Министерство образования РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

**Контрольная работа**

**по Информатике за 1 семестр**

**Вариант 9**

Работу выполнил:

Обучающийся группы РИС-20-1бзу

Канзепаров Рушан Разинович

Проверила Доцент кафедры ИТАС:

Полякова О.А

Пермь, 2020

**Лабораторная работа №1**

"Знакомство с Си++. Выполнение программы простой структуры"

1. Вычислить значение выражения при различных вещественных типах данных (float и double). Вычисления следует выполнять с использованием промежуточных переменных. Сравнить и объяснить полученные результаты.
2. Вычислить значения выражений. Объяснить полученные результаты.

Вариант 9.

1. ,

при а=100, b=0.001

Начало

коец

cout

c1 = pow(a, 4);

c2 = pow(b, 4);

c3 = pow(a, 2);

c4 = pow(b, 2);

c5 = pow(b, 3);

cout << (c1 + c2) - (c1 + 4 \* c1 \* b) / 6 \* c3 \* c4 + 4 \* a \* c5 + c2;

b = 0.001, c1, c2, c3, c4, c5, a = 100

Float

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

float b = 0.001, c1, c2, c3, c4, c5, a = 100;

c1 = pow(a, 4);

c2 = pow(b, 4);

c3 = pow(a, 2);

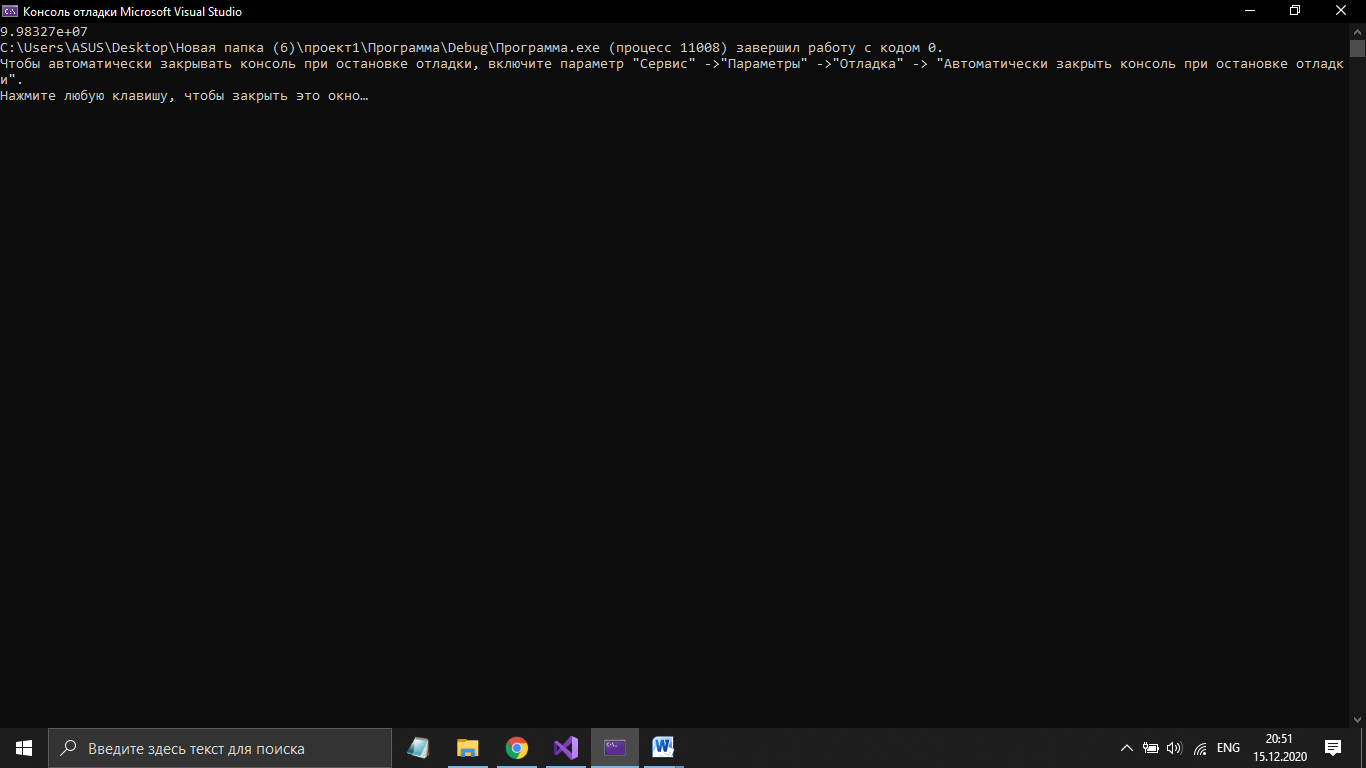
c4 = pow(b, 2);

c5 = pow(b, 3);

cout << (c1 + c2) - (c1 + 4 \* c1 \* b) / 6 \* c3 \* c4 + 4 \* a \* c5 + c2;

return 0;

}



Double

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

double b = 0.001, c1, c2, c3, c4, c5, a = 100;

c1 = pow(a, 4);

c2 = pow(b, 4);

c3 = pow(a, 2);

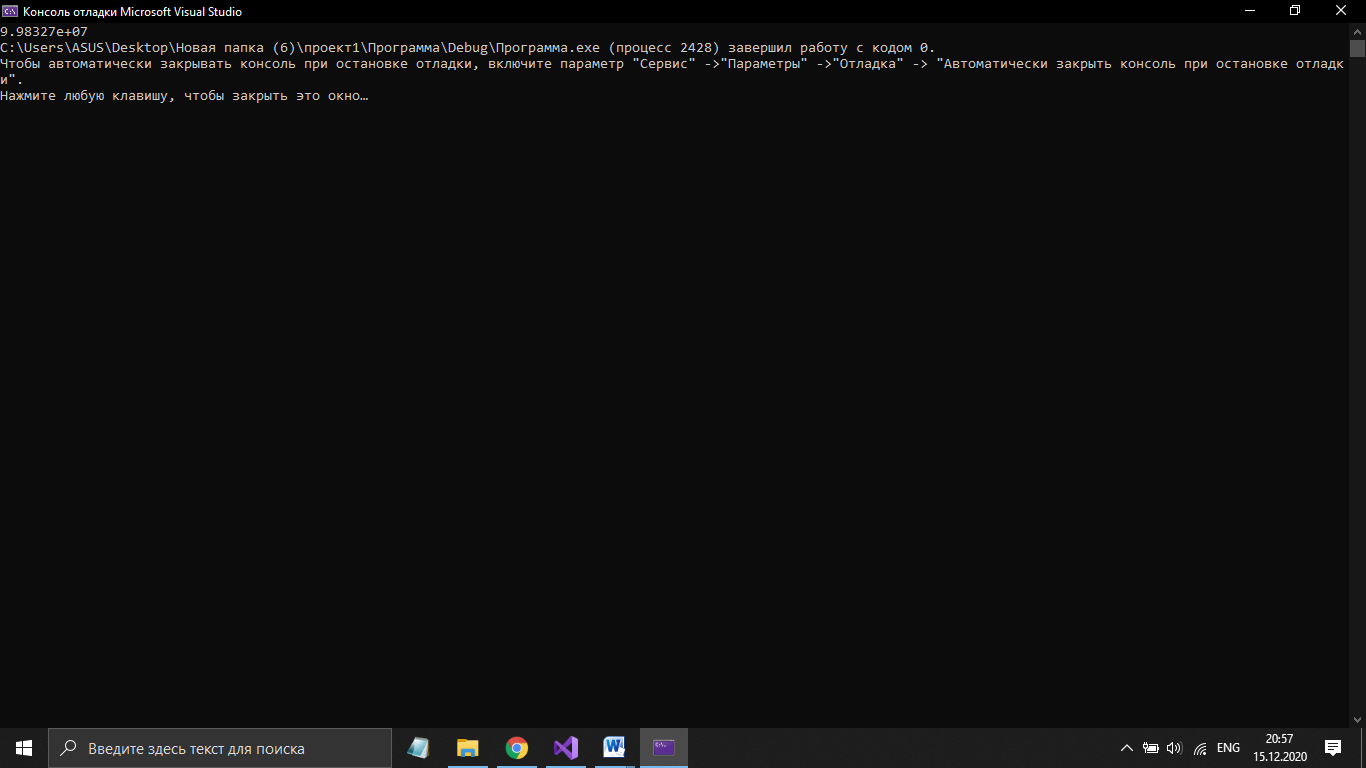
c4 = pow(b, 2);

c5 = pow(b, 3);

cout << (c1 + c2) - (c1 + 4 \* c1 \* b) / 6 \* c3 \* c4 + 4 \* a \* c5 + c2;

return 0;

}



2. 1)++n\*++m

Начало

int n=1,m=1 ;

cout << ++n\*++m;

return 0;

коец

cout

N=1,M=1

#include<iostream>

#include<math.h>

using namespace std;

int main()

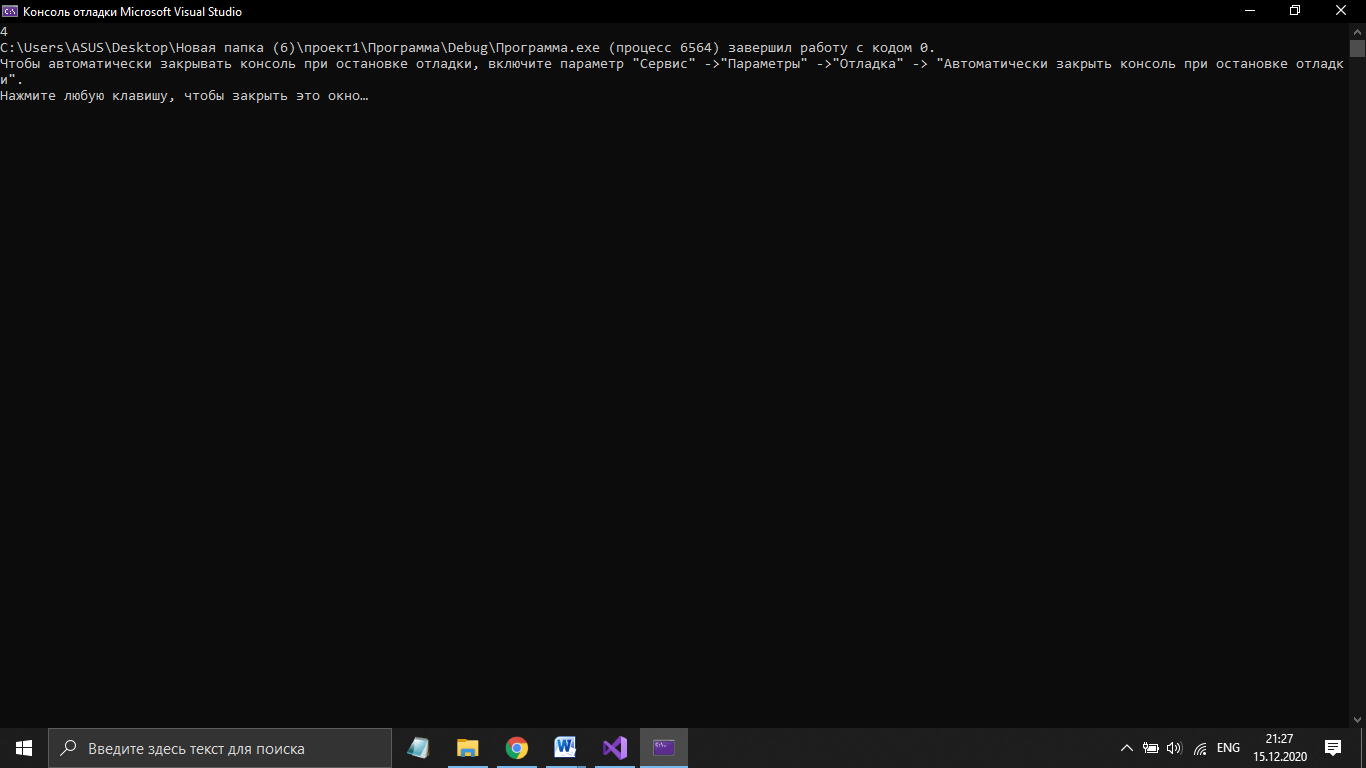
{

int n=1,m=1 ;

cout << ++n\*++m;

return 0;

