

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
“ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ УПРАВЛІННЯ

ЗВІТ
З ЛАБОРАТОРНОЇ РОБОТИ №6
ЗА ТЕМОЮ «ВКАЗІВНИКИ НА ФУНКЦІЙ ТА ЗАГОЛОВНІ ФАЙЛИ»

Виконав студент
групи КН-221в
Шулюпов Єгор Русланович

Перевірів
Асистент кафедри
Ольховий Олексій Михайлович

ХАРКІВ 2021

Хід роботи

Завдання 1.1

Створити окрему одиницю трансляції, в якій реалізувати функцію виведення таблиці значень певної функції з визначеним кроком. Параметри функції – початок, кінець інтервалу, крок і вказівник на функцію, значення якої будуть виведені в таблиці. У заголовному файлі описати тип вказівника на функцію і прототип функції виведення таблиці значень. У файлі реалізації визначити функцію виведення таблиці значень.

В іншій одиниці трансляції розташувати функції, для яких слід вивести значення у таблиці, а також функцію `main()`, в якій здійснюється виведення таблиць значень не менш, ніж двох різних функцій. Одна з функцій для тестування – функція, визначена в завданні першої лабораторної роботи та реалізована в третій лабораторній роботі.

```
#include <iostream>
#include <fstream>
#include "func.h"
#include "out.h"
using namespace std;
```

```
int main()
{
    setlocale(LC_ALL, "");
    double x_1, x_2, step;
    do {
        cout << "Введіть x_1, x_2, n, step: ";
        cin >> x_1 >> x_2 >> n >> step;
    } while (checkValidInput(x_1, x_2, n, step));

    output(x_1, x_2, step, y_1);

    output(x_1, x_2, step, y_2);
    return 0;
}
```

```
#ifndef FUNC_H
#define FUNC_H
#include <iostream>
using namespace std;

double y_1(double x_1, double x_2, int n, double step);
double y_2(double x_1, double x_2, int n, double step);
bool checkValidInput(double x_1, double x_2, int n, double step);

#endif // !FUNC_H
```

```

#include "func.h"

double y_1(double x_1, double x_2, int n, double step) {
    while(x_1 <= x_2){
        double y = 0;
        if (x_1 < 0) {
            for (int i = 0; i <= (n - 1); i++) {
                int j = 0;
                for (j; j <= (n - 1); j++) {
                    y += 1 / (x_1 - i + x_1 * j);
                }
            }
            return y;
        }
        else {
            cout << "division by zero (x) \t" << x_1 << endl;
            break;
        }
        x_1 += step;
    }
}

double y_2(double x_1, double x_2, int n, double step) {
    while (x_1 <= x_2) {
        if (x_1 != 0) {
            double y = 1;
            for (int i = 1; i <= n; i++) {
                y *= (1 / x_1 - 1 / i);
            }
            return y;
        }
        else {
            cout << "division by zero (x)\t" << x_1 << endl;
            break;
        }
        x_1 += step;
    }
}

bool checkValidInput(double x_1, double x_2, int n, double step)
{
    if ((n < 1) || (step <= 0) || (x_1 > x_2)) {
        cout << "data must be ((n >= 1) || (step > 0) || (x_1 <= x_2))" << endl;
        return true;
    }
    else {
        return false;
    }
}

```

```

#include "out.h"
#include "func.h"
using namespace std;
int n;
void output(double x_1, double x_2, double step, double (*func)(double, double, int, double)) {
    for (x_1; x_1 <= x_2; x_1 += step) {
        cout << "x = " << x_1 << " " << "y = " << func(x_1, x_2, n, step) << endl;
    }
    cout << endl << endl;
}

```

```

#ifndef OUT_H
#define OUT_H
#include <iostream>
using namespace std;
extern int n;
void output(double x_1, double x_2, double step, double (*func)(double, double, int, double));
#endif // !OUT_H

```

Завдання 1.2

Написати програму, яка реалізує перебір значень на визначеному інтервалі з певним кроком з метою пошуку деякого значення відповідно до індивідуального завдання, наведеного в таблиці. Необхідне значення може бути знайдено шляхом перевірки проміжних значень функції (або першої / другої похідної). Слід використати вказівник на функцію, для якого визначити typedef.

