

Чернівецький національний університет імені Юрія Федьковича
Навчально-науковий інститут фізико-технічних та комп'ютерних наук
Відділ комп'ютерних технологій
Кафедра математичних проблем управління і кібернетики

Звіт
про виконання лабораторної роботи №1

Тема: “ Обробка виключних ситуацій мови C++.”

з дисципліни
“ Літня обчислювальна практика”

Варіант № 12

Виконав:
ст. гр. 241 Подарунок М.
Прийняв:
доц. Лазорик В. В.

Чернівці – 2025

Практика. Лабораторна роботи №1.

Виконання лабораторної роботи

1. Зайти в свій обліковий запис на github.com. Зайти в [github classroom](https://github.com).
2. Клонувати репозиторій <https://classroom.github.com/a/VGcAmpbM> в свій обліковий запис на github.com.
3. Розв'язати завдання.
4. Вихідних код записати в створений репозиторій.

Завдання для лабораторної роботи

Задача 6. Задано текстові файли *dat1.dat*, *dat2.dat* та *dat3.dat*, які містять інформацію таблиця 4, таблиця 5 та таблиця 6 відповідно, та дійсні змінні x , y та z , текстовий рядок *text* які вводяться із стандартного потоку введення. Обчислити значення функції $func_regr(r, m, k)$, яка знаходиться за алгоритмами у порядку пріоритету. Алгоритм 1.

1. $func_regr(r, k, m) = 10 * k * r^2 + m * r;$
2. $r = func(x, y, z) = Rsv(x, y, z) + 0.5 * Rsv(y, z, x) * Rsv(z, x, y);$
 $z = Qnk(x, y) + x * y, z \quad x \text{ and } z \quad y * Qnk(z, y)$
 $y = z, x \quad y \text{ and } x \quad z$
3. $Rsv(x, y, z)$
 $y * Qnk(x, z)$
 $x * z, y \quad x \text{ and } y \quad z$
 $z * Qnk(y, z) + Qnk(z, x), z \quad x \text{ or } y \quad z \text{ or } x \quad y$
4. $Qnk(x, y) = Qkn(x, y) + x * Qkn(y, z)$
5. $Qkn(x, y) = x / U(x) + y * T(y)$
6. Функції $U(x)$ та $T(x)$ за даними, які вводяться, з файлів

відповідно *dat1.dat*, *dat2.dat*.

- 6.1. Якщо файл *dat1.dat* не відкривається, або відсутній, тоді функцію $Qnk(x, y)$ порахувати за Алгоритмом 2;
- 6.2. Якщо $x \leq 5$, тоді функцію $Qnk(x, y)$ порахувати за Алгоритмом 2;
- 6.3. Якщо $x = x_i, i = 1, n$, шукаємо x_i та x_{i-1} , такі що $x_i \leq x \leq x_{i+1}$, тоді $U(x) = (U(x_i) - U(x_{i-1})) * (x - x_{i-1}) / (x_i - x_{i-1});$
- 6.4. Якщо файл *dat2.dat* не відкривається, або відсутній функцію $func(x, y, z)$ порахувати за Алгоритмом 2.

- 6.5. Якщо $x \leq 10$, тоді функцію $Qnk(x, y)$ порахувати за Алгоритмом 2;
- 6.6. Якщо $x = x_i, i = 1, n$, шукаємо x_i та x_{i+1} , такі що $x_i \leq x \leq x_{i+1}$, тоді $T(x) = T(x_i) + (T(x_{i+1}) - T(x_i)) * (x - x_i) / (x_{i+1} - x_i)$.
7.
$$k = Ktext(x, y, z, text) = Stext(Min(x, y, x - z, y - z), text), z = 0$$
8. Функція $Min(x, y, z, u)$ обчислює мінімальне значення серед заданих параметрів. $GetFrom(text) = x, x = 0$;
9.
$$Stext(x, text) = GetFrom("text") = Gtext("set") = x, text$$