}



2) m++<n

N=1,M=1

m++<n;

cout

коец

Начало

Cin n,m

#include <iostream>

#include <Windows.h>

int main()

{

float n, m;

setlocale(0, "");

std::cout << "\n Введите n: ";

std::cin >> n;

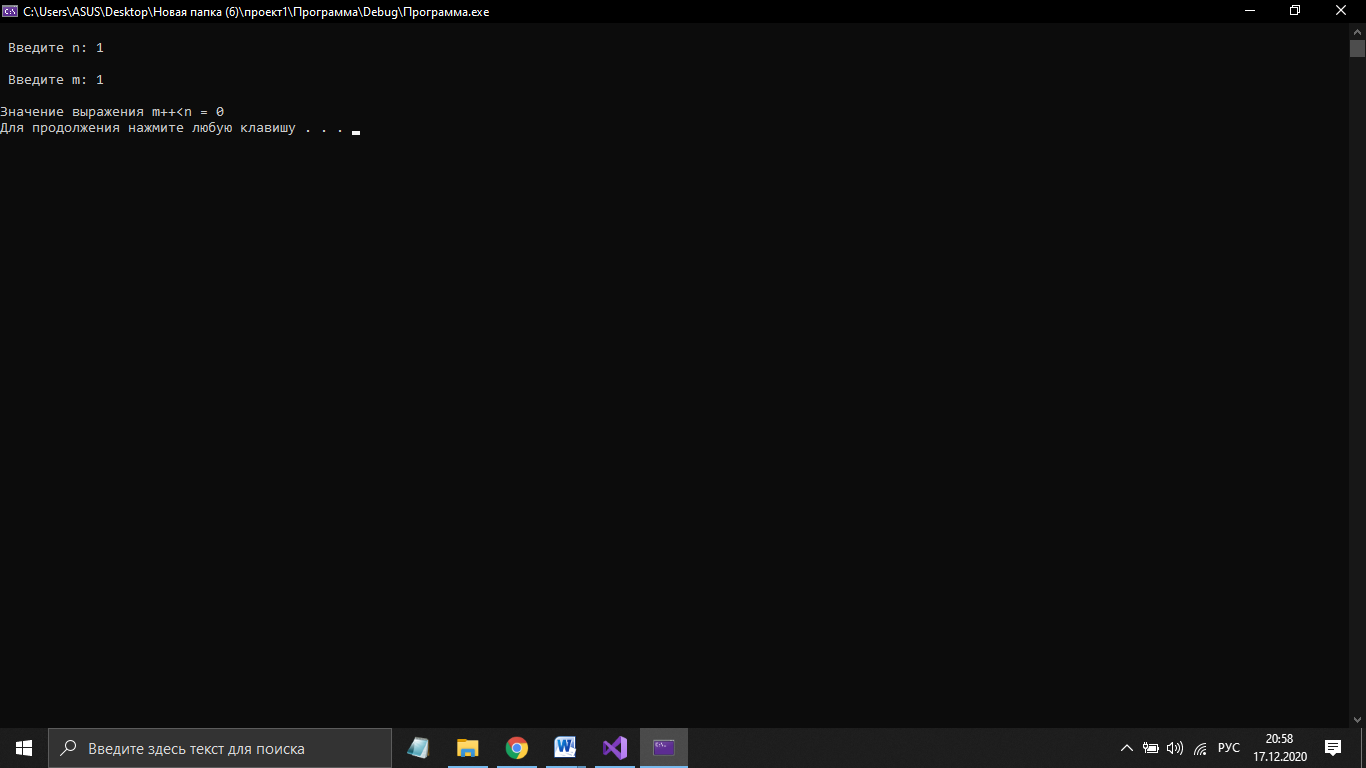
std::cout << "\n Введите m: ";

std::cin >> m;

std::cout << "\nЗначение выражения m++<n = " << (m++ < n);

std::cout << "\n";

system("PAUSE");

}

3) n++>m

N=1,M=1

m++>n;

cout

коец

Начало

Cin n,m

#include <iostream>

#include <Windows.h>

int main()

{

float n, m;

setlocale(0, "");

std::cout << "\n Введите n: ";

std::cin >> n;

std::cout << "\n Введите m: ";

