

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 9
з навчальної дисципліни
“Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ МОДУЛІВ РОЗГАЛУДЖЕННЯ ТА
ІТЕРАЦІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

ЗАВДАННЯ ВИДАВ
доцент кафедри кібербезпеки та
програмного забезпечення
Доренський О. П.
<https://github.com/odorenskyi/>

ВИКОНАВ
студент академічної групи КБ-20
Лук'яненко Р. О.

ПЕРЕВІРИВ
ст. викладач кафедри кібербезпеки
та програмного забезпечення
Поліщук Л. І.

Кропивницький – 2023

Варіант 2

Тема: Реалізація програмних модулів розгалуження та ітераційних обчислювальних процесів.

Мета: Набутті ґрунтовних вмінь і практичних навичок реалізації технології модульного програмування, застосування операторів C/C++ арифметичних, логічних, побітових операцій, умови, циклів та вибору під час розроблення статичних бібліотек, заголовкових файлів та програмних засобів.

Завдання 9.1:

Вхід: бал за офіційною шкалою Всесвітньої Морської Організації.

Вихід: висота хвиль (м) та опис хвилювання моря.

Лістинг 9.1:

```
#include <iostream>

#include <string>

using namespace std;

int main() {
    int grade;

    float min_wave_height, max_wave_height;

    string description;

    cout << "Enter the grade (0-9): ";

    cin >> grade;

    if (grade == 0) {
        description = "Calm (Glassy)";

        min_wave_height = 0;

        max_wave_height = 0;
```

```
}  
else if (grade == 1) {  
    description = "Light air (Rippled)";  
    min_wave_height = 1.0f;  
    max_wave_height = 1.0f + (1.0f/3.0f);  
}  
