***Министерство образования Республики Беларусь***

***Учреждение образования***

***“Брестский государственный технический университет”***

***Кафедра ИИТ***

***Отчёт по лабораторной работе №4***

***По дисциплине ТиИИТ за 2 семестр***

**Выполнил:**

Студент группы ИИ-23(1)

1 курса

Макаревич Н. Р.

**Проверил:**

Анфилец С.В.

Брест 2023

Вариант 2

Цель работы: Найти минимум функции

Пользовательская библиотека:

#include "Func.h"

double alfArr[8][125];

void population() {

srand(time(NULL));

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 125; j++) {

alfArr[i][j] = ((double)rand() / RAND\_MAX) \* 9 + 1;

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

}

cout << endl << endl;

}

}

double func(double x, int a, double b) {

return cos(x) + 1 / b \* cos(a \* x + 1) + 1 / pow(b, 2) \* cos(pow(a, 2) \* x + 2) + 1 / pow(b, 3) \* cos(pow(a, 3) \* x + 3) + 1 / pow(b, 4) \* cos(pow(a, 4) \* x + 4);

}

double minOfFunc(int x0, int a, int b, int Nmax, int ind) {

cout << "Start of " << ind << endl;

cout << endl;

double \*temp = new double[125];

temp = alfArr[ind];

double y0 = func(x0, a, b);

double yMin, y1, x1, Y = 0;

double finalMinY = y0;

bool check = false;

for (int j = 0; j < 125; j++) {

for (int i = 0; temp[j] > 0.0001 && i < Nmax; i++, temp[j] \*= 0.75) {

x1 = x0 + temp[j] \* ((double)rand() / RAND\_MAX) \* 2 - 1;

y1 = func(x1, a, b);

if (y1 < y0) {

x0 = x1;

Y = y1;

check = true;

}

}

if (check) yMin = Y;

else yMin = y0;

if (yMin < finalMinY)

finalMinY = yMin;

}

cout << "End of " << ind << endl;

return finalMinY;

}

double threadsFunc(int x0, int a, int b, int Nmax) {

double res = 10e9;

population();

thread threads[8];

double threadRes[8];

/\*for (int i = 0; i < 8; i++) {

threads[i] = thread(minOfFunc, x0, a, b, Nmax);

}

for (int i = 0; i < 8; i++) {

threads[i].join();

}\*/

threads[0] = thread([&]() {

threadRes[0] = minOfFunc(x0, a, b, Nmax, 0);

});

threads[1] = thread([&]() {

threadRes[1] = minOfFunc(x0, a, b, Nmax, 1);

});

threads[2] = thread([&]() {

threadRes[2] = minOfFunc(x0, a, b, Nmax, 2);

});

threads[3] = thread([&]() {

threadRes[3] = minOfFunc(x0, a, b, Nmax, 3);

});

threads[4] = thread([&]() {

threadRes[4] = minOfFunc(x0, a, b, Nmax, 4);

});

threads[5] = thread([&]() {

threadRes[5] = minOfFunc(x0, a, b, Nmax, 5);

});

threads[6] = thread([&]() {

threadRes[6] = minOfFunc(x0, a, b, Nmax, 6);

});

threads[7] = thread([&]() {

threadRes[7] = minOfFunc(x0, a, b, Nmax, 7);

});

threads[0].join();

threads[1].join();

threads[2].join();

threads[3].join();

threads[4].join();

threads[5].join();

threads[6].join();

threads[7].join();

for (int i = 0; i < 8; i++) {

//cout << threadRes[i] << " ";

res = min(res, threadRes[i]);

}

return res;

}

Описание функций в файле Header.h:

#pragma once

#include <cmath>

#include <iostream>

#include <thread>

#include <time.h>

#include <stdlib.h>

#include <chrono>

using namespace std;

void population();

double func(double, int, double);

double minOfFunc(int, int, int, int);

double threadsFunc(int, int, int, int);

Код программы:

#include <iostream>

#include "Func.h"

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

cout << "Введите Х0 и параметр b(2,10): " << endl;

int a = 9; //Макаревич

double x0, b;

cin >> x0 >> b;

while (true) {

if (b >= 2 && b <= 10)

break;

else {

cout << "Параметр b выходит за рамки, введите заново." << endl;

cin >> b;

}

}

cout << "Введите максимальное кол-во итераций: " << endl;

int Nmax;

cin >> Nmax;

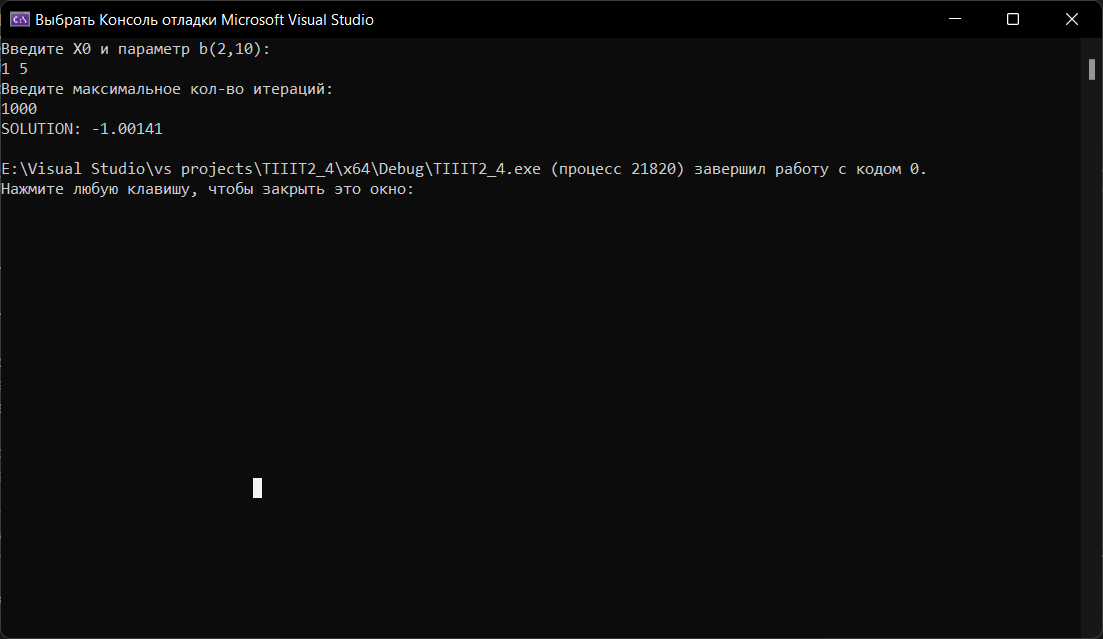
double res = threadsFunc(x0, a, b, Nmax);

cout << "SOLUTION: " << res << endl;

return 0;

}

Вывод программы:



Вывод: была решена задача методом случайного поиска. Программа написана на языке С++.