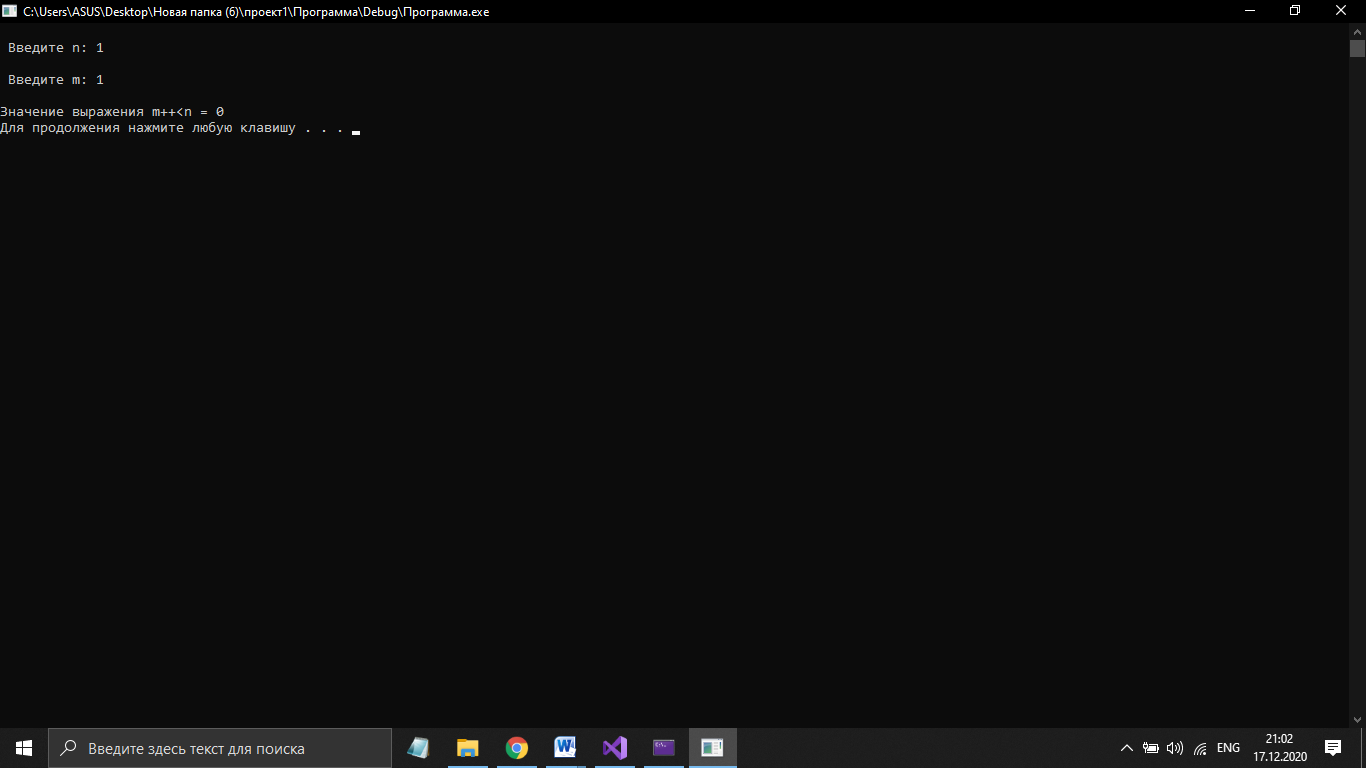
std::cin >> m;

std::cout << "\nЗначение выражения m++<n = " << (m++ > n);

std::cout << "\n";

system("PAUSE");

}



**Лабораторная работа №2**

# "Использование основных операторов языка Си"

Используя оператор цикла, найти сумму элементов, указанных в конкретном варианте. Результат напечатать, снабдив соответствующим заголовком.

Вариант 9.

Найти сумму ряда с точностью ε=10-4, общий член которого

.

#include <cstdlib>

#include <iostream>

#include <math.h>

#define E 10-4 //точность

int main(int argc, char\* argv[]) {

int n = 1;

double s = 0, a = 1;

do {

a \*= 10^n / (n + 1); //расчет следующего числа ряда по рекуррентной формуле

s += a; n++;

}

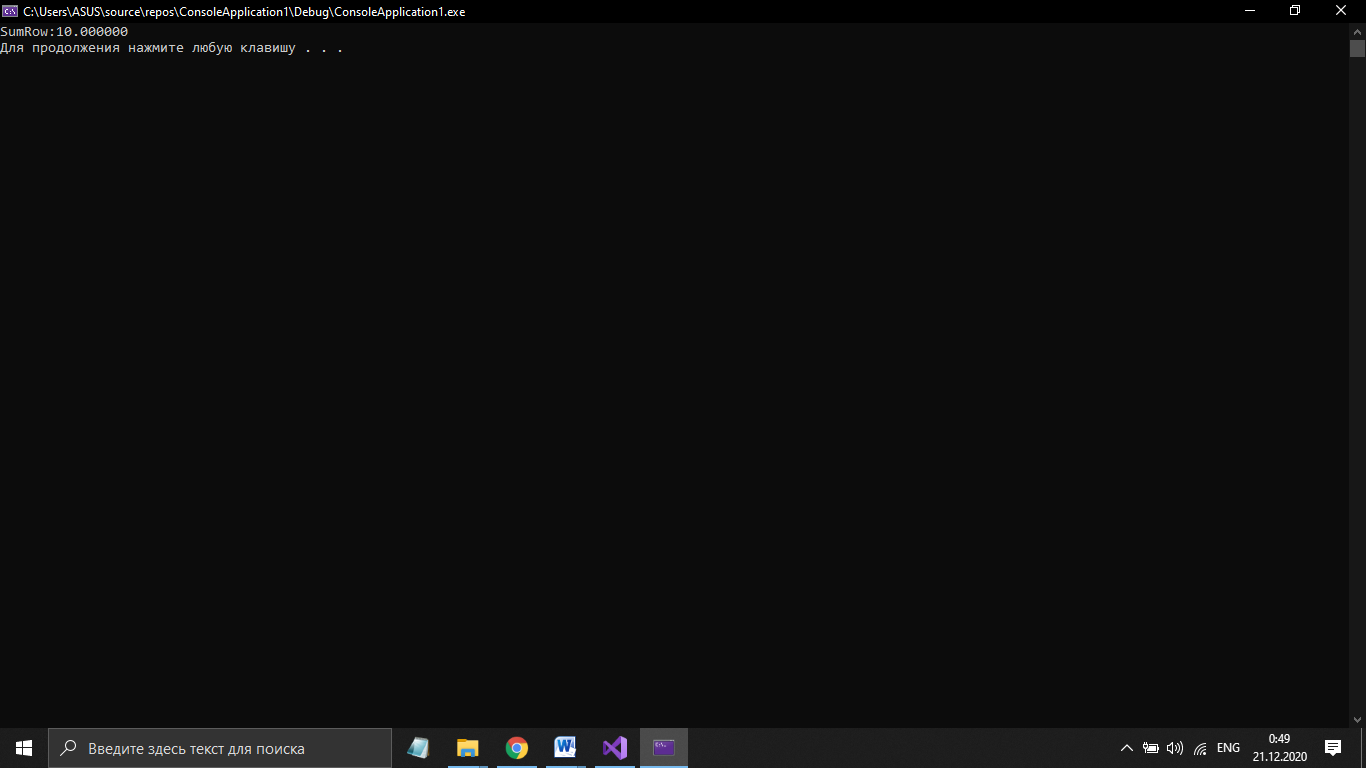
while (a < E); //выход из цикла при достижении нужной точности

printf("SumRow:%f\n", s);

system("PAUSE");

return EXIT\_SUCCESS;