else if (grade == 2) {  
    description = "Light breeze (Smooth)";  
    min_wave_height = 2.0f;  
    max_wave_height = 2.0f + (2.0f/3.0f);  
}  
else if (grade == 3) {  
    description = "Gentle breeze (Slight)";  
    min_wave_height = 3.0f;  
    max_wave_height = 3.0f + (3.0f/3.0f);  
}  
else if (grade == 4) {  
    description = "Moderate breeze (Moderate)";  
    min_wave_height = 4.0f;  
    max_wave_height = 4.0f + (8.0f/8.0f);  
}  
else if (grade == 5) {  
    description = "Fresh breeze (Rough)";  
    min_wave_height = 8.0f;  
    max_wave_height = 8.0f + (5.0f/13.0f);  
}  
else if (grade == 6) {  
    description = "Strong breeze (Very rough)";  
    min_wave_height = 13.0f;
```

```

    max_wave_height = 13.0f + (7.0f/20.0f);
}
else if (grade == 7) {
    description = "Near gale (High)";
    min_wave_height = 20.0f;
    max_wave_height = 20.0f + (10.0f/30.0f);
}
else if (grade == 8) {
    description = "Gale (Very high)";
    min_wave_height = 30.0f;
    max_wave_height = 30.0f + (15.0f/45.0f);
}
else if (grade == 9) {
    description = "Storm (Phenomenal)";
    min_wave_height = 45;
    max_wave_height = 1000;
}
else {
    cout << "Invalid grade entered." << endl;
    return 0;
}

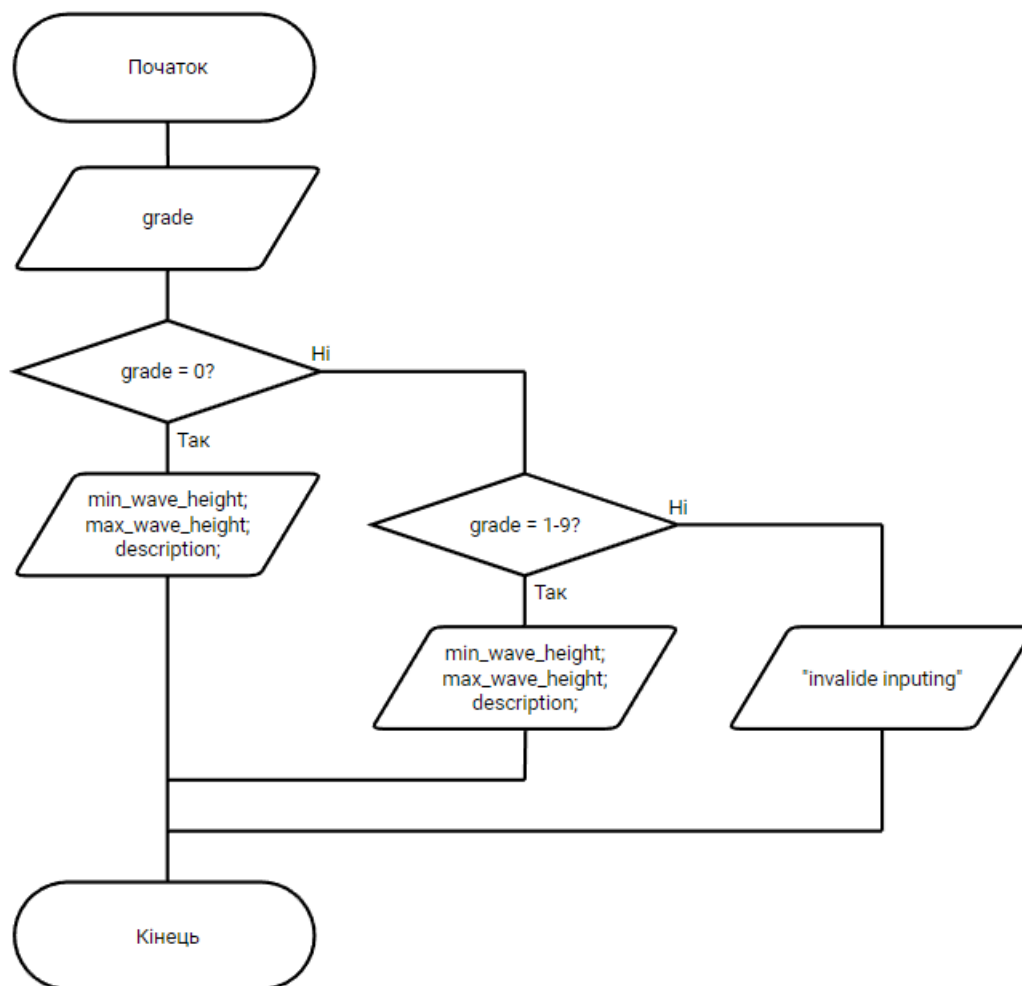
cout << "Wave height range: " << min_wave_height << "m - " << max_wave_height << "m" <<
endl;

cout << "Description: " << description << endl;

return 0;
}

```

Блок-схема 9.1:



Завдання 9.2:

Вхід: n чисел.

Вихід: кількість введених від'ємних чисел, рівних нулю, а також у діапазоні від 5 до 1024.

