Міністерство освіти і науки України

НТУУ „КПІ”

ТЕФ

**Кафедра АПЕПС**

***Звіт***

***До лабораторної роботи №1***

***З дисципліни***

***"Об’єктно-орієнтоване програмування"***

Виконав: **Артамонов О.Ю.**

Студент групи: **ТВ-61**

Перевірив викладач: **Карпенко С.Г.**

**Київ – 2017**

**Лістинг**

#include "stdafx.h"

#include <stdio.h>

#include <cmath>

#include <conio.h>

#include <stdlib.h>

#include <time.h>

typedef struct

{

float a, b, dx, x1, x2;

}Value;

float \*fun;

int m;

float x, fx;

int okr(double value)

{

return (int)floor(value + 0.5);

}

void line(int znak\_left, int znak\_center, int znak\_right, int znak)

{

printf("%c", znak\_left);

for (int i = 1; i < 18; i++)

{

if (i % 9 == 0)

printf("%c", znak\_center);

else

printf("%c", znak);

}

printf("%c", znak\_right);

printf("\n");

}

void ReadPar(Value & val)

{

printf(" Enter a:\n ");

scanf\_s("%f", &val.a);

printf(" Enter b:\n ");

scanf\_s("%f", &val.b);

printf(" Enter x1:\n ");

scanf\_s("%f", &val.x1);

printf(" Enter x2:\n ");

scanf\_s("%f", &val.x2);

printf(" Enter dx:\n ");

scanf\_s("%f", &val.dx);

}

int m\_f(Value & val) {

m = /\*round\*/okr(abs((val.x2 - val.x1) / val.dx)) + 1;

return m;

}

float Tab(Value & val, int m)

{

fun = new float[m];

int steps = 0;

double mistake;

x = val.x1;

line(201, 209, 187, 205);

printf("%c x %c f(x) %c\n", 186, 179, 186);

for (int i = 0; i < m; i++, x += val.dx)

{

if (x <= -0.7)

{

fx = -val.a \* x \* x;

fun[i] = fx;

line(199, 197, 182, 196);

printf("%c %6.1f %c %6.2f %c\n", 186, x, 179, fx, 186);

}

if (x > -0.7)

{

if ((val.b\*x) != 0) {

fx = (val.a - x) / (val.b\*x);

fun[i] = fx;

line(199, 197, 182, 196);

printf("%c %6.1f %c %6.2f %c\n", 186, x, 179, fx, 186);

}

else { printf("ERROR! Division by ZERO!"); }

}

steps++;

}

line(211, 193, 189, 196);

if ((m - 1) != 0) {

mistake = ((val.x2 + 0.1 - val.x1) / (m - 1)) - val.dx;

}

else { printf("ERROR! Division by ZERO!"); }

printf("\n\n");

printf(" Steps:\n %d\n", steps);

printf("\n\n");

printf(" Mistake of dx:\n %.5lf\n", mistake);

printf("\n");

return \*fun;

}

void Rand(float \* fun, Value & val, int m) {

float s1, s2;

s1 = fun[1];

s2 = fun[0];

for (int i = 1; i < m; i += 2) {

if (s1 > fun[i]) {

s1 = fun[i];

}

}

printf(" S1:\n %.2f", s1);

printf("\n\n");

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

if (fun[i] < s2)

{

s2 = fun[i];

}

}

printf(" S2:\n %.2f", s2);

printf("\n\n");

float s\_max, s\_min;

if (s1 > s2) {

s\_max = s1;

s\_min = s2;

}

else {

s\_max = s2;

s\_min = s1;

}

double step = abs((s\_max + s\_min)\*0.03 / 2);

printf(" Step:\n %lf\n\n", step);

float \*rnd = new float[m];

printf("%c", 201);

for (int i = 0; i < 12; i++)

printf("%c", 205);

printf("%c\n", 187);

printf("%c Random %c\n", 186, 186);

printf("%c", 204);

for (int i = 0; i < 12; i++)

printf("%c", 205);

printf("%c\n", 185);