}



# **Лабораторная работа №3** "Вычисление функций с использованием их разложения в степенной ряд"

Для х изменяющегося от a до b с шагом (b-a)/k, где (k=10), вычислить функцию f(x), используя ее разложение в степенной ряд в двух случаях:

а) для заданного n;

б) для заданной точности ε (ε=0.0001).

Для сравнения найти точное значение функции.

Вариант 9.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | 3 |  |

#include <cstdlib>

#include <iostream>

#include <math.h>

#define A 0.1

#define B 0.8

#define E 10e-4

int main(int argc, char\* argv[]) {

float x, k = (B - A) / 10.0, sn = 0, se = 0, t, y;

int i, j, n = 3;

for (x = A; x <= B; x += k) {

sn = x; se = x;

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

sn += pow(x, 4 \* i + 1) / (4 \* i + 1);

};

j = 1;

do {

t = pow(x, 4 \* j + 1) / (4 \* j + 1);

j++; se += t;

}

while (t > E);

y = log((1.0 + x) / (1.0 - x)) / 4.0 + atan(x) / 2.0;

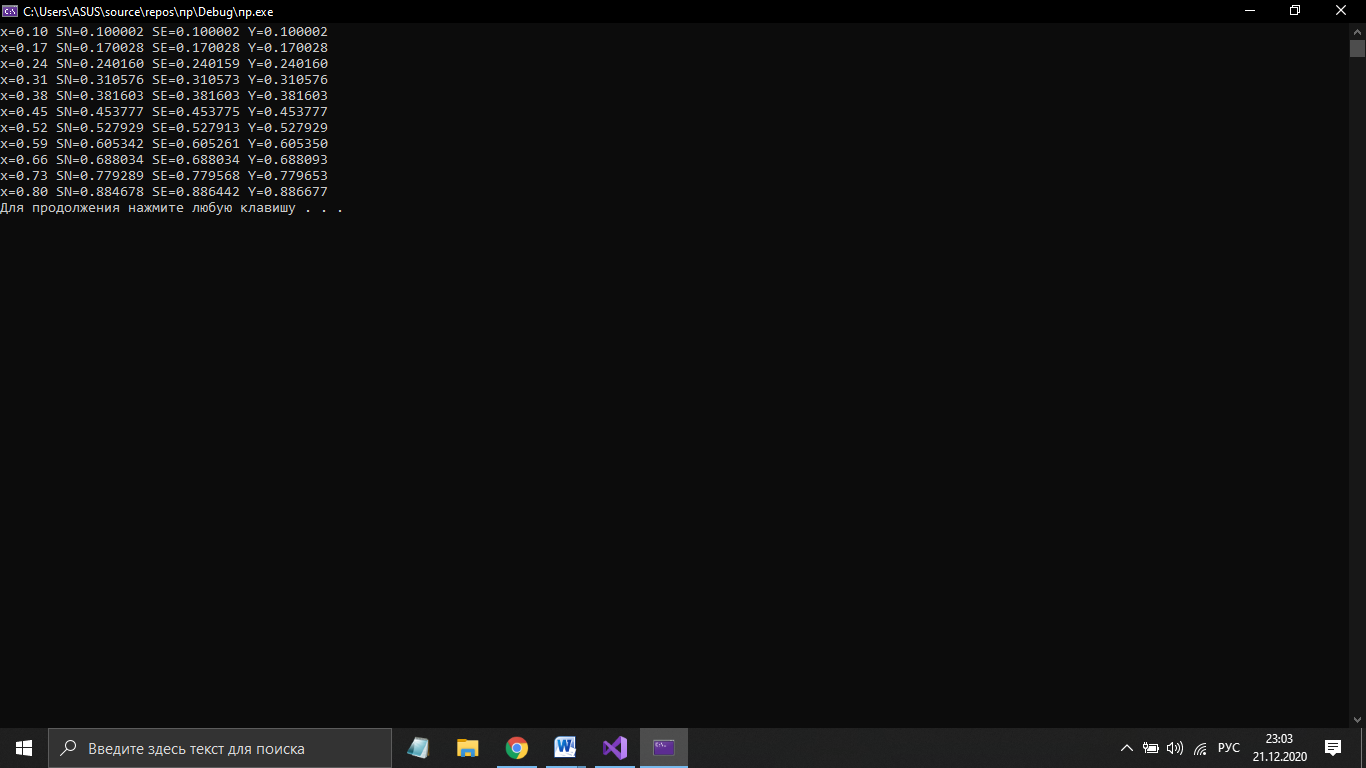
printf("x=%.2f SN=%f SE=%f Y=%f\n", x, sn, se, y);

}

system("PAUSE");

return EXIT\_SUCCESS;

}



**Лабораторная работа №4**

"Работа с одномерными массивами"

Вариант 9.

