

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №4
за курсом "Програмування та алгоритмічні мови"
студента/студентки групи ПС-24-1
Бушиної Ольги Владиславівни
Кафедри прикладної математики та інформаційних технологій
ДНУ, 2024-2025

Тема: «Вирішення задач за допомогою розгалужень: “if, else”, “switch, case”, “ternary operation”»
«Basic I/O operations»

Постановка задачі:

Скласти звіт по виконаних завданнях, та завантажити звіт та коди програм на GitHub.

Створивши папку

TheBasicsOfProgrammingCppCourse/Lab4/...

1) Задача ABS:

=====

Складіть програму “ABS”, котра:

- a. питає в людини дійсне число X ;
- b. обчислює модуль X ;
- c. виводить результат на екран у наступному вигляді:

$| \text{значення } X | = \text{Значення модуля } X$

Приклад:

$X = -12$

$| X | = | -12 | = 12$

Приклад:

$X = 31$

$| X | = | 31 | = 31$

Підказка: Легко зробити через “if, else” чи “ternary operation”

Зобразити алгоритм задачі «ABS» у вигляді блок-схеми. За допомогою одного з варіантів: yEd Graph Editor: <https://www.yworks.com/products/yed-live/>

DrawIO: <https://app.diagrams.net/>

Google Presentation: <https://docs.google.com/presentation> Або іншого зручного для вас софту.

2) Задача Temperature

Ви опинилися на планеті з невідомими атмосферними умовами, і у вас немає точних даних про температуру зовні. На планеті може бути холодно, тепло, можливі комфортні умови, проте можуть бути і надзвичайно високі або низькі температури. Але у вас є робот, який вимірює температуру в градусах Цельсія та передає її на корабель. Задача, написати програму для цього робота, яка на вхід приймає температуру в градусах Цельсія та надає інформацію у вигляді зрозумілому для людини.



Приклад:

Вхід: 18

Вихід: Комфортна температура: можна виходити.

3) Задача «Узник Замку IF»

Ви опинилися в таємничій камері «Замку IF». Досліджуючи її, ви знайшли потайні двері, які, схоже, ведуть назовні. Але двері зачинені та вимагають введення пароля для розблокування. Уважно оглянувши кімнату, ви знаходите старий запис з таємничим кодом: FR-33-D0M. Можливо, це підказка!

Завдання: Напишіть програму, яка поетапно прийматиме символи від користувача, щоб перевірити правильність введення пароля. Програма має запитувати символи один за одним та порівнювати їх із кодом FR-33-D0M. Якщо користувач вводить правильний символ на кожному етапі, двері відкриваються, і на екрані з'являється напис "Двері відчинено! Ви вільні!". Якщо хоча б один символ введено неправильно, програма виводить повідомлення "Неправильний пароль, спробуйте ще раз." і починає перевірку спочатку. Підказка:

- Пароль складається з символів, включаючи літери, цифри та дефіси.
- Якщо якийсь з символів буде введено неправильно, програма виведе повідомлення "Неправильний пароль, спробуйте ще раз." та закриється, і потрібно буде повторно її вмикати.

Приклад роботи програми: Введіть символ 1: F

Введіть символ 2: R

Введіть символ 3: - Введіть символ 4: 3

Введіть символ 5: 3

Введіть символ 6: - Введіть символ 7: D

Введіть символ 8: 0

Введіть символ 9: M Результат:

Двері відчинено! Ви вільні!

4) Задача «Day of the Week»

=====

Програма:

- Запитує у користувача, ввести з клавіатури будь-яке число.
- Знаходить відносно числа 1-7 день тижня Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday

Підказка: найкраще вирішити цю задачу, це за допомогою “switch, case”

Приклад:

Вхід: 5

Вихід: Friday

5) Задача «Chess Figure»

=====

Згідно з вашим номером варіанту, оберіть **одну** з шахових фігур:

1 - King, 2 – Queen, 3 – Rook, 4- Bishop, 5 - Knight, 6 - Pawn. (якщо у вас 7 номер, то 1 – **King**, ...)

Ваша задача — перевірити, чи погрожує обрана фігура певній клітинці на шаховій дошці.

У звіті необхідно описати логіку того, як рухається ваша фігура.

