

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 РОЗГАЛУЖЕНИХ ТА  
ІТЕРАЦІЙНИХ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

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

ВИКОНАВ  
студент академічної групи  
КІ-23  
Яровець Т. О.

ПЕРЕВІРИВ  
ст. викладач  
кафедри кібербезпеки  
та програмного забезпечення  
Дреєва Г. М.

**Тема:** Реалізація програмних модулів розгалужених та ітераційних обчислювальних процесів

**Мета:** полягає у 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 у кросплатформовому середовищі Code::Blocks.

**Завдання:**

1. Реалізувати функції розв’язування задач 9.1–9.3 як складових статичної бібліотеки libModulesПрізвище.а (проект ModulesПрізвище, створений під час виконання лабораторної роботи №8).
2. Реалізувати програмне забезпечення розв’язування задачі 9.4 на основі функцій статичної бібліотеки libModulesПрізвище.а.


### Варіант №17

#### ВАРІАНТ 17

##### — ЗАДАЧА 9.1 —

Вхід: бал сили вітру за шкалою Бофорта.


Вихід: швидкість та характеристика вітру.

 БАЛ БОФОРТА	ШВИДКІСТЬ ВІТРУ, м/сек	ХАРАКТЕ- РИСТИКА	Продовження таблиці		
			БАЛ БОФОРТА	ШВИДКІСТЬ ВІТРУ, м/сек	ХАРАКТЕ- РИСТИКА
0	< 0.3	Штиль	7	13.9 – 17.1	Міцний
1	0.3 – 1.5	Тихий	8	17.2 – 20.7	Дуже міцний
2	1.6 – 9.4	Легкий	9	20.8 – 24.4	Шторм
3	9.4 – 5.4	Слабкий	10	24.5 – 28.4	Сильний шторм
4	5.5 – 7.9	Помірний	11	28.5 – 32.6	Шквальний шторм
5	8.0 – 10.7	Свіжий	12	≥ 32.7	Ураган (буревій)
6	10.8 – 13.8	Сильний			

##### — ЗАДАЧА 9.2 —

Вхід: розмір чоловічих шкарпеток за українською системою.

Вихід: відповідники розмірів шкарпеток у системах США та ЄС.

 ЄС	Таблиця відповідності розмірів шкарпеток (чол.)				
	37/38	39/40	41/42	43/44	45/46
Великобританія, США	8	9	10	11	12
Україна	23	25	27	29	31

##### — ЗАДАЧА 9.3 —

Вхід: натуральне число  $N$  від 0 до 7483650.

Вихід: якщо біт  $D_7$  числа  $N$  рівний 1, кількість двійкових нулів у ньому, інакше — суму двійкових одиниць\*.

\*під час підрахунку кількості бінарних 0 або 1 рекомендовано використати тернарний оператор « ? : ».

##### — ЗАДАЧА 9.4 —

За введеним користувачем символом “u” викликається `s_calculation()`, “o” — функція задачі 9.1, “p” — функція задачі 9.2, “a” — 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 “s” або “S”, відбувається вихід з програми, інакше — виконання програми повторюється.



У випадку, якщо параметром `i` або результатом функції є дані нестандартного типу (наприклад, складового), то такий тип варто реалізувати у заголовковому файлі.

**Аналіз задачі 9.1:**

Вх.: bal – типу int

Вих: «Швидкість вітру: (speed) – (characteristic)».

**Особливості:**

Числа від 0 до 12

ПОЧАТОК

1. Задати значення балу хвиль.
2. Проходження всіх кейсів та обирання одного з них.
3. Вивід тексту що в обраному кейс.

КІНЕЦЬ

**Аналіз задачі 9.2:**

Вх.: sizeUA – типу int

Вих.: «За українською системою: (sizeUA), за європейською (sizeEU), за системою Великобританія, США (sizeUSA)».

**Особливості:**

Числа 23,25,27,29,31

ПОЧАТОК

1. Задати значення розміру.
2. Проходження всіх кейсів та обирання одного з них.
3. Вивід тексту що в обраному кейс.

КІНЕЦЬ

**Аналіз задачі 9.3:**

Вх.: N - типу int

Вих.: «Кількість двійкових нулів: <equalzero>», інакше: «Кількість двійкових одиниць: <equalone>».