1. Сформировать одномерный массив целых чисел, используя датчик случайных чисел.
2. Распечатать полученный массив.
3. Удалить первый элемент с заданным значением.
4. Сдвинуть массив циклически на К элементов вправо.
5. Распечатать полученный массив.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <cstdlib>

#include <iostream>

#include <time.h>

int main(int argc, char\* argv[]) {

int i, j, x, k, size;

srand(time(NULL));

int a[100];

int b[100];

printf("Enter size:>");

scanf("%d", &size);

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

a[i] = rand() % 100; printf("%d ", a[i]);

}

printf("\nEnter value to destroy:>"); scanf("%d", &x);

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

if (a[i] == x) {

for (j = i; j < size - 1; j++) {

a[j] = a[j + 1];

}

size--; break;

}

}

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

printf("%d ", a[i]);

}

printf("\nEnter K:>"); scanf("%d", &k);

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

if ((i + k) < size) { b[i + k] = a[i]; }

else { b[i + k - size] = a[i]; }

}

printf("Result:\n");

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

printf("%d ", b[i]);

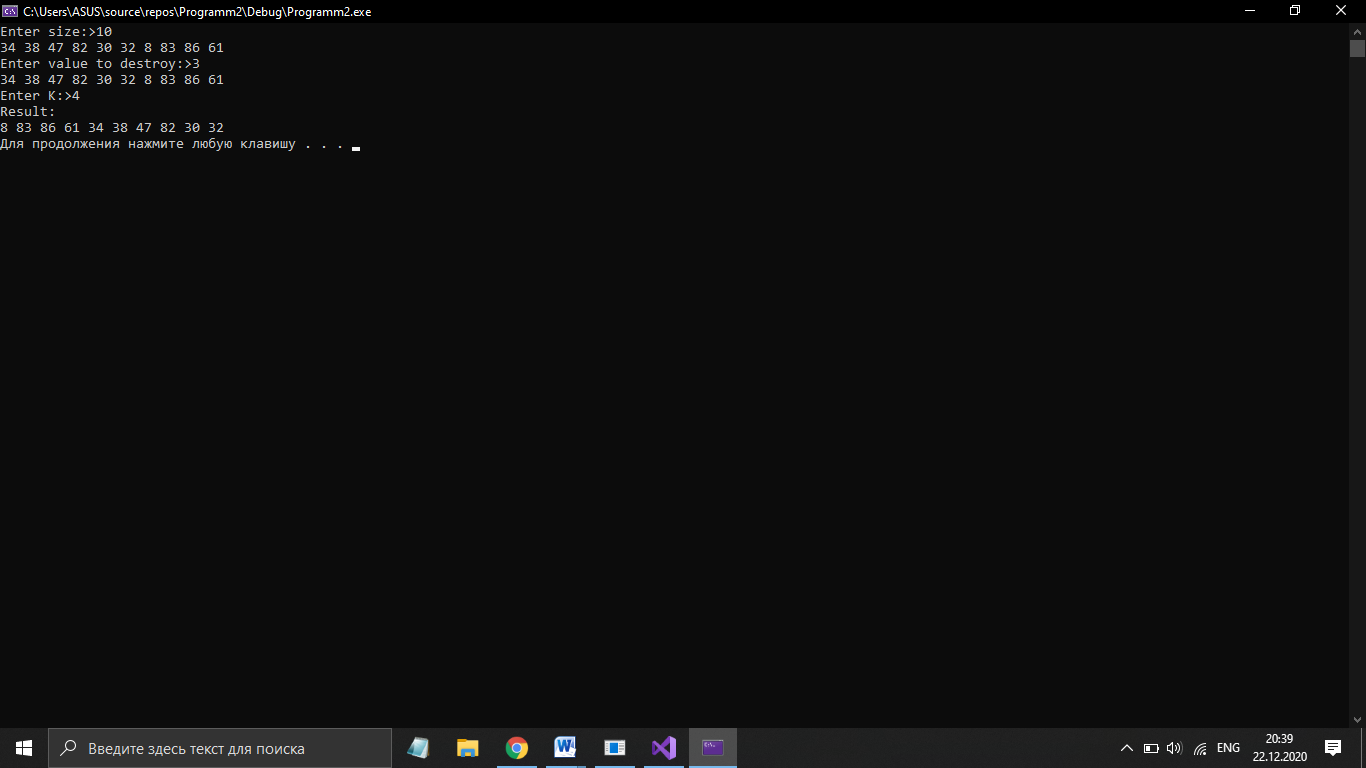
}

printf("\n");

system("PAUSE");

return EXIT\_SUCCESS;

}



**Лабораторная работа №5**

"Функции и массивы"

Используя функции, решить указанную в варианте задачу. Массив должен передаваться в функцию как параметр.

Вариант 9.

Написать функцию для вычисления суммы элементов квадратной матрицы, которые расположены ниже главной диагонали. С ее помощью найти максимальное значение такой суммы в n матрицах.

#include <iostream>

#include <time.h>

#include <math.h>

#define SIZE 5

tatic int a[SIZE][SIZE];

void CreateArray(int a[SIZE][SIZE]) {

int i, j;

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

for (j = 0; j < SIZE; j++) {

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

}

}

}

void PrintArray(int a[SIZE][SIZE]) {

int i, j;

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

for (j = 0; j < SIZE; j++) {

printf("%d ", a[i][j]);

}

printf("\n");

}

}

int MatrixSum1(int m[SIZE][SIZE]) {

int i, j, s = 0;

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

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

s += m[i][j];

}

}

return s;

}

int main(int argc, char\* argv[]) {

int s = 0, i, t, N;

N = atoi(argv[1]);

srand(time(NULL));

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

CreateArray(a);

PrintArray(a);

t = MatrixSum1(a);

printf("\nSumm:%d\n ", t);

if (t > s) { s = t; }

}

printf("\nMaximum %d\n", s);

system("PAUSE");

return EXIT\_SUCCESS;

}

**Лабораторная работа №6**

"Строки"

Задана строка, состоящая из символов. Символы объединяются в слова. Слова друг от друга отделяются одним или несколькими пробелами. В конце текста ставится точка. Текст содержит не более 255 символов. Выполнить ввод строки, используя функцию Gets(s) и обработку строки в соответствии со своим вариантом.

Вариант 9.

