

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”

Виконав
Студент академічної групи КН-22
Осадчий В.К

Перевірів
Викладач кафедри кібербезпеки
Та програмного забезпечення
О.СОБІНОВ

Мета роботи 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.

ВАРІАНТ 3

— ЗАДАЧА 9.1 —

Вхід: бал сили вітру за шкалою Бофорта.

Вихід: швидкість та характеристика вітру.



БАЛ БОФОРТА	ШВИДКІСТЬ ВІТРУ, м/сек	ХАРАКТЕРИСТИКА
0	< 0.3	Штиль
1	0.3 – 1.5	Тихий
2	1.6 – 9.4	Легкий
3	9.4 – 5.4	Слабкий
4	5.5 – 7.9	Помірний
5	8.0 – 10.7	Свіжий
6	10.8 – 13.8	Сильний

Продовження таблиці

БАЛ БОФОРТА	ШВИДКІСТЬ ВІТРУ, м/сек	ХАРАКТЕРИСТИКА
7	13.9 – 17.1	Міцний
8	17.2 – 20.7	Дуже міцний
9	20.8 – 24.4	Шторм
10	24.5 – 28.4	Сильний шторм
11	28.5 – 32.6	Шквальний шторм
12	≥ 32.7	Ураган (буревій)

— ЗАДАЧА 9.2 —

Вхід: цілі числа n та m .

Вихід: n^m і m^n , а також всі непарні числа у діапазоні від n до m .

— ЗАДАЧА 9.3 —

Вхід: натуральне число N від 0 до 80000.

Вихід: якщо біт D_2 числа N рівний 0, кількість двійкових нулів у ньому, інакше — кількість 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**.

Головні умови для алгоритмів кожної функції

1-Все створюємо у вигляді функцій.

Алгоритм виконання задачі 9.1

- 1-Підключаємо потрібні нам бібліотеки для зручної роботи;
- 2-Створюємо змінну типу `int x`;
- 3-Після цього ми вводимо наше число і якщо воно в діапазоні від 0-12. то виводимо вже конкретну інформацію звідти:

0-Швидкість вітру:<0.3.Характеристика:Штиль.

1-Швидкість вітру:0.3-1.5.Характеристика:Тихий.

2-Швидкість вітру:1.6-9.4.Характеристика:Легкий.

3-Швидкість вітру:9.4-5.4.Характеристика:Слабкий.

4-Швидкість вітру:5.5-7.9.Характеристика:Помірний.

5-Швидкість вітру:8.0-10.7.Характеристика:Свіжий.

6-Швидкість вітру:10.8-13.8.Характеристика:Сильний.

7-Швидкість вітру:13.9-17.1.Характеристика:Міцний.

8-Швидкість вітру:17.2-20.7.Характеристика:Дуже міцний.

9-Швидкість вітру:20.8-24.4.Характеристика:Шторм.

10-Швидкість вітру:24.5-28.4.Характеристика:Сильний шторм.

11-Швидкість вітру:28.5-32.6.Характеристика:Квальний шторм.

12-Швидкість вітру:>=32.7.Характеристика:Ураган(буревій).

- 4-Після чого відбувається кінець програми.

Алгоритм виконання задачі 9.2

- 1-Підключаємо всі потрібні для роботи бібліотеки;
- 2-Створюємо дві змінні:`n,n`;
- 3-Після чого, завдяки функції `pow` та `cout` ми виводимо `n` в степені `m` та навпаки;
- 4-Після чого, завдяки циклу `for` ми створюємо діапазон чисел від `n` до `m`;
- 5-Перевіряємо чи при діленні числа на два остача не дорівнює 0-виводимо це число на екран;
- 6-Після чого відбувається кінець програми.

Алгоритм виконання задачі 9.3

- 1-Підключаємо всі потрібні бібліотеки для роботи;
- 2-Створюємо змінну типу `int N`;
- 3-Вводимо значення цього числа;
- 4-Перетворюємо наше число на двоїчний код;
- 5-Створюємо змінну типу `bool secondbit`, що приймає значення другого елемента в двійковій системі;
- 6-Створюємо змінну типу `int counter`, яка дорівнює 0;
- 7-Робимо перевірку чи `secondbit` дорівнює 0 і у випадку вірності-ми додаємо 1 до `counter`(значення 0 в двійковому коді), у випадку іншого значення, `counter` вже буде приймати значення кількості 1.
- 8-Після чого відбувається кінець програми.