Правила можна почитати тут: https://en.wikipedia.org/wiki/Rules_of_chess

Зіграти тут: <https://www.chess.com/>

Умови завдання:

1. Спершу оберіть шахову фігуру відповідно до свого варіанту.

2. Введіть початкову позицію фігури на шаховій дошці (наприклад, e4).
3. Введіть координати клітинки, яку потрібно перевірити на загрозу (наприклад, g5).
4. Програма повинна визначити, чи може обрана вами фігура за правилами свого руху потрапити на клітинку, яку потрібно перевірити.

Підказка: з початку необхідно першу введену літеру a-h перевести у 1-8, і розглядати позицію e4 як пару чисел (5, 4), а далі порівнювати чи загрожує вона координатам другої клітинки.



2. Вихідний текст програми розв'язку задачі

```
#include <iostream>
#include <cmath>
#include <Windows.h>
int main()
{
    using namespace std;
    SetConsoleOutputCP(1251);
    cout << "Task1"<<endl;
    double x;
    double result;
    cout << "Введіть число X: ";
    cin >> x;
    if (x >= 0)
    {
        result = x;
    }
    else
    {

```

```

    result = -x;
}

cout << "| " << "X" << " | " << "=" << " | " << x << " | =" << result << endl;

cout << "Task2"<<endl;
int temperature;
cout << "Введіть температуру в градусах Цельсія: ";
cin >> temperature;

if (temperature < 0) {
    cout << "Дуже холодно! Залишайтеся всередині." << endl;
}
else if (temperature <= 8) {
    cout << "Холодно, краще не виходити." << endl;
}
else if (temperature <= 16) {
    cout << "Комфортна температура: можна виходити." << endl;
}
else if (temperature <= 40) {
    cout << "Тепло, можна виходити." << endl;
}
else if (temperature <= 45) {
    cout << "Дуже спекотно, можна обережно вийти." << endl;
}
else {
    cout << "Дуже спекотно! залишайтеся всередині." << endl;
}

cout << "Task3"<<endl;
const string password = "FR-33-D0M";
char y;
while (true) {
    int i = 0;
    while (i < password.size()) {
        cout << "Введіть символ " << i + 1 << ": ";
        cin >> y;

        if (y != password[i]) {

```

```

        cout << "Неправильний пароль, спробуйте ще раз." <<endl;
        break;
    }
    i++;
}
if (i == password.size()) {
    cout << "Двері відчинено! Ви вільні!" <<endl;
    break;
}
}

```

```

cout << "Task4"<<endl;
int day;
cout << "Введіть число від 1 до 7: ";
cin >> day;
switch (day) {
case 1:
    cout << "Monday" <<endl;
    break;
case 2:
    cout << "Tuesday" <<endl;
    break;
case 3:
    cout << "Wednesday" <<endl;
    break;
case 4:
    cout << "Thursday" <<endl;
    break;
case 5:
    cout << "Friday" <<endl;
    break;
case 6:
    cout << "Saturday" <<endl;
    break;
case 7:
    cout << "Sunday" <<endl;
    break;
default:
    cout << "Неправильний ввід. Введіть число від 1 до 7." <<endl;
}

```

```
}
```

```
cout << "Task5" << endl;
```

```
char column1, column2;
```

```
int row1, row2;
```

```
// Введення координат для початкової позиції тури
```

```
cout << "Введіть букву колонки початкової позиції тури (a-h): ";
```

```
cin >> column1;
```

```
cout << "Введіть номер рядка початкової позиції тури (1-8): ";
```

```
cin >> row1;
```

```
// Введення координат для позиції, яку потрібно перевірити на загрозу
```

```
cout << "Введіть букву колонки для перевірки загрози (a-h): ";
```

```
cin >> column2;
```

```
cout << "Введіть номер рядка для перевірки загрози (1-8): ";
```

```
cin >> row2;
```

```
// Перетворення літер на числа
```

```
int start1 = column1 - 'a' + 1;
```

```
int start2 = column2 - 'a' + 1;
```

```
if (row1 == row2 || start1 == start2) { // Якщо на одній вертикалі або  
горизонталі
```

```
    cout << "Тура загрожує клітинці " << column2 << row2 << "." << endl;
```

```
}
```

```
else {
```

```
    cout << "Тура не загрожує клітинці " << column2 << row2 << "." << endl;
```

```
}
```

```
return 0;
```

```
}
```

3.Опис інтерфейсу програми