Сирцевий код повинен бути розділений на дві одиниці трансляції. Перша одиниця трансляції буде представлена заголовним файлом і файлом реалізації. Визначення typedef, а також прототип функції пошуку потрібного значення, повинні бути розташовані в заголовному файлі. Визначення цієї функції слід здійснити у файлі реалізації. Функція для перевірки працездатності програми, а також функція main(), повинні бути розташовані в іншій одиниці трансляції.

Варіант 27 – Найбільший корінь другої похідної.

(ConsoleApplication2.cpp):

```
#include <iostream>
#include <cmath>
#include "def.h"
using namespace std;
int main()
{
    setlocale(LC_ALL, "");

    double x_1, x_2, step;
    do {
        cout << "Введіть x_1, x_2, step: ";
        cin >> x_1 >> x_2 >> step;

    } while (checkValidInput(x_1, x_2, step));

    cout << "max_1 = \t" << var27(x_1, x_2, step, mathFunc1) << endl;
    cout << "max_2 = \t" << var27(x_1, x_2, step, mathFunc2);

}
```

(def.h):

```
#include <iostream>
using namespace std;
typedef double (*sqrt2der)(double);

double mathFunc1(double x_1);

double mathFunc2(double x_1);

double var27(double x_1, double x_2, double step, sqrt2der func);

bool checkValidInput(double x_1, double x_2, double step);
```

(def.cpp):

```
#include "def.h"

double mathFunc1(double x_1) {
    return x_1 * x_1 - 2 * x_1 + 4;
}

double mathFunc2(double x_1) {
    return 8 * x_1 * x_1 * x_1 + 1;
}

double var27(double x_1, double x_2, double step, sqrt2der func) {
    double max = sqrt(((func(x_1 + 0.0000001)) - 2 * func(x_1) + (func(x_1 - 0.0000001)))
/ 0.0000001 * 0.0000001));
    for (x_1; x_1 <= x_2; x_1 += step)
    {
        double a = sqrt(((func(x_1 + 0.0000001)) - 2 * func(x_1) + (func(x_1 -
0.0000001))) / 0.0000001 * 0.0000001);
        if (a > max) {
            max = a;
        }
    }
    return max;
}

bool checkValidInput(double x_1, double x_2, double step)
{
    if ((step <= 0) || (x_1 > x_2)) {
        cout << "data must be ((step > 0) || (x_1 <= x_2))" << endl;
        return true;
    }
    else {
        return false;
    }
}
```

```

Введите x_1, x_2, step: 8
15
1
max_1 =          1.68587e-07
max_2 =          2.6974e-06
C:\Users\esulu\Desktop\folders\1 курс\labs_op\лаб_6\ConsoleApplication2\Debug\ConsoleApplication2.exe (процесс 14376) за
вершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Ав
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...

```

Вправи для контролю

Реалізувати приклади і вправи лабораторної роботи № 3 з розташуванням в окремій одиниці трансляції всіх функцій, крім main().

(ConsoleApplication3.cpp):

```

#include <iostream>
#include "dop_1.h"
#include "example_1.h"
using namespace std;

int main()
{
    setlocale(LC_ALL, "ru");

    return 0;
}

```

(example_1.cpp):

```

#include <iostream>
#include "example_1.h"
using namespace std;

double sumOfPowers(double x, int n)
{
    double sum = 0;
    double pow = 1;
    for (int i = 0; i < n; i++)
    {
        pow *= x;
        sum += pow;
    }
    return sum;
}

int add(int i)
{
    static int sum = 0;
    sum += i;
    return sum;
}

```

```

int read()
{
    int n;
    cout << "Input n:";
    cin >> n;
    return n;
}

int calc(int n)
{
    int y = 0;
    for (int i = 1; i <= n; i++)
    {
        y += i * i;
    }
    return y;
}

void write(int y)
{
    cout << "y = " << y;
}

int solve(double a, double b, double& x)
{
    if (a == 0)
    {
        if (b == 0)
        {
            return -1;
        }
        else
        {
            return 0;
        }
    }

    x = -b / a;
    return 1;
}

```

```

// функція введення даних з клавіатури, параметри передаються за посиланням
bool readData(double& from, double& to, double& step)
{
    cout << "Print start, end and step ";
    cin >> from >> to >> step;
    if (from >= to || step <= 0)
    {
        cerr << "Wrong data" << endl;
        return false;
    }
    return true;
}

// Функція, визначена в завданні
double y(double x)
{

