

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 3

На тему: «Програмування: алгоритм, програма, код. Системи числення.
Двійкова система числення. Розробка та середовище розробки програми.»
з дисципліни: «Основи програмування»

до:

Практичних Робіт до блоку № 3

Виконав:

Студент(ка) групи ІІІ-13
Яцишин Роман Олегович

Львів 2024

Тема: Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції зі змінною кількістю параметрів. Рекурсія. Вбудовані функції.

Мета: ознайомитись з роботою циклів та вкладених циклами. Розглянути функції, простір імен, перевантаження функцій, функції зі змінною кількістю параметрів, рекурсію та вбудовані функції.

Теоретичні відомості:

1. Теоретичні відомості з переліком важливих тем

- Тема №3.1 Введення в Цикли та їх Види в C++;
- Тема №3.2 Управління Виконанням Циклів;
- Тема №3.3 Вкладені Цикли;
- Тема №3.4 Основи Функцій у C++;
- Тема №3.5 Перевантаження Функцій та Простір Імен;
- Тема №3.6 Розширені Можливості Функцій;
- Тема №3.7 Вбудовані Функції в C++.

2. Індивідуальний план опрацювання теорії:

Тема №3.1 Введення в Цикли та їх Види в C++:

- Джерела інформації:
 1. https://www.w3schools.com/cpp/cpp_for_loop_foreach.asp
 2. https://www.w3schools.com/cpp/cpp_do_while_loop.asp
- Що опрацьовано:
 - Тема №3.1.1 Значення та роль циклів у програмуванні;
 - Тема №3.1.2 Огляд видів циклів: for, while, do-while;
 - Тема №3.1.3 Синтаксис та основи використання кожного типу циклу;
 - Тема №3.1.4 Приклади базових циклів для різних задач.
- Статус: Ознайомлений
- Початок опрацювання теми: 02.11
- Завершення опрацювання теми: 14.11

Тема №3.2 Управління Виконанням Циклів:

- Джерела інформації:
 1. https://www.w3schools.com/cpp/cpp_break.asp
- Що опрацьовано:
 - Тема №3.2.1 Застосування операторів break та continue;
 - Тема №3.2.2 Умови завершення циклів;
 - Тема №3.2.3 Передчасне завершення виконання циклу;
 - Тема №3.2.3 Приклади та вправи з управлінням циклами.
- Статус: Ознайомлений
- Початок опрацювання теми: 02.11
- Завершення опрацювання теми: 14.11

Тема №3.3 Вкладені Цикли:

- Джерела інформації:
 - 2. https://www.youtube.com/watch?v=_HaCgmTayMk
- Що опрацьовано:
 - Тема №3.3.1 Поняття та важливість вкладених циклів;
 - Тема №3.3.2 Реалізація вкладених циклів: приклади для різних сценаріїв;
 - Тема №3.3.3 Практичні завдання на вкладені цикли.
- Статус: Ознайомлений
- Початок опрацювання теми: 02.11
- Завершення опрацювання теми: 14.11

Тема №3.4 Основи Функцій у C++:

- Джерела інформації:
 - 3. <https://www.geeksforgeeks.org/functions-in-cpp/>
- Що опрацьовано:
 - Тема №3.4.1 Визначення та оголошення функцій;
 - Тема №3.4.2 Параметри функцій: передача за значенням і за посиланням;
 - Тема №3.4.3 Параметри за замовчуванням
 - Тема №3.4.4 Повернення значень з функцій;
 - Тема №3.4.5 Приклади створення та використання функцій.
- Статус: Ознайомлений
- Початок опрацювання теми: 02.11
- Завершення опрацювання теми: 14.11

Тема №3.5 Перевантаження Функцій та Простір Імен:

- Джерела інформації:
 - 4. <http://cpp.dp.ua/perevantazhennya-i-shablony-funktsij/>
- Що опрацьовано:
 - Тема №3.5.1 Концепція перевантаження функцій;
 - Тема №3.5.2 Правила та приклади перевантаження функцій;
 - Тема №3.5.3 Поняття та використання просторів імен;
 - Тема №3.5.4 Вкладені простори імен (C++ 17);
 - Тема №3.5.5 Роль просторів імен у організації коду.
- Статус: Ознайомлений
- Початок опрацювання теми: 02.11
- Завершення опрацювання теми: 14.11

Тема №3.6 Розширені Можливості Функцій:

- Джерела інформації:
 - 5. https://www.tutorialspoint.com/cplusplus/cpp_passing_arrays_to_functions.htm

6. <https://youtu.be/oDC208zvsdg?si=g0NDSK62LEfSKTn>

- Що опрацьовано:
 - Тема №3.6.1 Функції зі змінною кількістю параметрів (еліпсис): синтаксис та приклади;
 - Тема №3.6.2 Область видимості функції – static, extern;
 - Тема №3.6.3 Рекурсія: основи, приклади рекурсивних функцій та їх аналіз;
 - Тема №3.6.4 Передача масивів та об'єктів як параметрів;
 - Тема №3.6.5 Повернення масивів та об'єктів з функцій.
- Статус: Ознайомлений
- Початок опрацювання теми: 02.11
- Завершення опрацювання теми: 14.11

Тема №3.7 Вбудовані Функції в C++:

- Джерела інформації:
 - 7. <https://www.geeksforgeeks.org/inline-functions-cpp/>
- Що опрацьовано:
 - Тема №3.7.1 Огляд вбудованих функцій у C++;
 - Тема №3.7.2 Приклади використання стандартних функцій у програмуванні;
 - Тема №3.7.3 Роль вбудованих функцій у спрощенні коду;
 - Тема №3.7.4 Практичні завдання для розуміння вбудованих функцій
- Статус: Ознайомлений
- Початок опрацювання теми: 02.11
- Завершення опрацювання теми: 14.11

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Програмний код №1

- Метою завдання є отримання навичок у виборі й використанні операторів C; знайомство з ітераційними процесами; обчислення суми чисел кратних від 20 до 100.
- Важливо було ураховувати синтаксис циклу, крок кратності.