**Особливості:**

Числа від 0 до 7483650.

ПОЧАТОК

1. Задати значення N як параметр функції.

2. Якщо введено некоректні дані - вивести повідомлення у cout, потім - завершено роботу програми.
3. Переведення заданого числа у двійкову форму та записування у змінну рядкового типу.
4. Рахування кількості одиниць та нулів у рядку, записування їх у відповідні змінні цілого типу.
5. Перевірка чи число має 2 біти, якщо ні – виведення повідомлення у cout, потім завершення програми.
6. Обрання множинним вибором, що саме треба вивести (0 чи 1), потім виведення повідомлення у cout.

**КІНЕЦЬ**

#### **Аналіз задачі 9.4:**

Вх.: entering – типу char

Вих.: Функція за запитом користувача.

#### **Особливості:**

Введення символів, що може ввести користувач: «u», «o», «p», «a», «s», «S»- типу char. Інакше програма повторюється.

**ПОЧАТОК**

1. Вивести повідомлення про запит команди у cout.
2. Задати символ тієї функції що потрібна.
3. Множинним вибором обрати команду.
4. Виведення запиту на отримання даних, що відповідає обраній команді (Функції).
5. Якщо запит - некоректний вивести звуковий сигнал та повторити програму з самого початку, інакше - опрацювати запит.

**КІНЕЦЬ**

#### **Лістинг ModulesYarovets.h:**

```
#ifndef MODULESYAROVETS_H_INCLUDED
#define MODULESYAROVETS_H_INCLUDED

void s_calculation(int, int);
```

```

void bofort(int);
void socks(int);
void bit(int);
#endif // MODULESYAROVETS_H_INCLUDED

```

### Лістинг libModulesYarovets.a:

```

#include <iostream>
#include <cmath>
#include <iomanip>
#include <bits/stdc++.h>
#include <string>

using namespace std;

void s_calculation(int y, int z){
    float S;
    system("chcp 65001 > nul");
    if (z >= 0){
        S = pow((z + y + z),2)- 6.28*sqrt(0.33*z);
        cout << "S: " << setprecision(1) << fixed << S << endl;
    }
    else{
        cout << "Неможливо визначити, тому що корінь не може бути від'ємним!" << endl;
    }
}

void bofort(int bal){
    system("chcp 65001 > nul");
    switch (bal) {
        case 0:
            cout << "Швидкість відру: < 0.3 - Штиль" << endl;
            break;
        case 1:
            cout << "Швидкість відру: 0.3 - 1.5 - Тихий" << endl;
            break;
        case 2:
            cout << "Швидкість відру: 1.6 - 9.4 - Легкий" << endl;
            break;
        case 3:
            cout << "Швидкість відру: 9.4 - 5.4 - Слабкий" << endl;
            break;
        case 4:
            cout << "Швидкість відру: 5.5 - 7.9 - Помірний" << endl;

```

```

        break;
    case 5:
        cout << "Швидкість відру: 8.0 - 10.7 - Свіжий" << endl;
        break;
    case 6:
        cout << "Швидкість відру: 10.8 - 13.8 - Сильний" << endl;
        break;
    case 7:
        cout << "Швидкість відру: 13.9 - 17.1 - Міцний" << endl;
        break;
    case 8:
        cout << "Швидкість відру: 17.2 - 20.7 - Дуже міцний" <<
endl;

        break;
    case 9:
        cout << "Швидкість відру: 20.8 - 24.4 - Шторм" << endl;
        break;
    case 10:
        cout << "Швидкість відру: 24.5 - 28.4 - Сильний шторм"
<< endl;

        break;
    case 11:
        cout << "Швидкість відру: 28.5 - 32.6 - Шквальний шторм"
<< endl;

        break;
    case 12:
        cout << "Швидкість відру: >= 32.7 - Ураган(буревій)" <<
endl;

        break;
    default:
        cout << "Такого балу за шкалою Бофорта немає." << endl;
        break;
    }
}

void socks(int sizeUA){
    system("chcp 65001 > nul");
    switch (sizeUA) {
        case 23:
            cout << "За українською системою: 23, за європейською
37/38, за системою Великобританія, США 8" << endl;
            break;
        case 25:

```

```

        cout << "За українською системою: 25, за європейською
39/40, за системою Великобританія, США 9" << endl;
        break;
    case 27:
        cout << "За українською системою: 27, за європейською
41/42, за системою Великобританія, США 10" << endl;
        break;
    case 29:
        cout << "За українською системою: 29, за європейською
43/44, за системою Великобританія, США 11" << endl;
        break;
    case 31:
        cout << "За українською системою: 31, за європейською
45/46, за системою Великобританія, США 12" << endl;
        break;
    default:
        cout << "Такого розміру не існує." << endl;
        break;
    }
}

```

```

int bit(int N){
    system("chcp 65001 > nul");
    if (N < 0 || N > 7483650) {
        cout << "Число вийшло за заданий діапазон (0 - 7483650)" <<
endl;

        return 1;
    }
    int bin = 0, k = 1;
    while (N)
    {
        bin += (N%2) * k;
        k *= 10;
        N/=2;
    }
    string str = to_string(bin);
    int equalone = count(str.begin(), str.end(), '1');
    int equalzero = count(str.begin(), str.end(), '0');
    reverse(str.begin(), str.end());
    if (str.length() < 8) {
        cout << "Недостатньо бітів" << endl;
        return 1;
    }
}

```

```

    }
    switch (str[7]) {
        case '1':
            cout << "Кількість двійкових нулів: " << equalzero <<
endl;

            break;
        case '0':
            cout << "Сума двійкових одиниць: " << equalone << endl;
            break;
    }
    return 0;
}

```

### **Лістинг TestDriver:**

```

#include <iostream>
#include <ModulesYarovets.h>

using namespace std;

int main()
{
    system("chcp 65001 & cls");
    cout << "\n\n9.1\n\n\n\n\n" << endl;
    bofort(1);
    bofort(2);
    bofort(3);
    bofort(4);
    bofort(5);
    bofort(6);
    bofort(7);
    bofort(10);
    bofort(11);
    bofort(15);

    cout << "\n\n9.2\n\n\n\n\n" << endl;
    socks(25);
    socks(23);
    socks(29);
    socks(27);
    socks(31);
    socks(6);
    socks(8);
    socks(10);
}

```



```

        socks(18);
        socks(2);

        cout << "\n\n9.3\n\n\n\n\n" << endl;
        bit(130);
        bit(145);
        bit(150);
        bit(1000);
        bit(1);
        bit(2400);
        bit(995);
        bit(531);
        bit(231);
        bit(693);
        return 0;
    }

```

### **Лістинг Yarovets\_task:**

```

#include <iostream>
#include "ModulesYarovets.h"

using namespace std;

int main()
{
    system("chcp 65001 > nul");
    cout << "Введіть символ за який відповідає функція:" << endl;
    char entering;
    cin >> entering;
    switch (entering) {
        case 'u':
            int y;
            int z;
            cout << "Введіть значення y:";
            cin >> y;
            cout << "Введіть значення z:";
            cin >> z;
            s_calculation(y, z);
            break;
        case 'o':
            int bal;
            cout << "Введіть бал хвиль за Бофортом:" << endl;
            cin >> bal;

```

```

        bofort(bal);
        break;
    case 'p':
        int sizeUA;
        cout << "Введіть розмір шкарпеток за українською
системою:" << endl;
        cin >> sizeUA;
        socks(sizeUA);
        break;
    case 'a':
        int N;
        cout << "Введіть N:" << endl;
        cin >> N;
        bit(N);
        break;
    case 's':
        exit(1);
        break;
    case 'S':
        exit(1);
        break;
    default:
        cout << "\a\a" << endl;
}
main();
}

```

**Висновок:** Роблячи цю лабораторну роботу мені видали 4 завдання. У завданні 9.1 я проаналізував вимоги до ПЗ для завдання 9.1, і створив функцію «bofort» та додав її до статичної бібліотеки libModulesYarovets.a, також було додано до заголовкового файлу ModulesYarovets.h.

У завданні 9.2 я проаналізував вимоги до ПЗ для завдання 9.2, і створив функцію «socks» та додав її до статичної бібліотеки libModulesYarovets.a, також було додано до заголовкового файлу ModulesYarovets.h.