Алгоритм виконання задачі 9.4

1-Підключаємо саму бібліотеку, що містять розв'язки задач 9.1, 9.2, 9.2 та функції `s_calculation` з попередньої лабораторної роботи.

2-створюємо змінні типу `int first, second, third;`

2-Створюємо змінну типу `char action;`

3-Після чого передаємо їй значення та робимо перевірку, якщо те значення відповідає потребі:

u-Вводимо значення `first, second, third` та активуємо функцію `s_calculation()`, що приймає ті значення тих змінних;

o-Вводимо значення для `first, second` т активуємо функцію `windbyBafort()`, що приймає значення тих змінних;

p-Вводимо значення для `first, second` та активуємо функцію `elnm()`, що приймає значення тих змінних;

a-Вводимо значення для `first` та активуємо функцію `bitN()`, що приймає значення тих змінних;

4-У випадку введення не того елемента-видаємо звуковий сигнал помилки та робимо повторне прохання ввести якусь букву.Робимо перевірку:

S або s-Програма перестає працювати;

Любий елемент-Активуємо знову цю функцію;

5-Після чого програма закінчується.

Висновок

Мета цієї роботи полягала у набутті ґрунтовних вмінь і практичних навичок застосування теоретичних положень методології модульного програмування, реалізації метода функціональної декомпозиції задач(самі завдання були надані викладачем в репозиторії [odorenskyi/Osadchy-Vladyslav-KN22](#)), метода модульного (блочного) тестування, представлення мовою програмування C++ даних скалярних типів, арифметичних і логічних операцій, потокового введення й виведення інформації, представлення мовою програмування C++ методів розгалуження, циклів, розроблення програмних модулів та засобів у кросплатформовому середовищі `Code::Blocks` (GNU GCC Compiler).

У цій роботі я набув практичних навичок як правильно слід створювати розгалуження, цикли модулів і вписувати в них потрібні функції з потрібними нам умовами, на мові C++ у компіляторі `Code::Blocks` (GNU GCC Compiler), для правильної подальшої роботи з ними роботи, як було написано у завданні, та для тестування кожної з цих функцій для розуміння і вияснення чи правильно ця функція працює і при знаходженні певних помилок-виконував налагодження програмного застосунку для подальшої роботи з ним.Вивчив як правильно слід створювати власні статичні бібліотеки(модулі),на мові C++ в компіляторі `Code::Blocks` (GNU GCC Compiler), використовуючи різноманітні інтернет ресурси та джерела, які були надані, як варіант для розгляду, від викладачів даного предмету.Навчився як правильно реалізовувати цикли, розгалуження, ітераційних обчислювальних процесів для подальшої праці з ними та їхнього тестування.Всі помилки які виникали під час підключення файлу бібліотек до інших файлів, для подальшої роботи і тестування, були виявленні і був процес подальшого налагодження цього процесу.

Були створені всі необхідні папки для реалізації цієї лабораторної роботи:prj(для збереження всіх основних папок реалізації лабораторної роботи номер 9 мовою

програмування C++), TestSuite(де були збережені програми для перевірки кожного блоку завдання та збережені копії самих тест-с'ютів), Report(куди саме був збережений звіт), Software(куди саме були збережені exe файли тестування).

Для праці із завданням 9.1(саме завдання було представлено вище) був виконан процес дослідження всіх умов та потреб для подальшого виконання цієї задачі у мові програмування C++ у компіляторі Code::Blocks (GNU GCC Compiler), був виконаний процес дослідження найзручнішого методу реалізації цієї задачі.

Всі подальші дії для реалізації цієї функції :

1-Була створена папка ModulesOsadchiy(та збережений у папку rjr) для реалізації самої роботи;

2-Була створена бібліотека та Headers для реалізації цієї задачі(в ній будуть ще описані завдання 9.1, 9.2);

3-Був створений TestDriver для перевірки підключення самої бібліотеки та перевірки її працездатності;

4-Для більш коректного тестування був створений окремий Testdriver 9.1 для перевірки працездатності бібліотеки у різних форматах праці.

5-Всі результати тестування були задокументовані в тест-с'юті.

Для праці із завданням 9.2(саме завдання було представлено вище) був виконан процес дослідження всіх умов та потреб для подальшого виконання цієї задачі у мові програмування C++ у компіляторі Code::Blocks (GNU GCC Compiler), був виконаний процес дослідження найзручнішого методу реалізації цієї задачі.