$$";$$

$$GetFrom("get") = Gtext(text), x = 0$$

 Умова $text = ""$, означає що текстовий рядок порожній.
10. Для обчислення функції $GetFrom(text)$ відкриваємо файл $dat3.dat$.
- 10.1. Якщо файл $dat3.dat$ не відкривається, або відсутній, тоді повернути значення нуль (0);
- 10.2. Здійснити пошук відповідного слова.
- 10.3. Якщо слово знайдено повернути значення яке відповідає заданому слову у файлі;
- 10.4. Якщо слово відсутнє повернути значення нуль (0).
11.
$$m = Mtl(x, y, z) = x * Mts(x, y) + z * Mts(z, y);$$
12.
$$Mts(x, y) = x * Tsm(x, y) + y * Tsm(x, x)$$
13.
$$Tsm(x, y) = \log_2(5x^4 + 3x^2 + 2 * 4y^4 + x^2) * Ysm(y, x)$$
- 13.1. Якщо $4 * y^4 + x^2 \leq 0$, тоді функція $Mts(x, y)$ повертає значення нуль (0);
- 13.2. Якщо $5x^4 + 3x^2 + 2 * 4y^4 + x^2 > 0$, тоді змінні m присвоїти нуль (0);
14.
$$Ysm(x, y) = Y(x) * y + 0.7 * Y(y)$$
15.
$$Y(x) = \sqrt{\ln(x * 100 + x^2)}$$
- 15.1. Якщо $100 + x^2 \leq 0$, тоді змінні m присвоїти – нуль (0);
- 15.2. Якщо $x * \sqrt{100 + x^2} > 1$, тоді змінні m присвоїти – один (1); Алгоритм 2.

9. $Qnk(x, y) \square x * Qnk1(x, y) \quad y * Qnk1(y, x) \quad 0.05 * Qnk1(x, y) * Qnk1(y, x);$
10. $Qnk1(x, y) \square 1.15 * Qqn1(x, y, x \quad y) \quad 0.95 * Qqn1(y, x, x \quad y)$
11. $Qqn1(x, y, z) \square x / U1(x) \quad y * T1(y) \quad U1(z) * T1(z)$
12. $U1(x) \square arctg(arcsin(\sin(3x)))$
13. $T1(x) \square arctg(arccos(\sin(2x)))$

Таблица 4

x	U(x)
-5,0000	0,2801
-4,5000	0,2093
-4,0000	0,6190
-3,5000	0,8811
-3,0000	1,0422
-2,5000	1,1463
-2,0000	1,2176
-1,5000	1,2560
-1,0000	1,1998
-0,5000	1,1209
0,0000	1,0039
0,5000	0,8196
1,0000	0,5187
1,5000	0,0707
2,0000	0,4054

Таблица 5

x	T(x)
-10,0000	0,7832
-9,0000	1,1063
-8,0000	1,2486
-7,0000	1,1587
-6,0000	0,9105
-5,0000	0,2801
-4,0000	0,6190
-3,0000	1,0422
-2,0000	1,2176
-1,0000	1,1998
0,0000	1,0039
1,0000	0,5187
2,0000	0,4054
3,0000	0,9603
4,0000	1,1803

Таблица 6

text	x
aet	1,175
bet	1,278
cet	1,381
set	1,484
get	1,587
ret	1,69
het	1,793
met	1,896
net	1,999
qet	2,102
tet	2,205
wet	2,308
yet	2,411
iet	2,514
oet	2,617

2,5000	0,7487
3,0000	0,9603
3,5000	1,0926
4,0000	1,1803
4,5000	1,2418
5,0000	1,2338