Лістинг 9.2:

```
#include <iostream>

int main() {

    int n, negative_count = 0, zero_count = 0, range_count = 0;

    std::cout << "Enter the number of inputs: ";
    std::cin >> n;

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

        int num;

        std::cout << "Enter a number: ";
        std::cin >> num;

        if (std::cin.fail()) {
            std::cout << "Invalid input. Exiting..." << std::endl;
            break;
        }

        if (num < 0) {
            negative_count++;
        } else if (num == 0) {
            zero_count++;
        } else if (num >= 5 && num <= 1024) {
            range_count++;
        }
    }
}
```

```

    }

}

std::cout << "Negative count: " << negative_count << std::endl;

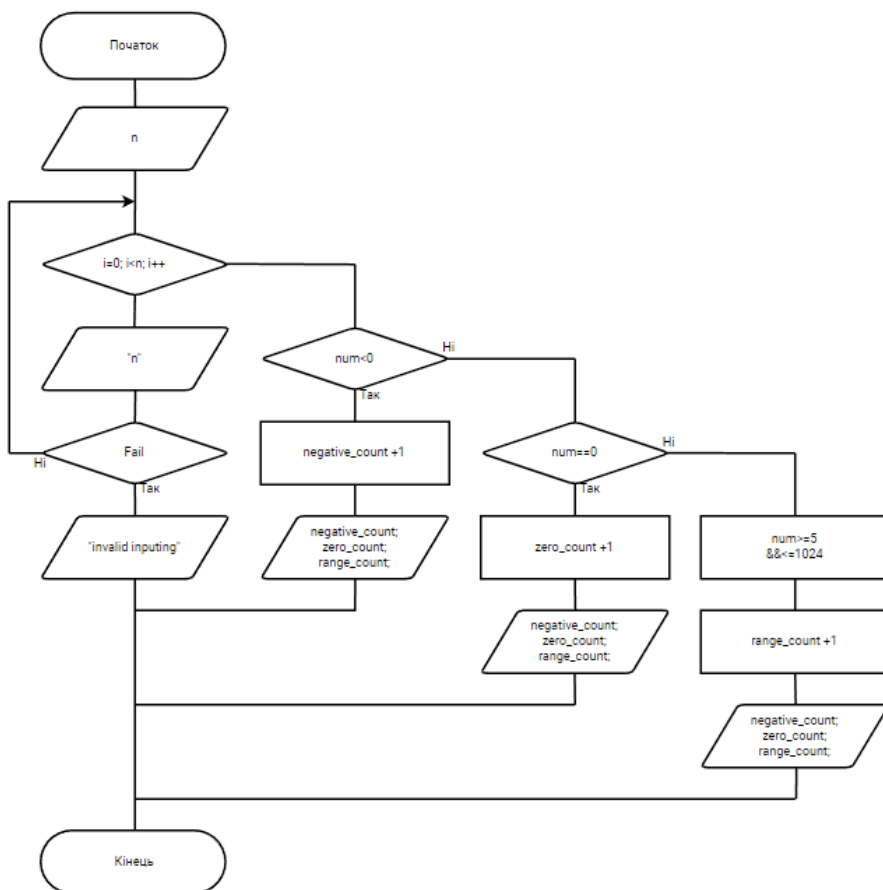
std::cout << "Zero count: " << zero_count << std::endl;
std::cout << "Range count: " << range_count << std::endl;

return 0;

}

```

Блок-схема 9.2:



Завдання 9.3:

Вхід: натуральне число N від 0 до 70700.

Вихід: якщо біт D_1 числа N рівний 1, кількість двійкових нулів у ньому, інакше — кількість двійкових одиниць*.

Лістинг 9.3:

```
#include <iostream>
#include <bitset>

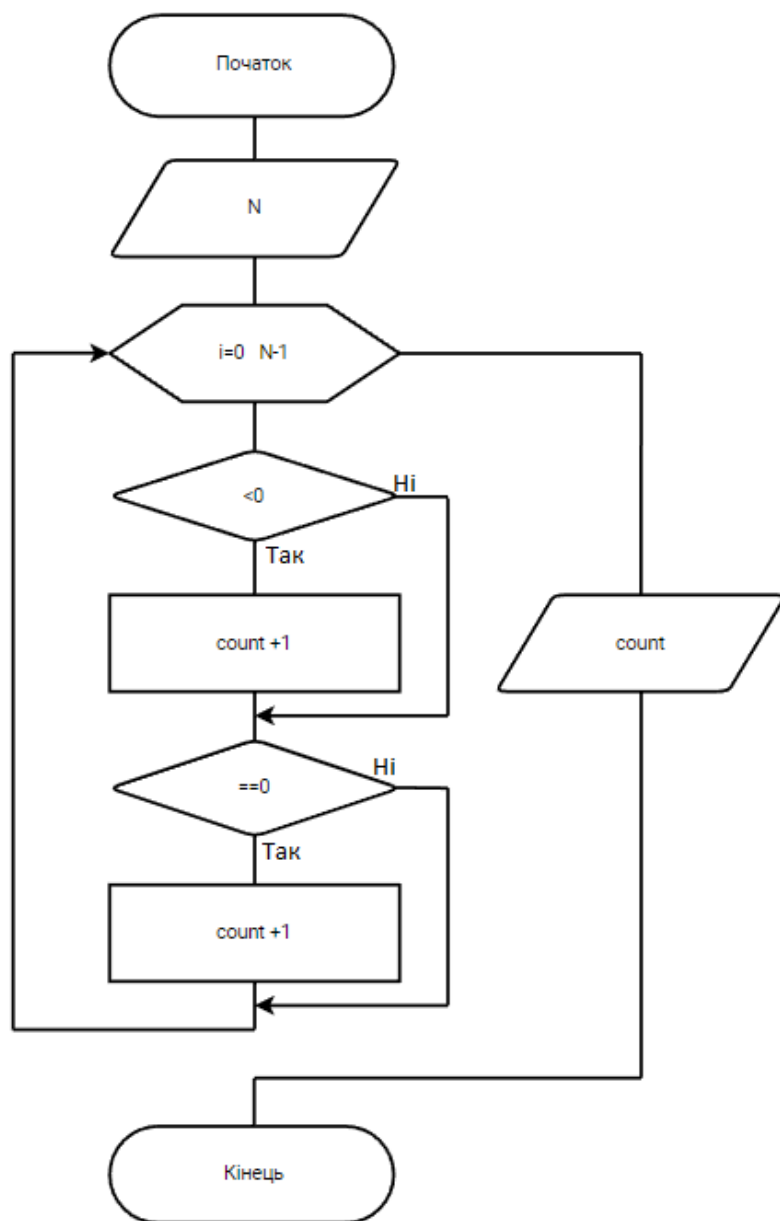
int main() {
    int N;
    std::cin >> N;

    bool D1 = std::bitset<32>(N).test(0);

    int count = 0;
    if (D1) {
        while (N) {
            if ((N & 1) == 0) {
                count++;
            }
            N >>= 1;
        }
    }
    else {
        while (N) {
            if ((N & 1) == 1) {
                count++;
            }
            N >>= 1;
        }
    }

    std::cout << count << std::endl;
    return 0;
}
```


Блок-схема 9.3:



Завдання 9.4:

За введеним користувачем символом “q” викликається `s_calculation()`, “e” – функція задачі 9.1, “r” – функція задачі 9.2, “t” – функція задачі 9.3; якщо користувач вводить інші символи, вони ігноруються, при чому видається звуковий сигнал про помилкове введення. Після цього, якщо користувач за запитом додатка вводить символ “Y”, “y” або “h”, відбувається вихід з програми, інакше — виконання програми повторюється.

Лістинг 9.4:

```
#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

double s_calculation(int x, int y, int z) {
    double s = pow(x, y) + sqrt(abs(z));
    return s;
}

int main() {
    char choice = ' ';

    while (choice != 'y' && choice != 'Y' && choice != 'h') {
        cout << "Enter 'q' to calculate s, 'e' for task 8.1, 'r' for task 8.2, 't' for task 8.3: ";
        cin >> choice;

        switch (choice) {
            case 'q': {
                int x, y, z;
                cout << "Enter x, y, z: ";
```

```

    cin >> x >> y >> z;

    double s = s_calculation(x, y, z);

    cout << "Result: " << setprecision(5) << fixed << s << endl;

    break;
}

case 'e': {

    int n, negative_count = 0, zero_count = 0, range_count = 0;


    cout << "Enter the number of inputs: ";

    cin >> n;


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

        int num;

        cout << "Enter a number: ";

        cin >> num;


        if (cin.fail()) {

            cout << "Invalid input. Exiting..." << endl;

            break;

        }


        if (num < 0) {

            negative_count++;

        } else if (num == 0) {

            zero_count++;

        } else if (num >= 5 && num <= 1024) {

            range_count++;

        }

    }

}