Всі подальші дії для реалізації цієї функції:

1-Розв'язок цієї задачі був збережений у папці ModulesOsadchiy(та збережений у папку rjr) для реалізації самої роботи;

2-Був створений TestDriver для перевірки підключення самої бібліотеки та перевірки її працездатності;

3-Для більш коректного тестування був створений окремий Testdriver 9.2 для перевірки працездатності функції бібліотеки у різних форматах праці.

Для праці із завданням 9.3(саме завдання було представлено вище) був виконан процес дослідження всіх умов та потреб для подальшого виконання цієї задачі у мові програмування C++ у компіляторі Code::Blocks (GNU GCC Compiler), був виконаний процес дослідження найзручнішого методу реалізації цієї задачі.

Всі подальші дії для реалізації цієї програми:

1-Розв'язок цієї задачі був збережений у папці ModulesOsadchiy(та збережений у папку rjr) для реалізації самої роботи;

2-Був створений TestDriver для перевірки підключення самої бібліотеки та перевірки її працездатності;

3-Для більш коректного тестування був створений окремий Testdriver 9.3 для перевірки працездатності функції бібліотеки у різних форматах праці.

Для праці із завданням 9.4(саме завдання було представлено вище) був виконан процес дослідження всіх умов та потреб для подальшого виконання цієї задачі у мові програмування C++ у компіляторі Code::Blocks (GNU GCC Compiler), був виконаний процес дослідження найзручнішого методу реалізації цієї задачі.

Всі подальші дії для реалізації цієї програми:

1-Була створена папка ModulesOsadchiy(та збережений у папку rjr) для реалізації самої роботи;

2-Була підключена бібліотека з розв'язками задач 9.1, 9.2, 9.3 та з підключеним модулем, де була реалізована функція s_calculation();

3-Був створений тест-сьют для тестування самої програми.
Від себе хочу додати, що виконання цієї роботи було дуже цікавим і повчальним.

Додатки для завдання 9.1

```
#include "ModulesOsadchiy.h"
#include <iostream>
#include <locale>
#include <iomanip>
#include <cmath>
#include <bitset>
#include <main.h>
using namespace std;
int windbyBafort(int x)
{
    setlocale(LC_ALL,"ukr");
    if(x==0)
        cout<<"Швидкість вітру:<0.3.Характеристика:Штиль."<<endl;
    if(x==1)
        cout<<"Швидкість вітру:0.3-1.5.Характеристика:Тихий."<<endl;
    if(x==2)
        cout<<"Швидкість вітру:1.6-9.4.Характеристика:Легкий."<<endl;
    if(x==3)
        cout<<"Швидкість вітру:9.4-5.4.Характеристика:Слабкий."<<endl;
    if(x==4)
        cout<<"Швидкість вітру:5.5-7.9.Характеристика:Помірний."<<endl;
    if(x==5)
        cout<<"Швидкість вітру:8.0-10.7.Характеристика:Свіжий."<<endl;
    if(x==6)
        cout<<"Швидкість вітру:10.8-13.8.Характеристика:Сильний."<<endl;
    if(x==7)
        cout<<"Швидкість вітру:13.9-17.1.Характеристика:Міцний."<<endl;
    if(x==8)
        cout<<"Швидкість вітру:17.2-20.7.Характеристика:Дуже міцний."<<endl;
    if(x==9)
        cout<<"Швидкість вітру:20.8-24.4.Характеристика:Шторм."<<endl;
    if(x==10)
        cout<<"Швидкість вітру:24.5-28.4.Характеристика:Сильний шторм."<<endl;
    if(x==11)
        cout<<"Швидкість вітру:28.5-32.6.Характеристика:Квальний шторм."<<endl;
    if(x==12)
        cout<<"Швидкість вітру:>=32.7.Характеристика:Ураган(буревій)."<<endl;

    return 0;
}
```

Лістинг Testdriver 9.1

```
#include <iostream>
#include <ModulesOsadchiy.h>
```

```
using namespace std;
```

```
int main()
{
    windbyBafort(0);
    windbyBafort(1);
    windbyBafort(2);
    windbyBafort(3);
    windbyBafort(4);
    windbyBafort(5);
    windbyBafort(6);
    windbyBafort(7);
    windbyBafort(8);
    windbyBafort(9);
    windbyBafort(10);
    windbyBafort(11);
    windbyBafort(12);

    return 0;
}
```

Тест-сьют

Назва тестового набору Test Suite Description	TestSuit 9.1
Name of Project / Software	Osadchyi-Vladyslav-KN22/lab09/TestSuit/TestSuit 9.1
Рівень тестування Level of Testing	Модульне тестування
Автор тест-сьюта Test Suite Author	Осадчий Владислав Костянтинович
Виконавець Implementer	Осадчий Владислав Костянтинович