Програмний код №2

- Метою завдання була практика в організації ітераційних й арифметичних циклів; обчислити функцію $f(x)$, використовуючи її розклад в степеневий ряд у двох випадках.
- Важливо було ураховувати вимоги до виводу результату, синтаксис функцій.

Програмний код №3 Варіант 4 Завдання 1

- Метою завдання було знайомство з організацією функцій зі змінною кількістю параметрів та перевантаженням функцій; написати функцію `sum` зі змінною кількістю параметрів, що знаходить суму чисел типу `int` за формулою « $S=a_1*a_2+a_3*a_4+a_5*a_6+\dots$ »; написати викликаючу функцію `main`, що звертається до функції `sum` не менше трьох разів з кількістю параметрів 8, 10, 12.
- Важливим було чітко дотримання принципів роботи з функціями зі змінною кількістю параметрів.

Програмний код №3 Варіант 4 Завдання 2

- Метою завдання було написати перевантажені функції й основну програму, що їх викликає.

Програмний код №4

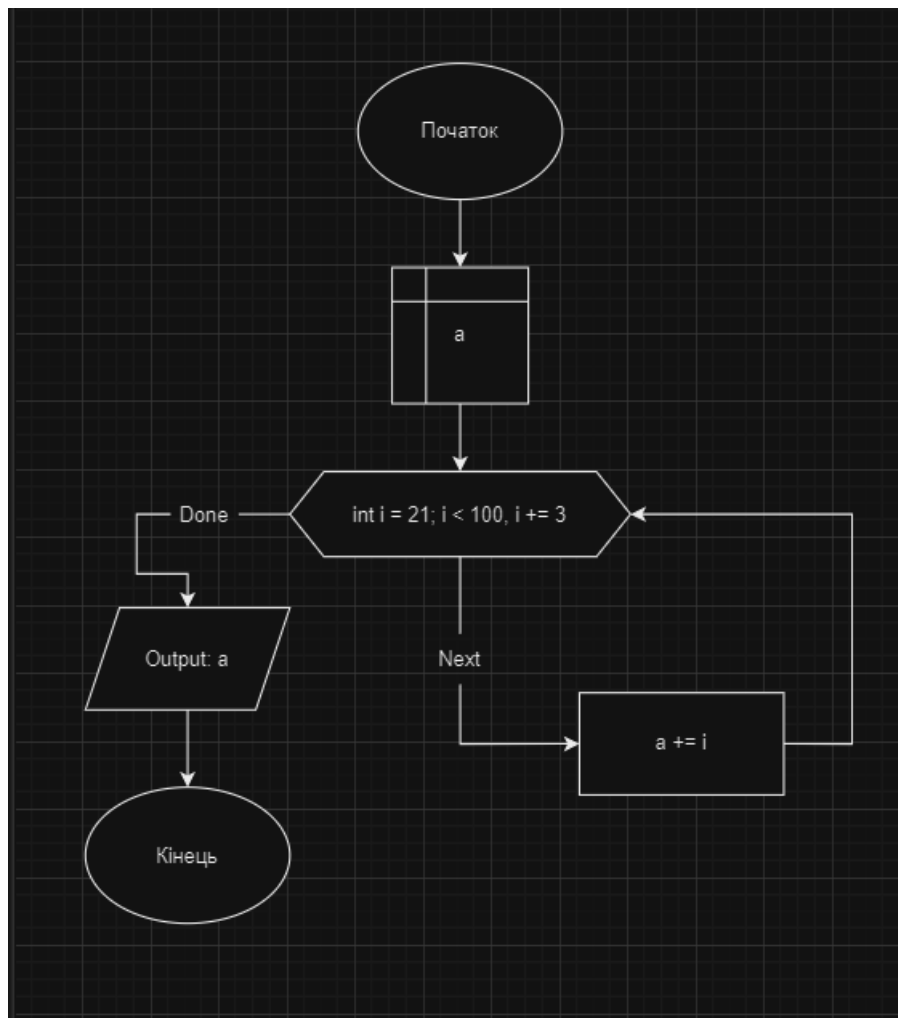
- Метою завдання було створити просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.
- Важливим було чітке дотримання вказівок до використання тих чи інших циклів.

Програмний код №5

- Метою завдання була перевірка кількості переможних партій в настільний теніс, маючи переможця кожної подачі як стрічку.

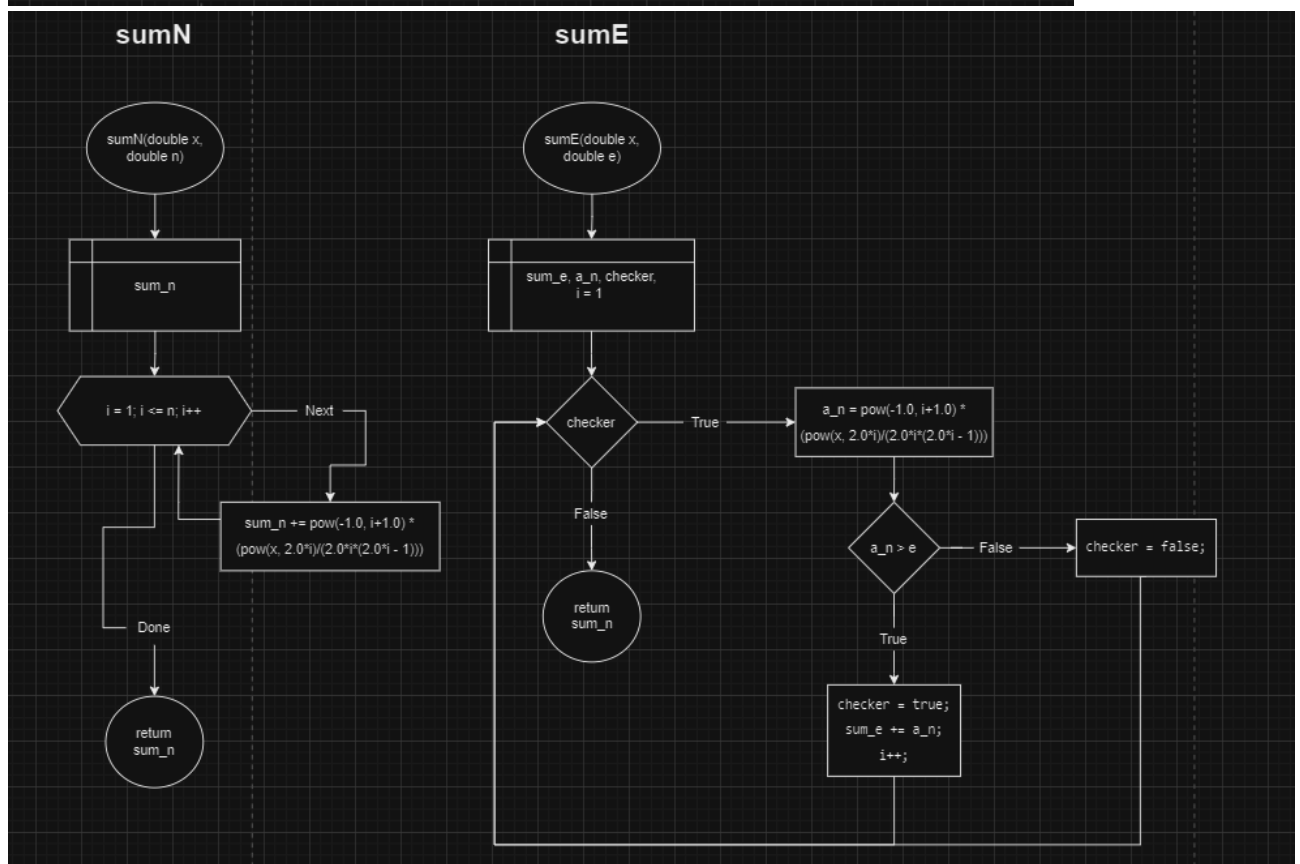
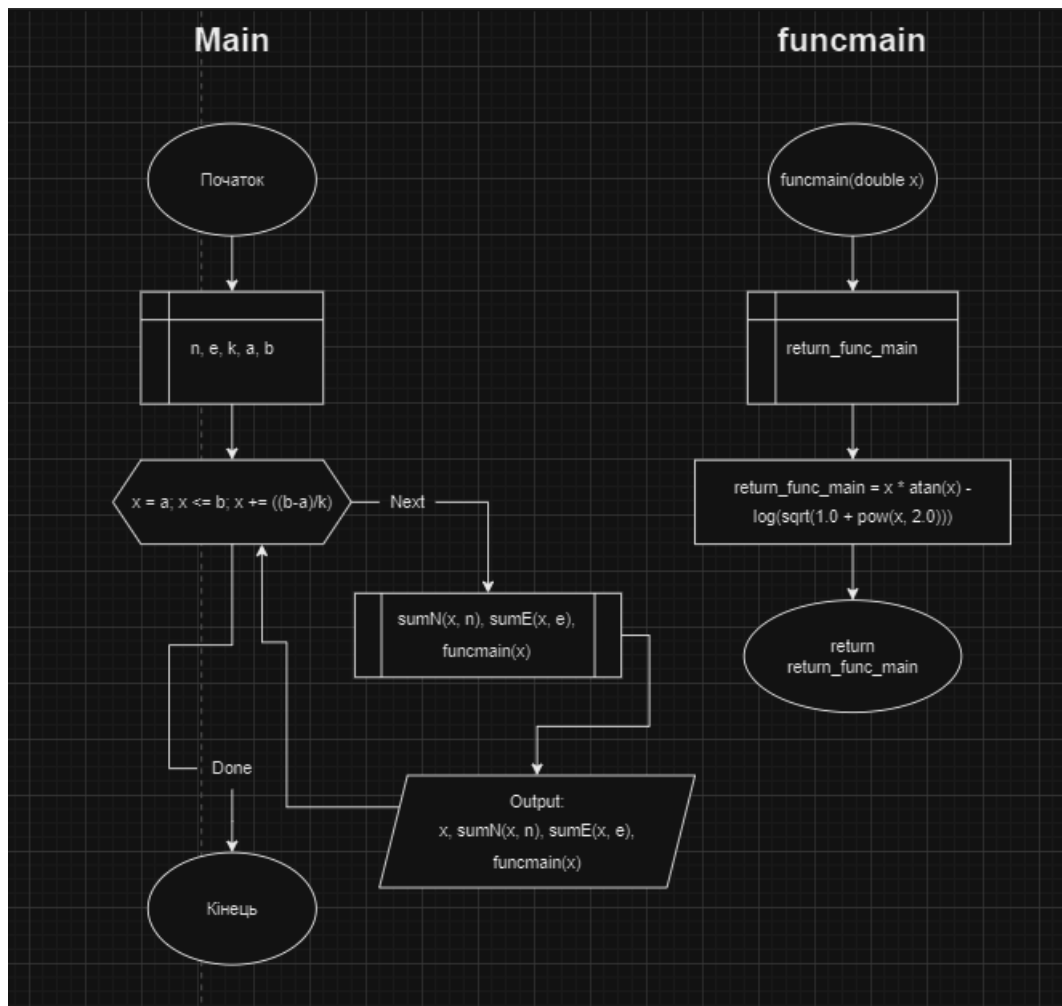
2. Дизайн та планована оцінка часу виконання завдань:

VNS Lab 2



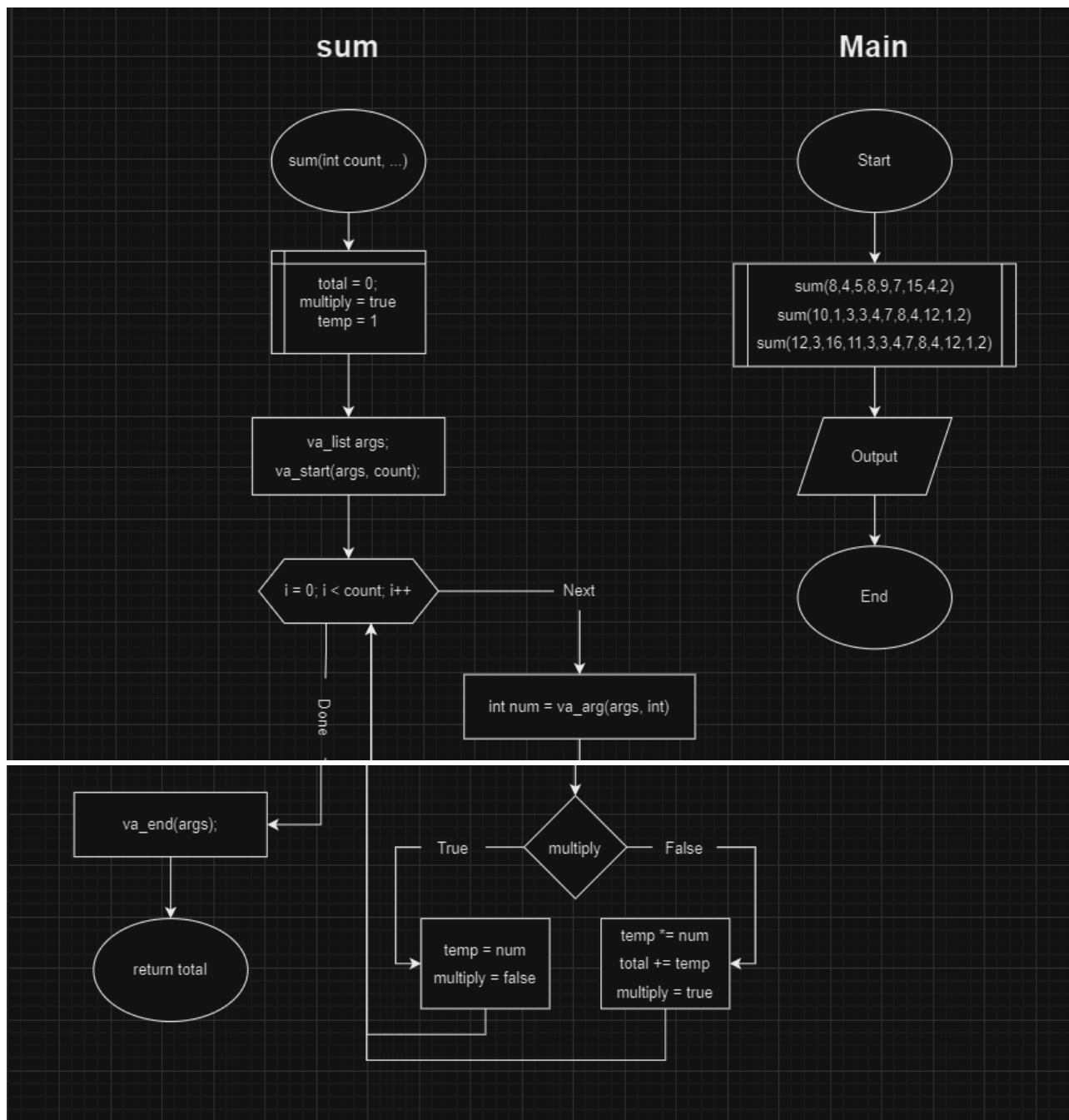
Оцінюваний час виконання завдання: 5 хв.

VNS Lab 3



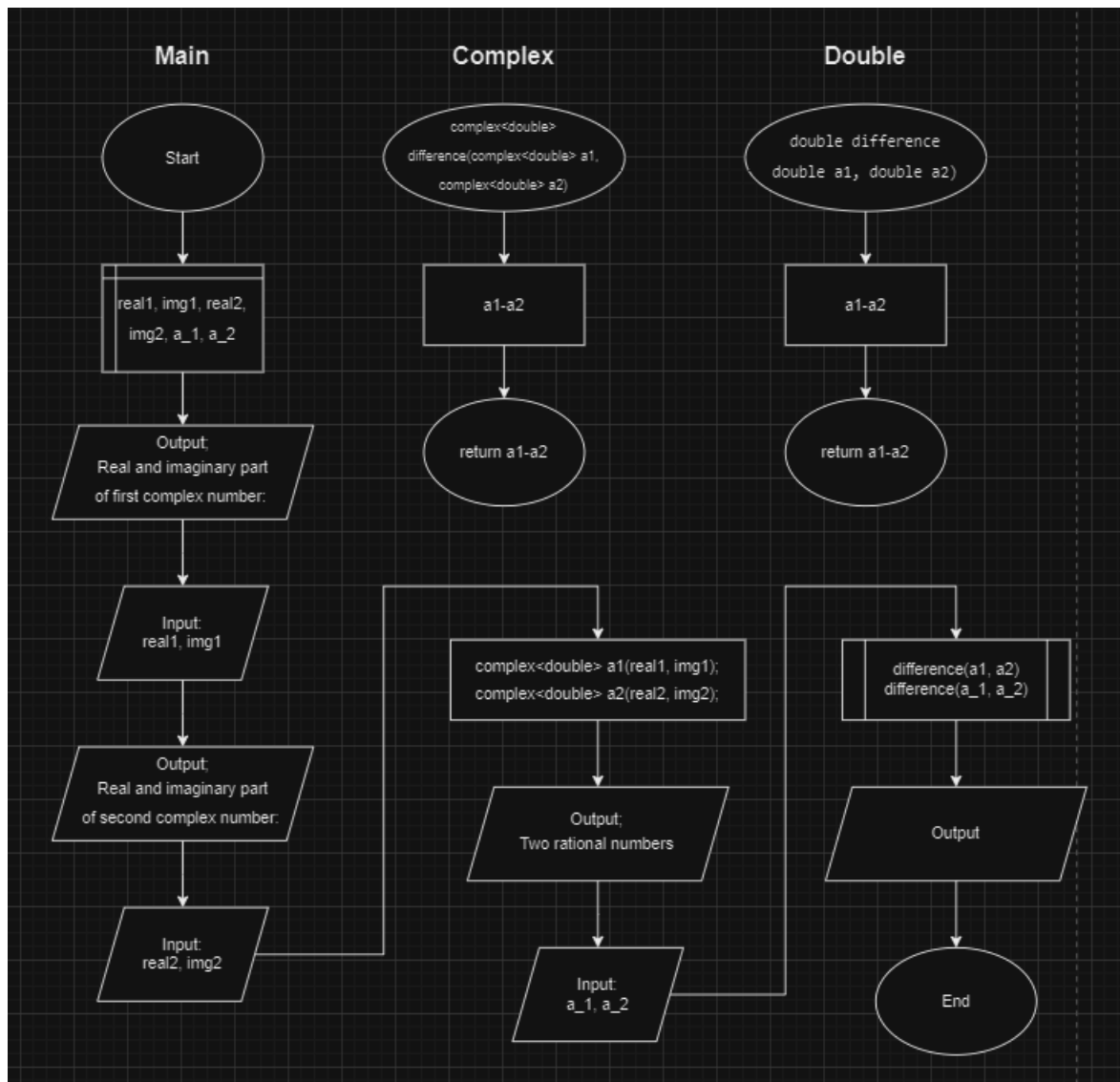
Оцінюваний час виконання: 40 хвилин.

VNS Lab 7 Task 1



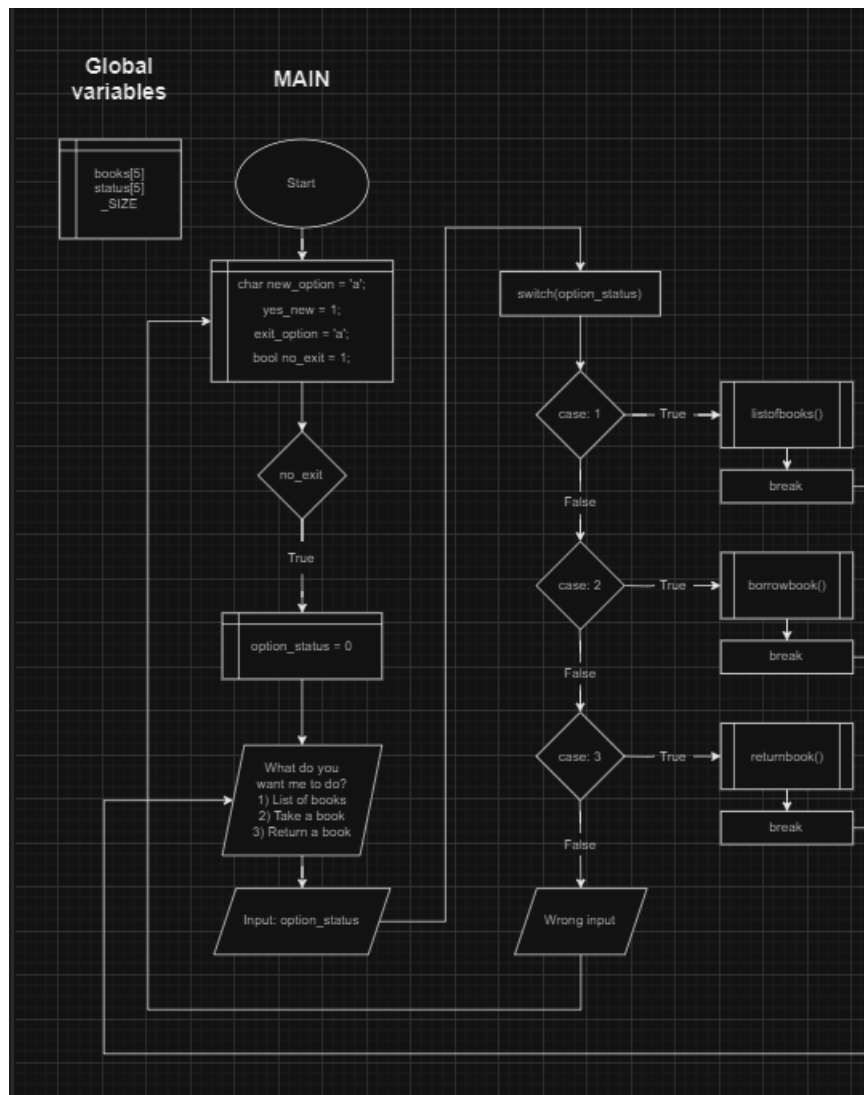
Оцінюваний час виконання: 30 хвилин.

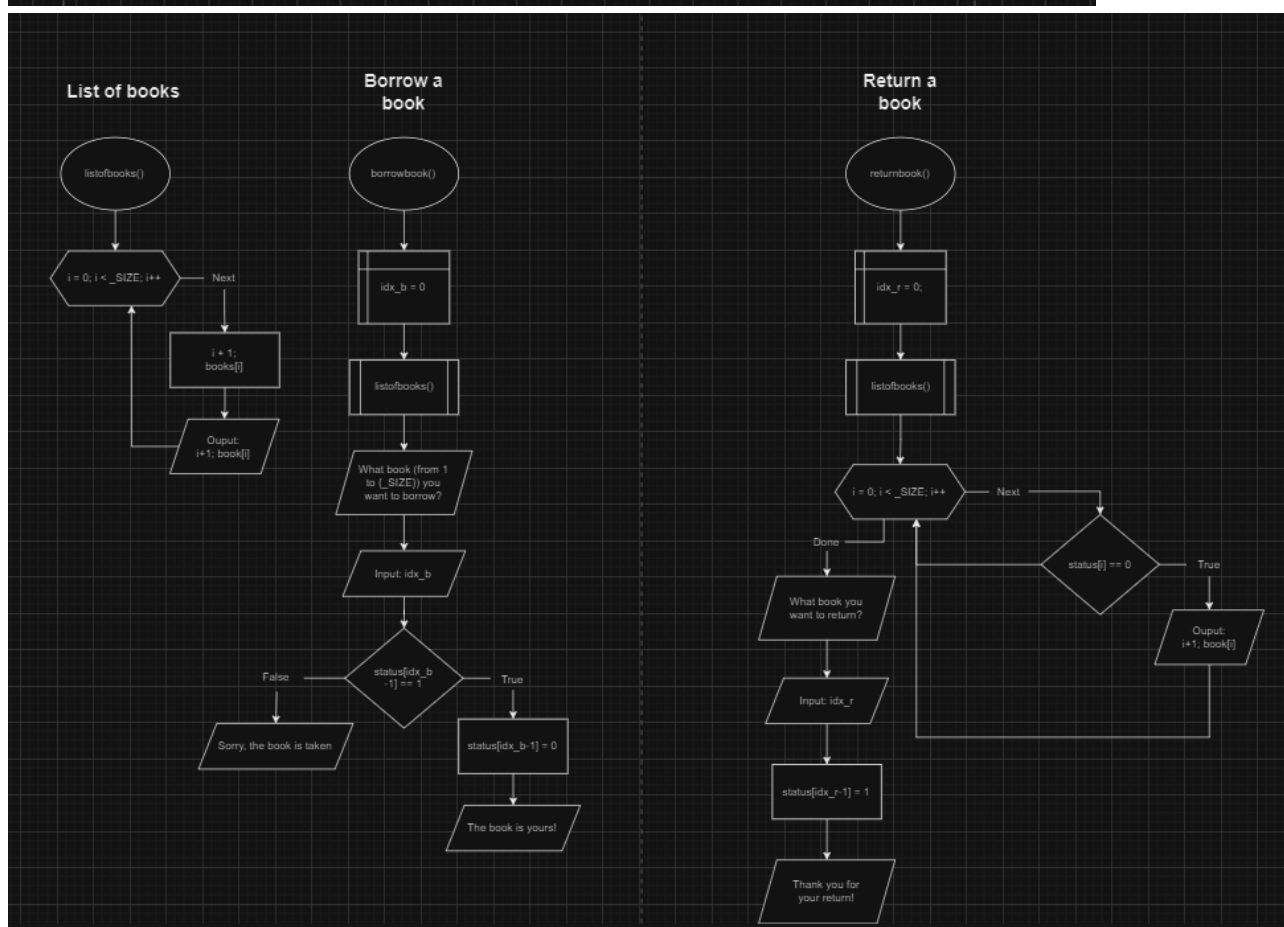
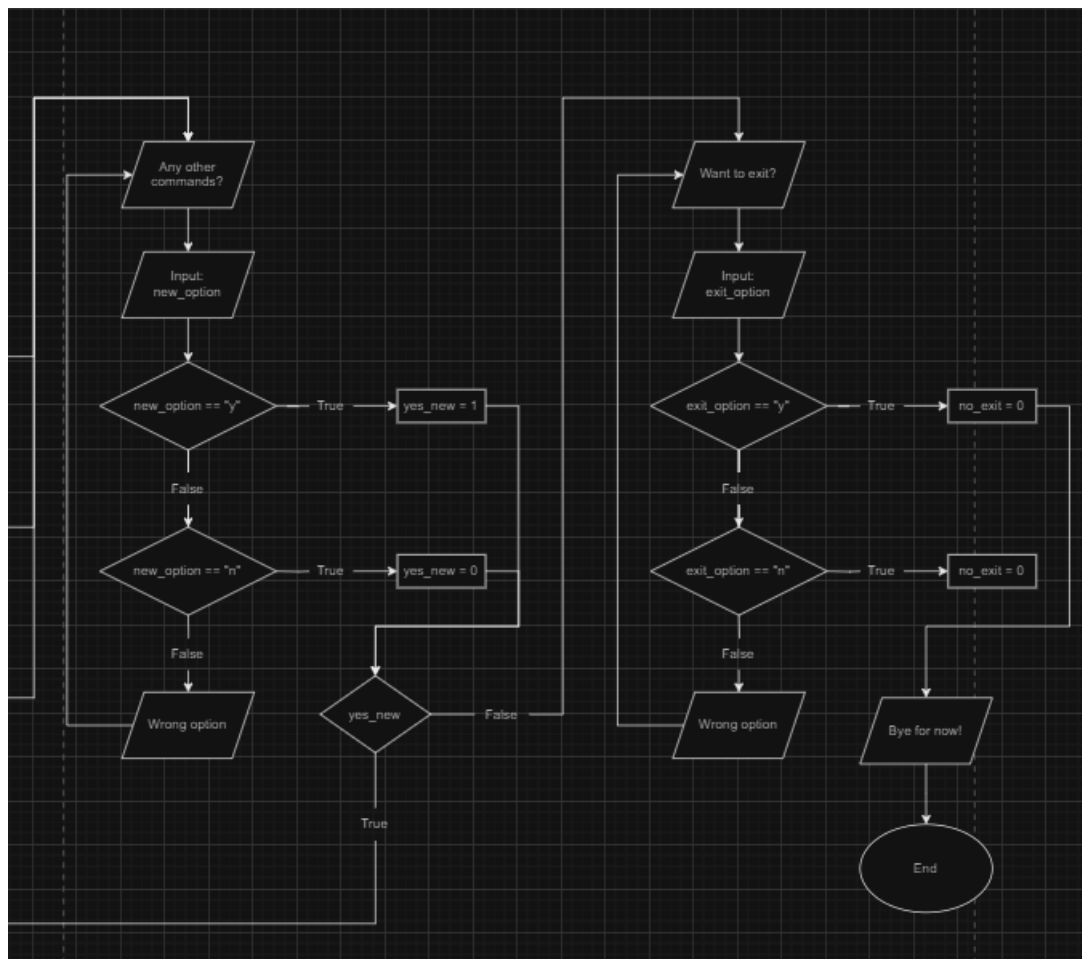
VNS Lab 7 Task 2



Оцінюваний час виконання: 20 хвилин.

Class Practice work





Оцінюваний час виконання: 1 години.

Self Practice work

Оцінюваний час виконання: 45 хв.

3. Код програми:

VNS Lab 2

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

// Знайти суму цілих додатніх чисел, більших 20, менших 100 і кратних 3

using namespace std;

int main(){
    int a;
    for (int i = 21; i < 100; i += 3)
    {
        a += i;
    }
    cout << a;
}
```

VNS Lab 3

```
#include <stdio.h>
#include <iostream>
#include <cmath>

using namespace std;

const double n = 10;
const double e = 0.0001;
const int k = 10;
const double a = 0.1;
const double b = 0.8;

double funcmain(double x)
{
    double return_func_main;
    return_func_main = x * atan(x) - log(sqrt(1.0 + pow(x, 2.0)));
    return return_func_main;
}

double sumN(double x, double n)
{
    double sum_n;
    for (int i = 1; i <= n; i++)
    {
        sum_n += pow(-1.0, i+1.0) * (pow(x, 2.0*i)/(2.0*i*(2.0*i - 1)));
    }
    return sum_n;
}
```

```

double sumE(double x, double e)
{
    double sum_e;
    double a_n;
    bool checker = true;
    int i = 1;
    while (checker)
    {
        a_n = pow(-1.0, i+1.0) * (pow(x, 2.0*i)/(2.0*i*(2.0*i - 1)));
        if (a_n > e)
        {
            checker = true;
            sum_e += a_n;
            i++;
        }
        else{
            checker = false;
        }
    }
    return sum_e;
}

int main()
{
    for (double x = a; x <= b; x += ((b-a)/k))
    {
        cout << "X = " << x << " SN = " << sumN(x, n) << " SE = " << sumE(x, e) << " Y = " << funcmain(x) << endl;
    }
}

```

VNS Lab 7 Task 1

```

#include <iostream>
#include <stdarg.h>

using namespace std;

int sum(int count, ...)
{
    va_list args;
    va_start(args, count);

    int total = 0;
    bool multiply = true;
    int temp = 1;

    for (int i = 0; i < count; i++) {
        int num = va_arg(args, int);

        if (multiply) {
            temp = num;
            multiply = false;
        } else {
            temp *= num;
            total += temp;
            multiply = true;
        }
    }

    va_end(args);
    return total;
}

int main() {
    cout << sum(8, 4, 5, 8, 9, 7, 15, 4, 2) << endl;
    cout << sum(10, 1, 3, 3, 4, 7, 8, 4, 12, 1, 2) << endl;
    cout << sum(12, 3, 16, 11, 3, 3, 4, 7, 8, 4, 12, 1, 2) << endl;
    return 0;
}

```

VNS Lab 7 Task 2

```

✓ #include <iostream>
  #include <complex>

  using namespace std;

✓ complex<double> difference(complex<double> a1, complex<double> a2){
    return a1-a2;
}

✓ double difference(double a1, double a2){
    return a1-a2;
}

✓ int main(){
    double real1, img1, real2, img2, a_1, a_2;
    cout << "Real and imaginary part of first complex number: ";
    cin >> real1 >> img1;
    cout << "Real and imaginary part of second complex number: ";
    cin >> real2 >> img2;
    complex<double> a1(real1, img1);
    complex<double> a2(real2, img2);

    cout << "Two rational numbers: ";
    cin >> a_1 >> a_2;

    cout << difference(a1, a2) << "\n";
    cout << difference(a_1, a_2) << "\n";
}

```

Class Practice work


```
1  #include <stdio.h>
2  #include <iostream>
3  #include <string>
4
5  using namespace std;
6
7  string books[5] = {"To Kill a Mockingbird",
8                    "1984",
9                    "Harry Potter and the Philosopher\'s Stone",
10                   "Pride and Prejudice",
11                   "The Great Gatsby"};
12
13  int status[5] = {1, 1, 1, 1, 1};
14
15  int _SIZE = sizeof(books) / sizeof(books[0]);
16
17  void recount(){
18      // for each loop
19      int a = 0;
20
21      for (int i : status)
22      {
23          if (i == 1)
24          {
25              cout << a+1 << ") " << books[a] << " ☐ доступна\n"; a--;
26          }
27          else
28          {
29              cout << a+1 << ") " << books[a] << " ☐ НЕдоступна\n"; a--;
30          }
31
32          a++;
33      }
```

```
33     }
34 }
35
36 void listofbooks(){
37     cout << "\n";
38     for (int i = 0; i < _SIZE; i++)
39     {
40         cout << i+1 << ") " << books[i] << "\n";
41     }
42     cout << "\n";
43 }
44
45 void borrowbook(){
46     int idx_b = 0;
47     listofbooks();
48     cout << "What book (from 1 to " << _SIZE << ") you want to borrow?\n";
49     cin >> idx_b;
50     if (status[idx_b-1] == 1)
51     {
52         status[idx_b-1] = 0;
53         cout << "The book is yours!\n\n";
54     }
55     else{
56         cout << "Sorry, the book is taken!\n\n";
57     }
58 }
59
60 void returnbook(){
61     int idx_r = 0;
```

```

60  ∨ void returnbook(){
61      int idx_r = 0;
62      // int borrowed = 0;
63  ∨  for (int i = 0; i < _SIZE; i++)
64      {
65  ∨      if (status[i] == 0){
66          cout << i+1 << ") " << books[i] << "\n";
67      }
68      }
69      cout << "What book you want to return?\n";
70      cin >> idx_r;
71      status[idx_r-1] = 1;
72      cout << "Thank you for your return\n";
73  }
74
75  ∨ int main(){
76      main_menu:
77      char new_option = 'a'; // (y/n) of new command
78      bool yes_new = 1; // boolean of new command
79      char exit_option = 'a'; // (y/n) of exit
80      bool no_exit = 1; // boolean of while(exit)
81
82      // while
83  ∨  while (no_exit)
84      {
85          int option_status = 0; // switch expression
86
87          // do while
88  ∨  do{
89      cout << "What do you want me to do? (option 1-3)\n";

```