5,0000	1,2338
6,0000	1,0761
7,0000	0,7068
8,0000	0,1450
9,0000	0,8533
10,0000	1,1347

pet	2,72
det	2,823
fet	2,926
let	3,029
zet	3,132
vet	3,235

```

#include <iostream>
#include <fstream>
#include <cmath>
#include <string>
#include <map>
#include <stdexcept>
using namespace std;

// ----- Таблична інтерполяція -----
// Шукаємо значення функції для довільного x з таблиці в заданому файлі
double interpolate(double x, const string& filename) {
    ifstream file(filename);
    if (!file) throw runtime_error("Не вдалося відкрити файл: " + filename);

    double x0, y0, x1, y1;
    file >> x0 >> y0;

    if (x == x0) return y0;

    while (file >> x1 >> y1) {
        if (x == x1) return y1;
        if (x > x0 && x < x1)
            return y0 + (y1 - y0) * (x - x0) / (x1 - x0); // лінійна інтерполяція
        x0 = x1;
        y0 = y1;
    }

    throw out_of_range("Значення " + to_string(x) + " поза діапазоном в " + filename);
}

// ----- Функція Y(x) -----
// Використовується в логарифмах. Перевіряємо допустимість підлогарифмічного виразу
double Y(double x) {
    double val = 100 * x - x * x;
    if (val <= 0) throw domain_error("Y(x): Недопустиме значення під логарифмом");
    return log(val);
}

// ----- Tsm(x, y) -----
// Частина підрахунку m: log10 виразу з Y(x) і Y(y)
double Tsm(double x, double y) {
    if (4 * x * y < 0) return 0;
    try {
        double val = 4 * x * y - Y(x) + Y(y);
        if (val <= 0) return 0;
    }
}

```

```

        return log10(val);
    }
    catch (...) {
        return 0;
    }
}

// ----- Mts(x, y) = Tsm(x, y) - Tsm(y, x) -----
double Mts(double x, double y) {
    return Tsm(x, y) - Tsm(y, x);
}

// ----- Mtl(x, y, z) -----
// Обчислення змінної m, з урахуванням винятків
double Mtl(double x, double y, double z, int& m) {
    double mts1 = Mts(x, y);
    double mts2 = Mts(y, z);
    double result = mts1 + mts2;

    // Перевірки, які можуть змінити значення m
    try {
        double yx = Y(x);
        if (100 * x - x * x < 0) m = 0;
        if (100 * x - x * x < 100) m = 1;
        if (4 * x * y - yx + Y(y) < 0 || 4 * y * z - Y(y) + Y(z) < 0) m = 0;
    }
    catch (...) {
        m = 0;
    }

    return result;
}

// ----- GetFrom(text, x) -----
// Читання значення, що відповідає слову 'text' з файлу dat3.dat
double GetFrom(const string& text, double x) {
    ifstream file("dat3.dat");
    if (!file) return 0;

    string key;
    double val;
    while (file >> key >> val) {
        if (key == text) return val;
    }
    return 0;
}

```

```

// ----- Пошук мінімального -----
double MinS(double x, double y, double z, const string& text) {
    return min({ x, y, z });
}

// ----- Qkn(x, y) = (U(x) + T(y)) / x -----
double Qkn(double x, double y) {
    try {
        double Ux = interpolate(x, "dat1.dat");
        double Ty = interpolate(y, "dat2.dat");
        if (x == 0) throw domain_error("Ділення на нуль у Qkn");
        return (Ux + Ty) / x;
    }
    catch (...) {
        throw;
    }
}

// ----- Qnk(x, y) = Qkn(x, y) + Qkn(y, x) -----
double Qnk(double x, double y) {
    return Qkn(x, y) + Qkn(y, x);
}

// ----- Rsv - умовна функція з 4-ма гілками -----
double Rsv(double x, double y, double z) {
    if (x > y && z > x) return Qnk(z, y) * z;
    else if (z > x && y > z) return Qnk(x, z) * x;
    else if (y > x && z > y) return Qnk(y, x) * y;
    else if (x == z || y == z) return Qnk(z, y) - Qnk(z, y);
    return 0;
}

// ----- fun_cr(x, y, z) -----
double fun_cr(double x, double y, double z) {
    return Rsv(x, y, z) + 0.5 * Rsv(y, z, x) + Rsv(z, x, y);
}

// ----- k = Ktext(x, y, z, text) -----
double Ktext(double x, double y, double z, const string& text) {
    if (text == "") {
        if (x + GetFrom(text, x) <= 0) return MinS(x, y, z, text);
    }
    else {