Action	Expected Result	Test Result(passed/failed/blocked)
windbyBafort(0)	Швидкість вітру:<0.3.Характеристика: Штиль.	passed
windbyBafort(1)	Швидкість вітру:0.3-1.5.Характеристи	passed

	ка:Тихий.	
windbyBafort(2)	Швидкість вітру:1.6-9.4.Характеристика:Легкий.	passed
windbyBafort(3)	Швидкість вітру:9.4-5.4.Характеристика:Слабкий.	passed
windbyBafort(4)	Швидкість вітру:5.5-7.9.Характеристика:Помірний.	passed
windbyBafort(5)	Швидкість вітру:8.0-10.7.Характеристика:Свіжий.	passed
windbyBafort(6)	Швидкість вітру:10.8-13.8.Характеристика:Сильний.	passed
windbyBafort(7)	Швидкість вітру:13.9-17.1.Характеристика:Міцний.	passed
windbyBafort(8)	Швидкість вітру:17.2-20.7.Характеристика:Дуже міцний.	passed
windbyBafort(9)	Швидкість вітру:20.8-24.4.Характеристика:Шторм.	passed
windbyBafort(10)	Швидкість вітру:24.5-28.4.Характеристика:Сильний шторм.	passed
windbyBafort(11)	Швидкість вітру:28.5-32.6.Характеристика:Квальний шторм.	passed
windbyBafort(12)	Швидкість вітру:>=32.7.Характеристика:Ураган(буревій).	passed
windbyBafort(якісь інші знаки, окрім потрібних чисел від 0 до 12)	Error	passed

Додатки для завдання 9.2

```
int elnm(int n, int m)
{ setlocale(LC_ALL,"ukr");
  cout<<"n:"<<pow(n,m)<<endl;
  cout<<"m:"<<pow(m,n)<<endl;
```



```

if(n>=m)
    cout<<"Наступна функція не буде працювати."<<endl;
else{
    for(int i=n;i<=m;i++)
    {
        if(i%2!=0)
            cout<<i<<" ";
    }
    cout<<endl;
}
return 0;
}

```

Лістинг Testdriver 9.2

```

#include <iostream>
#include <ModulesOsadchiy.h>
using namespace std;

int main()
{
    elnm(1,1);
    cout<<"====="<<endl;
    elnm(2,2);
    cout<<"====="<<endl;
    elnm(3,2);
    cout<<"====="<<endl;
    elnm(4,4);
    cout<<"====="<<endl;
    elnm(1,5);
    cout<<"====="<<endl;
    elnm(1,11);
    cout<<"====="<<endl;
    elnm(1,19);
    cout<<"====="<<endl;
    elnm(2,15);
    cout<<"====="<<endl;
    elnm(2,18);
    cout<<"====="<<endl;
    elnm(0,40);
    cout<<"====="<<endl;

    return 0;
}

```

Тест-сьют

Назва тестового набору Test Suite Description	TestSuit 9.2
Name of Project / Software	Osadchyi-Vladyslav-KN22/lab09/TestSuit/TestSuit 9.2
Рівень тестування Level of Testing	Модульне тестування
Автор тест-сьюта Test Suite Author	Осадчий Владислав Костянтинович
Виконавець Implementer	Осадчий Владислав Костянтинович

Action	Expected Result	Test Result(passed/failed/blocked)
elnm(1,1)	n:1 m:1 Наступна функція не буде працювати.	passed
elnm(2,2)	n:4 m:4 Наступна функція не буде працювати.	passed
elnm(3,2)	n:9 m:8 Наступна функція не буде працювати.	passed
elnm(4,4)	n:256 m:256 Наступна функція не буде працювати.	passed
elnm(1,5)	n:1 m:5 1 3 5	passed
elnm(1,11)	n:1 m:11 1 3 5 7 9 11	passed

eInm(1,19)	n1 m:19 1 3 5 7 9 11 13 15 17 19	passed
eInm(2,15)	n:32768 m:225 3 5 7 9 11 13 15	passed
eInm(2,18)	n:262144 m:324 3 5 7 9 11 13 15 17	passed
eInm(0,40)	n:0 m:1 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39	passed
eInm(інші символи, окрім цифр)	Error	passed

Додатки для завдання 9.3

```
int bitN(int N)
{
    bitset<16> binary(N);
    cout<<binary.to_string()<<endl;
    bool secondbit=((N>>1)&1);
    int counter=0;
    if(secondbit=0){
        counter=binary.count()-1;
        cout<<counter<<endl;
    }
    else{
        counter=binary.size()-binary.count();
        cout<<counter<<endl;
    }
    return counter;
}
```