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

rnd[i] = s\_min + step \* (rand() % (int)((s\_max - s\_min) / step));

printf("%c %-4.3f %c\n", 186, rnd[i], 186);

printf("%c", 204);

for (int i = 0; i < 12; i++)

printf("%c", 205);

printf("%c\n", 185);

}

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

rnd[i] = s\_min + step\*(rand() % (int)((s\_max - s\_min) / step));

printf("%c %-4.3f %c\n", 186, rnd[i], 186);

}printf("%c", 200);

for (int i = 0; i < 12; i++)

printf("%c", 205);

printf("%c\n", 188);

printf("\n");

}

void Conv(float \* fun, int m, int \*arr) {

int A = 65;

float min = fun[0];

float max = min;

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

{

if (fun[i] > max) {

max = fun[i];

}

if (fun[i] < min)

{

min = fun[i];

}

}

double k = (double)A / (max - min);

printf("\n\nNatural numbers:\n");

printf("%c", 201);

for (int i = 0; i < 12; i++)

printf("%c", 205);

printf("%c", 203);

for (int i = 0; i < 12; i++)

printf("%c", 205);

printf("%c\n", 187);

printf("%c f(x) %c natural %c\n", 186, 186, 186);

printf("%c", 204);

for (int i = 0; i < 12; i++)

printf("%c", 205);

printf("%c", 206);

for (int i = 0; i < 12; i++)

printf("%c", 205);

printf("%c\n", 185);

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

{

printf("%c%10.2lf %c", 186, fun[i], 186);

fun[i] = ((fun[i] - min) \* (k));

arr[i] = /\*round\*/okr(fun[i]);

printf("%7d %c\n", arr[i], 186);

}

printf("%c", 200);

for (int i = 0; i < 12; i++)

printf("%c", 205);

printf("%c", 202);

for (int i = 0; i < 12; i++)

printf("%c", 205);

printf("%c\n", 188);

}

void Print(int \*arr, int m)

{

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

{

printf("%3d|", i);

for (int k = 0; k < arr[i]; k++)

printf(" ");

printf("%c\n", 0x2A/\*, arr[i]\*/);

}

printf("\n");

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

srand(time(NULL));

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

Value val;

bool flg = false;

do {

ReadPar(val);

if (val.x1 == val.x2 || val.x1 < val.x2 && val.dx < 0 || val.dx >(val.x2 - val.x1) || val.x1 > val.x2 && val.dx > 0) {

printf("ERROR! Invalid data.\n");

\_getch();

exit(1);

}

else if (val.x1 == 0 || val.dx == 0)

{

printf("'x1' and 'dx' must be different from 0! Check your input data.\n");

\_getch();

exit(1);

}

else {

flg = true;

}

} while (flg != true);

m\_f(val);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

Tab(val, m);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

Rand(fun, val, m);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

int \*arr = new int[m];

Conv(fun, m, arr);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n\n");

Print(arr, m);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_~~~~~~~Program by Artamonov O. U. TV - 61~~~~~~~\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\n\n");

\_getch();

return 0;

}

Print(arr, m);

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_~~~~~~~Program by Artamonov O. U. TV - 61~~~~~~~\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("\n\n");

\_getch();

return 0;

}

**Що таке імперативна парадигма програмування?**

Це парадигма програмування, згідно з якою описується процес отримання

результатів як послідовність інструкцій зміни стану програми. Подібно до того,

як з допомогою наказового способу в мовознавстві перелічується послідовність

дій, що необхідно виконати, імперативні програми є послідовністю

операцій комп'ютера для виконання. Поширений синонім імперативному

програмуванню є процедурне програмування.

**floor(val) -** функція виконує округлення значення val і повертає найбільше ціле значення, яке більше, ніж val.

**%** - залишок від ділення

**rand() -**  генерує числа в діапазоні від 0 до RAND\_MAX. RAND\_MAX - це константа, визначена в бібліотеці <cstdlib>. Для MVS RAND\_MAX = 32767, але воно може бути і більше, в залежності від компілятора.