```



```

    }
    return 1;
}

// ----- Основна функція з алгоритму -----
double func_regr(double r, double m, double k) {
    return r * m * k - 10 * r * r;
}

// ----- Головна функція -----
int main() {
    double x, y, z;
    string text;

    cout << "Введіть x, y, z та текстовий рядок (без пробілів): ";
    cin >> x >> y >> z >> text;

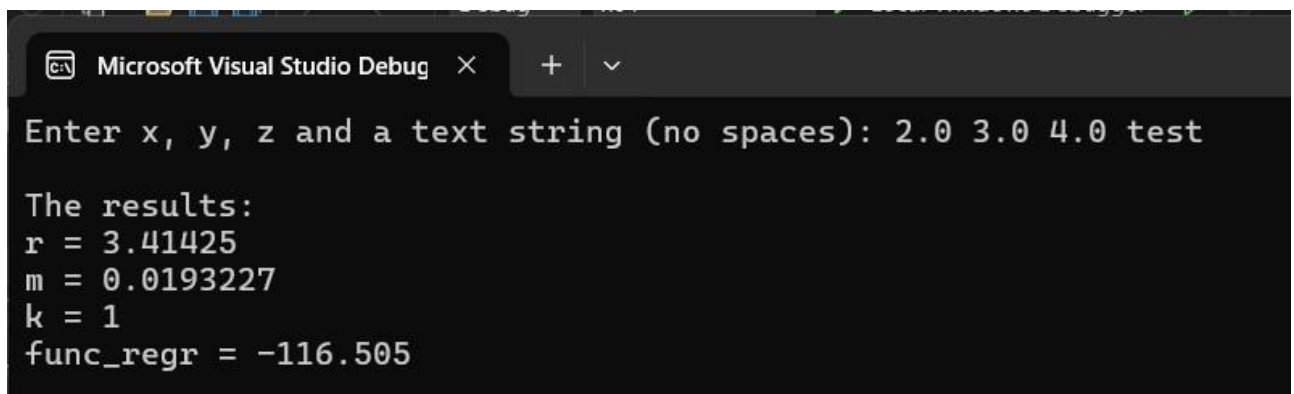
    try {
        int m_flag = -1; // використовується для фіксації змін у m
        double r = fun_cr(x, y, z); // розрахунок r
        double m = Mtl(x, y, z, m_flag); // розрахунок m
        double k = Ktext(x, y, z, text); // розрахунок k
        double result = func_regr(r, m, k); // обчислення підсумкової функції

        cout << "\nРезультати:\n";
        cout << "r = " << r << "\nm = " << m << "\nk = " << k << "\nfunc_regr = " << result << endl;
    }
    catch (const exception& e) {
        cerr << "\nПомилка: " << e.what() << endl;
    }

    return 0;
}

```

Рис. 1 – Код завдання



```

Microsoft Visual Studio Debug  X  +  v

Enter x, y, z and a text string (no spaces): 2.0 3.0 4.0 test

The results:
r = 3.41425
m = 0.0193227
k = 1
func_regr = -116.505

```

Рис. 2 – Результат виконання завдання

Висновок: Під час виконання лабораторної роботи я навчився використовувати механізм обробки виключних ситуацій у мові C++. Була реалізована функція $\text{fun}(x, y, z)$ з вкладеними обчисленнями та обробкою помилок, таких як відсутність файлів і ділення на нуль. Також використано інтерполяцію значень із файлів. Завдяки цьому програма стала більш надійною та захищеною від збоїв.