Лістинг Testdriver 9.3

```
#include <iostream>
#include <ModulesOsadchiy.h>
using namespace std;
void testdriver(void){
    int test1=bitN(1);
    if(test1==15)
```

```

        cout<<"Test 1 is correct"<<endl;
int test2=bitN(2);
    if(test2==15)
        cout<<"Test 2 is correct"<<endl;
int test3=bitN(3);
    if(test3==14)
        cout<<"Test 3 is correct"<<endl;
int test4=bitN(4);
    if(test4==15)
        cout<<"Test 4 is correct"<<endl;
int test5=bitN(5);
    if(test5==14)
        cout<<"Test 5 is correct"<<endl;
int test6=bitN(40);
    if(test5==14)
        cout<<"Test 6 is correct"<<endl;
int test7=bitN(57);
    if(test7==12)
        cout<<"Test 7 is correct"<<endl;
int test8=bitN(200);
    if(test8==13)
        cout<<"Test 8 is correct"<<endl;
int test9=bitN(356);
    if(test9==12)
        cout<<"Test 9 is correct"<<endl;
int test10=bitN(8000);
    if(test10==10)
        cout<<"Test 10 is correct"<<endl;
    }

```

```

int main()
{   testdriver();

    return 0;
}

```

Тест-сьют

Назва тестового набору Test Suite Description	TestSuit 9.3
Name of Project / Software	Osadchyi-Vladyslav-KN22/lab09/TestSuit/TestSuit 9.2

Рівень тестування Level of Testing	Модульне тестування
Автор тест-сьюта Test Suite Author	Осадчий Владислав Костянтинович
Виконавець Implementer	Осадчий Владислав Костянтинович

Action	Expected Result	Test Result(passed/failed/blocked)
bitN(якісь символи. що не є числом)	Error	passed
bitN(1)	Test 1 is correct	passed
bitN(2)	Test 2 is correct	passed
bitN(3)	Test 3 is correct	passed
bitN(4)	Test 4 is correct	passed
bitN(5)	Test 5 is correct	passed
bitN(40)	Test 6 is correct	passed
bitN(57)	Test 7 is correct	passed
bitN(200)	Test 8 is correct	passed
bitN(356)	Test 9 is correct	passed
bitN(8000)	Test 10 is correct	passed

Додатки для завдання 9.4

```
#include <iostream>
#include <ModulesOsadchiy.h>
#include <main.h>
```

```

using namespace std;
void mainmenu(){
    int first,second,third;
    char action;
    cout<<"Enter your action"<<endl;
    cin>>action;
    if(action=='u')
    { cout<<"Enter your x:"<<endl;
      cin>>first;
      cout<<"Enter your y:"<<endl;
      cin>>second;
      cout<<"Enter your z:"<<endl;
      cin>>third;
      cout<<s_calculation(first,second,third)<<endl;
    }
    if(action=='o')
    {
        cout<<"Enter your number:"<<endl;
        cin>>first;
        windbyBafort(first);

    }
    if(action=='p')
    {
        cout<<"Enter your n:"<<endl;
        cin>>first;
        cout<<"Enter your m:"<<endl;
        cin>>second;
        elnm(first, second);
    }
    if(action=='a')
    {
        cout<<"Enter your N:"<<endl;
        cin>>first;
        bitN(first);
    }
    else{
        cout<<"\a"<<endl;
        cout<<"Enter new action: "<<endl;
        cin>>action;
        if(action=='s'||action=='S')
        {
            exit(0);
        }
        else{
            mainmenu();
        }
    }
}

```

```

}
int main()
{  mainmenu();
   return 0;
}

```

Тест-сьют

Назва тестового набору Test Suite Description	TestSuit 9.4
Name of Project / Software	Osadchyi-Vladyslav-KN22/lab09/TestSuit/TestSuit 9.4
Рівень тестування Level of Testing	Модульне тестування
Автор тест-сьюта Test Suite Author	Осадчий Владислав Костянтинович
Виконавець Implementer	Осадчий Владислав Костянтинович

Action	Expected Result	Test Result(passed/failed/blocked)
Спрацювання головної функції	Виконання	passed
Виконання роботи s_calculation()	Виконання	passed
Виконання роботи windbyBafort()	Виконання	passed
Виконання роботи elnm()	Виконання	passed
Виконання роботи bitN()	Виконання	passed