```

```

        cout << "Negative count: " << negative_count << endl;

        cout << "Zero count: " << zero_count << endl;

        cout << "Range count: " << range_count << endl;

        break;
    }

    case 'r': {
        unsigned int n;

        cout << "Enter a positive integer n: ";
        cin >> n;

        int count = 0;

        bool d1 = (n & 0x1); // Check bit D1

        if (d1) {
            // Count binary zeros

            while ((n & 0x1) == 0 && n != 0) {

                count++;

                n >>= 1;

            }

        } else {

            while (n != 0) {

                count += (n & 0x1);

                n >>= 1;

            }

        }

        cout << "Result: " << count << endl;

        break;
    }

```

```

case 't': {
    unsigned int n;

    cout << "Enter a positive integer n: ";
    cin >> n;

    int count = 0;
    bool d1 = (n & 0x1); // Check bit D1
    if (d1) {
        while ((n & 0x1) == 0 && n != 0) {
            count++;
            n >>= 1;
        }
    } else {
        // Count binary ones
        while (n != 0) {
            count += (n & 0x1);
            n >>= 1;
        }
    }

    cout << "Result: " << count << endl;
    break;
}

default: {
    cout << '\a'; // Beep sound
    break;
}
}
}

```

return 0

Принцип роботи коду 9.4:

Користувач вводить символ з клавіатури.

1. Програма перевіряє введений символ і виконує певну дію в залежності від нього.
2. Якщо введений символ є "q", програма викликає функцію `s_calculation()`, яка обчислює значення виразу $s = x^y + \sqrt{|z|}$, де x , y та z - цілі числа, які вводяться з клавіатури користувачем. Результат обчислення виводиться на екран.
3. Якщо введений символ є "e", програма викликає функцію з завдання 8.1, яка зчитує кількість цілих чисел, які вводить користувач, та обчислює кількість від'ємних чисел, нулів та чисел, які належать діапазону $[5, 1024]$. Результати обчислень виводяться на екран.
4. Якщо введений символ є "r", програма викликає функцію з завдання 8.2, яка зчитує натуральне число n та обчислює кількість двійкових одиниць в його двійковому поданні, якщо біт $D1$ дорівнює 0, та кількість двійкових нулів, якщо біт $D1$ дорівнює 1. Результат обчислення виводиться на екран.
5. Якщо введений символ є "t", програма викликає функцію з завдання 8.3, яка зчитує натуральне число n та обчислює кількість двійкових одиниць в його двійковому поданні, якщо біт $D1$ дорівнює 0, та кількість двійкових нулів, якщо біт $D1$ дорівнює 1. Результат обчислення виводиться на екран.
6. Якщо введений символ не відповідає жодному з перерахованих вище варіантів, програма ігнорує його та відтворює звуковий сигнал.

Висновок: Ця лабораторна робота може бути корисною для практикування роботи з циклами, умовними конструкціями, бітовими операціями та функціями.

Виконання цих завдань допоможе збільшити розуміння того, як працюють різні елементи мов програмування та як їх можна використовувати для рішення різноманітних задач.

Також важливо зазначити, що здійснення правильних перевірок та обробка помилок є дуже важливим елементом в програмуванні. Це допомагає уникнути некоректної роботи програм та їх збоїв, які можуть призвести до небезпечних ситуацій. Тому слід дотримуватися найкращих практик з перевірки введених користувачем даних та обробки помилок.