Федеральное государственное образовательное бюджетное

учреждение высшего образования   
«**Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

**Колледж информатики и программирования**

**ОДП.01 «Операционные системы»**

**Индивидуальная работа №6**

Преподаватель: Сибирев И.В.

Выполнил: Гуцалов

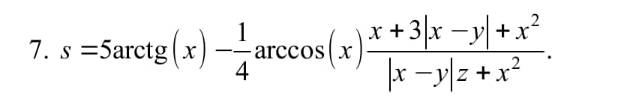
Группа: 2 ОИЬАС - 1322

Москва 2023

ЛАБОРАТОРНАЯ РАБОТА №1

СРЕДА ПРОГРАММИРОВАНИЯ VISUAL C++.

ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ

:

#include <iostream>

using namespace std;

#include <math.h>

int main()

{

double x, y, z, a, b, c, s;

cout << "Vvedite x: ";

cin >> x;

cout << "Vvedite y: ";

cin >> y;

cout << "Vvedite z: ";

cin >> z;

a = 5\*atan(x)-(1/4) \* acos(x);

b = x + 3 \* fabs(x - y)+pow(x, 2);

c = fabs(x-y)\*z + pow(x, 2);

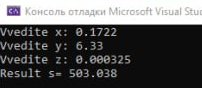
s = a \* (b/c);

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

return 0;

}

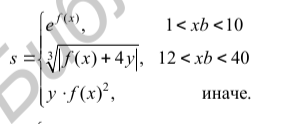
Результат:



ЛАБОРАТОРНАЯ РАБОТА №2

ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ

АЛГОРИТМОВ

Код: 

#include <iostream>

using namespace std;

#include <math.h>

int main()

{

double x;

double y=0;

double f;

double a;

double s;

int k;

cout << "Vvedite x "; cin >> x;

cout << "Vvedite y "; cin >> y;

cout << "Viberite f: 1 - sh(x), 2 - x^2, 3 – exp(x) "; cin >> k;

switch (k) {

case 1: f = sinh(x); break;

case 2: f = pow(x, 2); break;

case 3: f = exp(x); break;

default: cout << "Ne vuibrana funkciya "; return 1;

}

a = fabs(x \* y);

if (a < 5) {

cout << "Net rezultata" << endl;

return 1;

}

else

if (a > 10) s = fabs(f) + log(y);

else

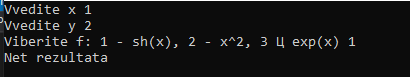
if (a <= 10 && a > 5) s = exp(f + y);

else s = sin(x) + tan(y);

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

return 0;

}



ЛАБОРАТОРНАЯ РАБОТА №3

ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ



Код:

#include <iostream>

#include <cmath>

int factorial(int n) {

if (n == 0) {

return 1;

}

else {

return n \* factorial(n - 1);

}

}

int main() {

double a = 0.1;

double b = 1;

double step = (6 - a) / 10;

int n = 120;

std::cout << "x\tY(x)\t\t\n";

for (double x = a; x <= b; x += step) {

double y = 1 + x \* x / 2 - std::atan(x) / 3 + (std::pow(x, 4 \* n + 1) / (2 \* factorial(2 \* n + 1)));

std::cout << x << "\t" << y << "\t";

double taylor\_series = 0;

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

taylor\_series += std::pow(-1, i) \* std::pow(x, 2 \* i + 1) / (2 \* i + 1);

}

std::cout << taylor\_series << std::endl;

}

return 0;

}

Результат:

