,array)<<endl;

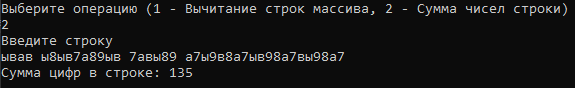
break;

return 0;

}

}

Результат: 



**Лабораторная работа №18 на тему «Указатели и ссылки при работе с функциями пользователя»**

**Задание:** В соответствии со своим вариантом написать главную функцию, в которой имеются вызовы ***функций пользователя***, реализующих задачи из таблицы ниже. Ввод исходных данных и вывод результатов осуществить в главной функции, при этом использовать***динамические* *массивы***.

Для передачи параметров в функции пользователя и возвращения результатов применить ***указатели*** и ***ссылки***.

**Условие:** 1. Создать одномерный массив, содержащий 15 элементов, наполнить его случайными значениями в интервале от 1 до 200. Определить сумму всех нечетных элементов массива.

2. Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные.

**Решение:** #include <iostream>

#include <ctime>

using namespace std;

int oddsum(int\* array,int size) {

int sum = 0;

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

if (array[i] % 2 == 1) sum += array[i];

}

return sum;

}

int searchrow(int\*\* A, int rows, int cols) {

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

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

if (A[i][j] > 0) {

return i;

}

}

}

return -1;

}

void switchsigns(int\*\* A, int rows, int cols, int number) {

if (number > 0) {

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

A[number - 1][j] \*= -1;

}

}

}

int main(){

setlocale(LC\_ALL, "rus");

srand(time(NULL));

int size;

cout << "Введите размер массива" << endl;

cin >> size;

int\* array = new int[size];

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

array[i] = rand() % 200;

cout << array[i] << ' ';

}

cout << "\nСумма нечетных элементов массива: " << oddsum(array, size) << endl;

delete[] array;

int rows;

int cols;

int\*\* A;

cout << "Введите количество строк: ";

cin >> rows;

cout << "Введите количество столбцов: ";

cin >> cols;

A = new int\*[rows];

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

A[i] = new int[cols];

}

cout << "Введите элементы массива" << endl;

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

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

cin >> A[i][j];

}

}

cout << "Введенный массив" << endl;

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

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

cout << A[i][j] << ' ';

}

cout << "\n";

}

int number = searchrow(A, rows, cols);

if (number < 0) cout << "В матрице нет положительных элементов" << endl;

cout << "Номер строки, содержащий в себе хотя бы один положительный элемент: " << number+1 << endl;

cout << "Измененный массив: " << endl;

switchsigns(A, rows, cols,number);

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

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

cout << A[i][j] << ' ';

}

cout << "\n";

}

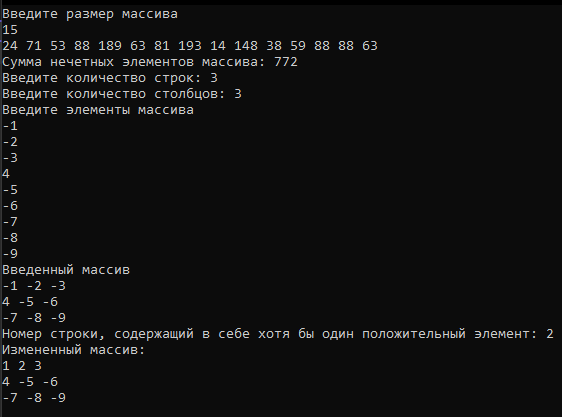
for (int i = 0; i < rows; ++i)

delete[] A[i];

delete[] A;

return 0;

}

**Результат:** 

**Лабораторная работа №19 на тему «Указатели на функции»**

**Задание 1:**

В соответствии со своим вариантом ***отделить корни*** двух уравнений (т. е. для каждого уравнения найти отрезок, внутри которого имеется один корень) и вычислить корни методом ***дихотомии*** для исходных данных из таблицы, приведенной ниже. Точность вычислений принять равной **e** = 0,001 для всех вариантов.

Операторы метода вычисления корня оформить в виде ***функции пользователя***, уравнения записать также в виде ***функций пользователя***.

В главной функции предусмотреть ввод исходных данных, обращения к функции, реализующей метод дихотомии для двух уравнений. В процессе выполнения программы определить корни двух уравнений. Использовать ***указатель на функцию****.*

**Условие 1: Функции:** x3 + 2x – 1,

ex − 2

**Задание 2:** В соответствии со своим вариантом написать программы по условиям задач из таблицы ниже. Программа должна содержать функцию пользователя с ***переменным числом параметров*** и не менее трех обращений к ней с различным количеством параметров.

**Условие 2:** Написать функцию **mn** с переменным числом параметров, которая находит максимальное из чисел типа **int**.

**Решение:**

#include <iostream>

#include <cstdarg>

using namespace std;

int mn(int n, ...) {

va\_list args;

va\_start(args, n);

int max = va\_arg(args, int);

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

int num = va\_arg(args, int);

if (num > max) max = num;

}

va\_end(args);

return max;

}

double f1(double x) {

return pow(x, 3) + 2 \* x - 1;

}

double f2(double x) {

return exp(x) - 2;

}

double dichotomy(double (\*f)(double), double a, double b, double e) {

double c;

while ((b - a) / 2 > e) {

c = (a + b) / 2;

if ((\*f)(c) == 0.0)

break;

else if ((\*f)(a) \* (\*f)(c) < 0)

b = c;

else

a = c;

}

return (a + b) / 2;

}

int main() {

setlocale(0, "");

// 2 Задание:

cout << "Максимальное число: " << mn(5, 3, 7, 9, 0, 2) << endl;

// 1 Задание:

double a1, b1, a2, b2;

cout << "Введите интервал для первого уравнения (a1, b1): ";

cin >> a1 >> b1;

cout << "Введите интервал для второго уравнения (a2, b2): ";

cin >> a2 >> b2;

double root1 = dichotomy(f1, a1, b1, 0.001);

double root2 = dichotomy(f2, a2, b2, 0.001);

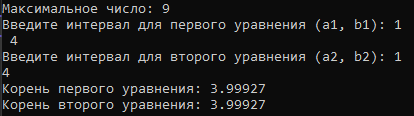
cout << "Корень первого уравнения: " << root1 << endl;

cout << "Корень второго уравнения: " << root2 << endl;

return 0;

}

**Результат:**

****