```

```

    if (x < 0)
    {
        return sin(x);
    }
    else
    {
        return sqrt(x);
    }
}

// Виведення значень аргументів і функції на інтервалі з визначеним кроком
void printInALoop(double from, double to, double step)
{
    cout << "x\ty" << endl; // "шапка" таблиці
    for (double x = from; x <= to; x += step)
    {
        cout << x << "\t" << y(x) << endl;
    }
}

void ex_3_2() {
    int i;
    do
    {
        cin >> i;
        cout << add(i) << endl;
    } while (i);
}

void ex_3_3() {
    int n = read();
    int y = calc(n);
    write(y);
}

void ex_3_4() {
    double a, b, x;
    cin >> a >> b;
    int count = solve(a, b, x);
    switch (count)
    {
        case -1:
            cout << "infinity roots";
            break;
        case 0:
            cout << "No root";
            break;
        case 1:
            cout << "x= " << x;
    }
    cout << endl;
}

int ex_3_5() {
    double from, to, step;
    if (readData(from, to, step))
    {
        printInALoop(from, to, step);
        return 0; // таблицю виведено
    }
    return -1; // помилка введення даних
}

```


(example_1.h):

```
#ifndef HEADER_H_EX_1
#define HEADER_H_EX_1
double sumOfPowers(double x, int n);

int add(int i);

int read();
int calc(int n);
void write(int y);

int solve(double a, double b, double& x);

bool readData(double& from, double& to, double& step);
double y(double x);
void printInALoop(double from, double to, double step);

void ex_3_2();

void ex_3_3();

void ex_3_4();

int ex_3_5();

#endif
```

(dop_1.cpp):

```
#include <iostream>
#include "dop_1.h"
using namespace std;

void signum(double x)
{
    if (x == 0)
    {
        cout << "y = " << 0 << endl;
    }
    if (x > 0)
    {
        cout << "y = " << 1 << endl;
    }
    if (x < 0)
    {
        cout << "y = " << -1 << endl;
    }
}

double multy_3x(double z, double x, double c)
{
    return z * x * c;
}

double multiply_uneven(int n) {
    if (n > 1) {
        int y = 1, a = 1;
    }
}
```

```

        for (int i = 1; i <= n; i++) {
            y *= a;
            a += 2;
        }
        return y;
    }
    else
        cout << "data must be n > 0";
}

double ePowX(double x)
{
    double z = 1, y = 0, i = 1;

    while (z > 0.000000000000000001)
    {
        y += z;
        z *= x / i;
        i++;
    }
    return round(y * 10000) / 10000;
}

int factorial(int n) {
    int y = 0;
    if (n == 0) {
        y = 1;
    }
    else
        if (n < 0) {
            cout << "data must be n >= 0 ";
        }
        else
            if (n > 0) {
                y = 1;
                for (int i = 1; i <= n; i++) {
                    y *= i;
                }
            }
    return y;
}

double even(double x_1, double x_2) {
    if (x_1 <= x_2) {
        int x = ceil(x_1);
        for (x; x <= x_2; x++) {
            if ((x % 2) == 0) {
                return x;
            }
        }
    }
}

double multiply_even(int n) {
    if (n > 1) {
        int y = 1, a = 2;
        for (int i = 1; i <= n; i++) {
            y *= a;
            a += 2;
        }
    }
}

```

```

    }
    return y;
}
else
    cout << "data must be n > 0";
}

```

```

int gcd(int a, int b) {
    while (a > 0 && b > 0) {
        if (a > b) {
            a %= b;
        }
        else {
            b %= a;
        }
    }
    return a + b;
}

```

(dop_1.h):

```

#ifndef HEADER_H
#define HEADER_H
void signum(double x);

double multiply_3x(double z, double x, double c);

double multiply_uneven(int n);

double ePowX(double x);

int factorial(int n);

double even(double x_1, double x_2);

double multiply_even(int n);

int gcd(int a, int b);

#endif

```

Висновок:

Отже, в ході роботи я навчився на практиці використовувати вказівки на функції (також в якості аргумента іншої функції) та їх опис в заголовних файлах.