Преобразовать строку так, чтобы все слова в ней стали идентификаторами, слова состоящие только из цифр - удалить.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <cstdlib>

#include <iostream>

#include <string.h>

void ClearString(char string[255]) {

int i;

for (i = 0; i < 255; i++) { string[i] = '\0'; }

}

int main(int argc, char\* argv[]) {

char str[255], s[255] = "", t[255] = "", c; bool r;

int i = 0, j = 0;

printf("Enter text>"); gets\_s(s);

strcpy(str, "");

do {

r = false;

do {

c = s[i]; i++;

if (c > '9') { r = true; }

if (c == ' ' || c == '.' || c == '\0') {

if (t[0] > '9') {

if (r) {

strcat(str, t); strcat(str, " ");

}

} t[j] = '\0';

j = 0; ClearString(t); break;

}

else { t[j] = c; j++; }

} while (1);

} while (c != '.' && c != '\0');

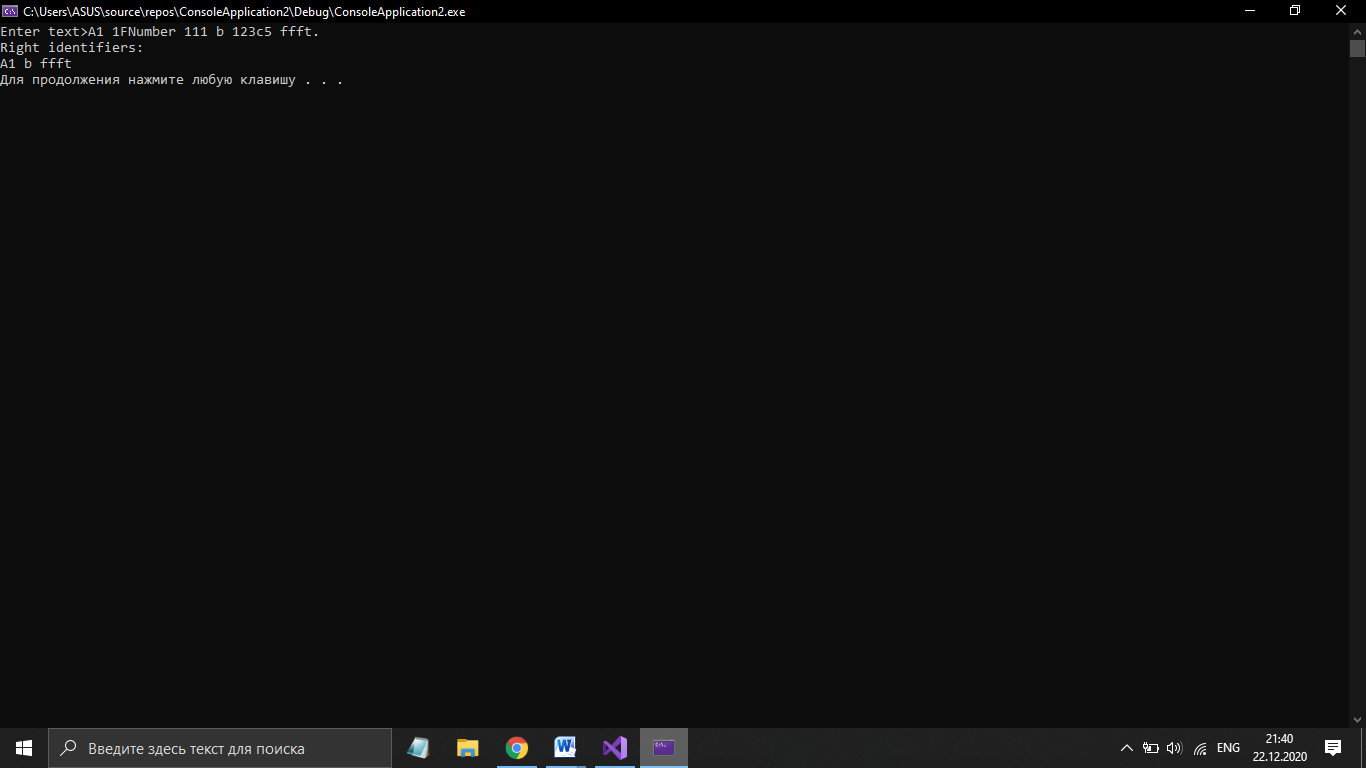
printf("Right identifiers:\n");

puts(str);

system("PAUSE");

return EXIT\_SUCCESS;

}



**Лабораторная работа №7**

# "Функции в С++"

Вариант 9.

1. Написать функцию с продемонстрировать различные способы вызова функции умалчиваемыми параметрами в соответствии с вариантом,:

* с параметрами заданными явно,
* с опущенными параметрами
* часть параметров задана явно, а часть опущена.

2. Написать функцию с переменным числом параметров в соответствии с вариантом, продемонстрировать вызов функции с различным числом параметров.

3. Написать перегруженные функции в соответствии с вариантом. Написать демонстрационную программу для вызова этих функций.

4. Написать шаблон функций вместо перегруженных функций из задания 3. Написать демонстрационную программу для вызова этих функций. списка параметров

5. Решить уравнение указанным в варианте методом. Уравнение передать в функцию как параметр с помощью указателя.

|  |  |  |  |
| --- | --- | --- | --- |
| Печать координат точки | Среднее арифметическое из элементов в списке параметров, стоящих на четных местах | Поиск заданного элемента в отсортированном массиве | Метод половинного деления  t2  Отрезок, содержащий корень: [0;0,85]  Точное значение: 0,2624 |

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include<Windows.h>

#include <stdio.h>

#include <string.h>

#include <stdarg.h>

#include <ctime>

#include <math.h>

#include <conio.h>

#include <algorithm>

using namespace std;

const int l\_x = 10, l\_y = 10;

//Объявление структуры

struct Cordinata {

char x[l\_x + 1];

char y[l\_y + 1];

int r\_;

};