У завданні 9.3 я проаналізував вимоги до ПЗ для завдання 9.3, і створив функцію «bit» та додав її до статичної бібліотеки libModulesYarovets.a, також було додано до заголовкового файлу ModulesYarovets.h.

У завданні 9.4 я проаналізував вимоги до ПЗ для завдання 9.4, і створив консольний додаток «Yarovets\_task» до якого імпортував статичну бібліотеку libModulesYarovets.a, яка містить у собі функції s\_calculation, bofort, socks а також bit також заголовковий файл ModulesYarovets.h, що містить прототип цих же функцій.

Були створені 3 тест-сюти з 10 тестовими випадками під кожен функцію, а також один загальний тест-сют для ПЗ «Yarovets\_task».

### **Контрольні запитання:**

1. Яким оператором C/C++ можливо повноцінно замінити тернарний оператор? Відповідь обґрунтуйте й доведіть експериментально.

Повноцінно замінивши тернарний оператор можна використовувати if else. Наприклад:

```
#include <iostream>

using namespace std;

int main() {
    int a = 16;
    int b = 100;
    int ternary = (a > b) ? a : b;
    cout << "Результат тернарного оператора: " << ternary << endl;
    // Заміна тернарного оператора
    int unternary;
    if (a > b)
    {
        unternary = a;
    } else
    {
        unternary = b;
    }
    cout << "Результат заміненого оператора: " << unternary << endl;
    return 0;
}
```

2. Що в програмуванні розуміють під пріоритетом виконання операцій та асоціативністю?

Пріоритет виконання операції- це коли визначається яка з операцій буде виконуватись першою, другою і т.д., а асоціативність- це коли визначається в якому саме порядку буде використовуватись оператор з однаковим пріоритетом.

3. Яку область видимості мають змінні, оголошені в тілі циклу або умови (вибору)? Відповідь обґрунтуйте та доведіть експериментально.

Вони мають локальну область видимості. Наприклад:

```
#include <iostream>

using namespace std;

int main()
{
    if (true) {
        int localnyi = 1;
    }
    cout << localnyi << endl;
    return 0;
}
```

4. Якою є асоціативність операцій арифметичних, логічних, логічних порозрядних, інкрементна, декрементна, тернарної та порівняння в мові програмування C/C++?

Арифметичні оператори- асоціативність зліва направо

Логічні оператори- асоціативність зліва направо

Логічні порозрядні оператори- асоціативність зліва направо

Інкремент/Декремент- асоціативність: справа наліво

Тернарний оператор- асоціативність зліва направо

Оператори порівняння- асоціативність зліва направо

У мові програмування C/C++ операції мають однакову асоціативність

5. Перелічіть випадки, за яких доцільно використовувати тернарний оператор C/C++, й наведіть приклад його запису.

Використовується для заміни оператора if else. Наприклад:

За допомогою if else:

```
if (x > y)
    temp = x;
else
    temp = y;
```

За допомогою тернарного оператора:

```
temp = (x > y) ? x : y;
```

6. Яке значення міститиме змінна cnt після виконання наступної інструкції: cnt--; ?

На одиницю менше ніж було задано.

7. Чим константна змінна, оголошена за допомогою кваліфікатора типів const, відрізняється від змінної? Сформулюйте правило, коли змінну варто оголошувати саме константною.

Відрізняється тим, що значення константної змінної не змінити після її ініціалізації.

Змінну варто оголосити тільки тоді, значення константи не повинно змінюватись під час виконання програми.

8. Яких типів можуть бути операнди логічних операторів C/C++? Вони можуть бути або типу bool або числового типу.

9. Яке значення міститиме змінна cnt при: bool cnt = !!0; ?

Буде містити значення 0, тому що перед ним подвійне заперечення.

10. Сформулюйте правило запису виразу ініціалізації у циклах з параметром (for) C++.

Вираз ініціалізації повинен містити змінну та її тип, якщо її не оголосили раніше(int i=0;). Якщо вона була оголошена раніше, то можливо просто надати у цикл чи перевизначити її. Наприклад:

```
for (i ; умова ; інкремент/декремент)
for (i = 0 ; умова ; інкремент/декремент)
```