//Объявление функции с умалчиваемыми параметрами для печати структуры

void print(const char\* x = "x", const char\* y = "y", int r = 1) { cout << "\n" << x << ' ' << y << ' ' << r; }

//Объявление функции с переменным числом параметров для определения среднего арифметического элементов

float sr\_ar(int a, ...);

//Объявление перегруженных функций типа int, double, char для нахождения минимального элемента

int min\_(const int\* mas, const int n);

double min\_(const double\* mas, const int n);

char\* min\_(char\*\* mas, const int n);

//Объявление шаблона функций для вычисления минимального элемента

template <class T> T Min(T\* b, int n);

//Тип-указатель на функцию уравнения

typedef float(\*fptr)(float);

//Функция, реализующая метод итераций

float root(fptr f, float e);

//Функция находит корень уравнения

float testf(float x);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(0));

Cordinata Cordinata;

char buf[3];

char x[l\_x + 1];

char y[l\_y + 1];

cout << "Задание 1" << endl;

cout << endl;

puts("Введите x: "); gets\_s(x);

if (strlen(x) < l\_x)

for (int i = strlen(x); i < l\_x; i++)

x[i] = ' ';

x[l\_x] = 0;

//Копирование введённого значения в структуру

strncpy(Cordinata.x, x, l\_x + 1);

puts("Введите y: "); gets\_s(y);

if (strlen(y) < l\_y)

for (int i = strlen(y); i < l\_y; i++)

y[i] = ' ';

y[l\_y] = 0;

strncpy(Cordinata.y, y, l\_y + 1)

cout << "Результаты вызова функции с умалчиваемыми параметрами" << endl;

//Вызов функции с параметрами заданными явно

print(Cordinata.x, Cordinata.y, Cordinata.r\_);

//Вызов функции с опущенными параметрами

print();

//Вызов функции, часть параметров задана явно, а часть опущена

print(Cordinata.x, Cordinata.y);

cout << endl;

cout << endl;

cout << "Задание 2" << endl;

cout << endl;

srand(time(NULL));

int\* arr = NULL;

int n, cnt = 0;

double aver = 0.;

cout << "Размер Массива = "; cin >> n;

arr = new int[n];

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

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

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

if (!(i % 2) && !(arr[i] % 2)) {

cnt++;

aver += arr[i];

}

}

if (cnt) cout << endl << "Среднее значение = " << aver / cnt;

else cout << endl << "Не совподает";

delete[] arr;

cout << endl;

int i, m;

cout << "Задание 3" << endl;

cout << endl;

int ar[10]; // создали массив на 10 элементов

int key; // создали переменную в которой будет находиться ключ

cout << "Введите 10 чисел для заполнения массива: " << endl;

for (int i = 0; i < 10; i++) {

cin >> ar[i]; // считываем элементы массива

}

sort(ar, ar + 10); // сортируем с помощью функции sort (быстрая сортировка)

cout << endl << "Введите ключ: ";

cin >> key; // считываем ключ

bool flag = false;

int l = 0; // левая граница

int r = 9; // правая граница

int mid;

while ((l <= r) && (flag != true)) {

mid = (l + r) / 2; // считываем срединный индекс отрезка [l,r]

if (ar[mid] == key) flag = true; //проверяем ключ со серединным элементом

if (ar[mid] > key) r = mid - 1; // проверяем, какую часть нужно отбросить

else l = mid + 1;

}

if (flag) cout << "Индекс элемента " << key << " в массиве равен: " << mid;

else cout << "Извините, но такого элемента в массиве нет" << endl;

cout << endl;

cout << "Задание 4" << endl;

cout << endl;

//В функцию root передается указатель на функцию и точность

float res = root(testf, 2.624);

cout << "X = " << res << endl;

return 0;

}

//Вычисление среднего арифметического элементов

float sr\_ar(int a, ...) {

va\_list arg;

va\_start(arg, a);

float sum = 0, n = 0;

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

int x = va\_arg(arg, int);

sum += x;

n++;

}

float sr = sum / n;

va\_end(arg);

return sr;

}

//Нахождение минимального элемента int

int min\_(const int\* mas, const int n) {

cout << "Массив чисел типа int : " << endl;

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

cout << "mas[" << i << "]=" << mas[i];

cout << endl;

}

int min = mas[0];

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

if (mas[i] < min) min = mas[i];

}

cout << endl;

return min;

}

//Нахождение минимального элемента double

double min\_(const double\* mas, const int n) {

cout << "Массив чисел типа double : " << endl;

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

cout << "mas[" << i << "]=" << mas[i];

cout << endl;

}

double min = mas[0];

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

if (mas[i] < min) min = mas[i];

}

cout << endl;

return min;

}

//Нахождение минимального элемента char

char\* min\_(char\*\* mas, const int n) {

cout << "Массив чисел типа char : " << endl;

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

cout << "mas[" << i << "]=" << mas[i];

cout << endl;

}

char\* min1 = new char[n];

int min = strlen(mas[0]);

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

if (strlen(mas[i]) < min) {

min = strlen(mas[i]);

min1 = mas[i];

}

else if (strlen(mas[i]) == min) {

min = strlen(mas[i]);

min1 = mas[i];

}

}

cout << endl;

return min1;

}

//Шаблон функции нахождение минимального элемента

template <class T> T Min(T\* b, int n) {

cout << "Массив чисел : " << endl;

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

cout << "mas[" << i << "]=" << b[i];

cout << endl;

}

double min = b[0];

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

if (b[i] < min) min = b[i];

}

cout << endl;

return min;

}

//Реализация метода итераций

float root(fptr f, float e)

{

float x; float res;

float x0 = 0, x1;

do

{

x1 = x0;

//Присваиваем значение найденное в функции

x0 = (\*f)(x1);

} while (fabs(x0 - x1) > e);

return x0;

}

//Уравнение для вычисления корня

float testf(float x)

{

return x - (1/ (3 + sin(3.6) \* x);

}