```

88         do {
90             cout << "1) List of books.\n2) Take a book.\n3) Return a book.\n";
91             cin >> option_status;
92
93             switch (option_status)
94             {
95                 case 1:
96                     listofbooks();
97                     break;
98                 case 2:
99                     borrowbook();
100                    break;
101                 case 3:
102                     returnbook();
103                     break;
104                 default:
105                     cout << "Wrong input!\n";
106                     goto main_menu;
107                     break;
108             }
109
110             new_command_label:
111             cout << "Any other commands? (y/n)\n";
112             cin >> new_option;
113
114             if (new_option == 'y'){yes_new = 1;}
115             else if (new_option == 'n'){yes_new = 0;}
116             else{cout << "Wrong option\n"; goto new_command_label;}
117
118         }
119     }
120     while(yes_new);
121
122     exit_label:
123     cout << "Want to exit? (y/n)\n";
124     cin >> exit_option;
125
126     if (exit_option == 'y'){no_exit = 0;}
127     else if (exit_option == 'n'){no_exit = 1;}
128     else{cout << "Wrong option\n"; goto exit_label;}
129 }
130 cout << "Bye for now!";
131 }
132

```

Self Practice work

```
#include <stdio.h>
#include <iostream>
#include <string>
// #include <math.h>

using namespace std;

int main(){
    int n = 0;
    cin >> n;
    string a;
    cin >> a;

    char _k = 'K';
    char _v = 'V';

    int k_win = 0;
    int v_win = 0;

    int advantage_k_over_v = 0;
    int nmbr_wins_k = 0;
    int nmbr_wins_v = 0;

    for (int i = 0; i < n; i++)
    {
        if (a[i] == _k)
        {
            k_win++;
        }
        else if (a[i] == _v)
        {
            v_win++;
        }
    }
}
```

```

34         advantage_k_over_v = k_win - v_win;
35         if ((k_win >= 11) && (advantage_k_over_v >= 2))
36         {
37             k_win = 0;
38             v_win = 0;
39             nmbr_wins_k++;
40         }
41         else if ((v_win >= 11) && (advantage_k_over_v <= -2))
42         {
43             k_win = 0;
44             v_win = 0;
45             nmbr_wins_v++;
46         }
47     }
48     cout << nmbr_wins_k << ":" << nmbr_wins_v;
49     if ((k_win != 0) || (v_win != 0))
50     {
51         cout << "\n" << k_win << ":" << v_win;
52     }
53 }

```

5. Результати виконання завдань, тестування та фактично затрачений час:

VNS Lab 2

```

PS D:\Epic 3> & 'c:\
ine-In-yjzdnpos.bhs'
a1.gm3' '--dbgExe=C:
1620
PS D:\Epic 3>

```

Фактичний час виконання завдання: 5 хв.

VNS Lab 3

```

PS D:\Epic 3> & 'c:\Users\1\.vscode\extensions\ms-vsco
ine-In-iei3kfzm.xml3' '--stdout=Microsoft-MIEngine-Out-k
3n.xmlf' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--inte
X = 0.1 SN = 0.0049917 SE = 0.005 Y = 0.0049917
X = 0.17 SN = 0.0143812 SE = 0.01445 Y = 0.0143812
X = 0.24 SN = 0.0285297 SE = 0.0288 Y = 0.0285297
X = 0.31 SN = 0.0473085 SE = 0.04805 Y = 0.0473085
X = 0.38 SN = 0.0705556 SE = 0.0722 Y = 0.0705556
X = 0.45 SN = 0.0980829 SE = 0.10125 Y = 0.0980829
X = 0.52 SN = 0.129684 SE = 0.1352 Y = 0.129684
X = 0.59 SN = 0.165142 SE = 0.17405 Y = 0.165142
X = 0.66 SN = 0.204235 SE = 0.2178 Y = 0.204235
X = 0.73 SN = 0.24674 SE = 0.26645 Y = 0.246741
PS D:\Epic 3>

```

Фактичний час виконання завдання: 45 хв.

VNS Lab 7 Task 1

```

PS D:\Epic 3> & 'c:\
ine-In-p1xmsh3h.iku'
tg,i3i' '--dbgExe=C:\
205
121
199
PS D:\Epic 3>

```

Фактичний час виконання завдання: 2 год.

VNS Lab 7 Task 2

```

PS D:\Epic 3> & 'c:\Users\1\.vscode\extensions\ms-vsco
ine-In-uppindij.svo' '--stdout=Microsoft-MIEngine-Out-5
ps.kau' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--inte
Real and imaginary part of first complex number: 4 2
Real and imaginary part of second complex number: 3 6
Two rational numbers: 12 -6
(1,-4)
18
PS D:\Epic 3> █

```

Фактичний час виконання завдання: 15 хв.

Class Practice work

```

PS D:\Epic 3> & 'c:\Users\1\.vscode\extensions\ms-vsco
ine-In-llddhex.t04' '--stdout=Microsoft-MIEngine-Out-j
m4.rqx' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--inte
What do you want me to do? (option 1-3)
1) List of books.
2) Take a book.
3) Return a book.
1

1) To Kill a Mockingbird
2) 1984
3) Harry Potter and the Philosopher's Stone
4) Pride and Prejudice
5) The Great Gatsby

Any other commands? (y/n)
y
What do you want me to do? (option 1-3)
1) List of books.
2) Take a book.
3) Return a book.
2

1) To Kill a Mockingbird
2) 1984
3) Harry Potter and the Philosopher's Stone
4) Pride and Prejudice
5) The Great Gatsby

What book (from 1 to 5) you want to borrow?
3
The book is yours!

Any other commands? (y/n)
y
What do you want me to do? (option 1-3)
1) List of books.
2) Take a book.
3) Return a book.

```

```

What do you want me to do? (option 1-3)
1) List of books.
2) Take a book.
3) Return a book.

3
3) Harry Potter and the Philosopher's Stone
What book you want to return?
3
Thank you for your return
Any other commands? (y/n)
n
Want to exit? (y/n)
y
Bye for now!
PS D:\Epic 3>

```

Фактичний час виконання завдання: 2.5 год.

Self Practice work

```

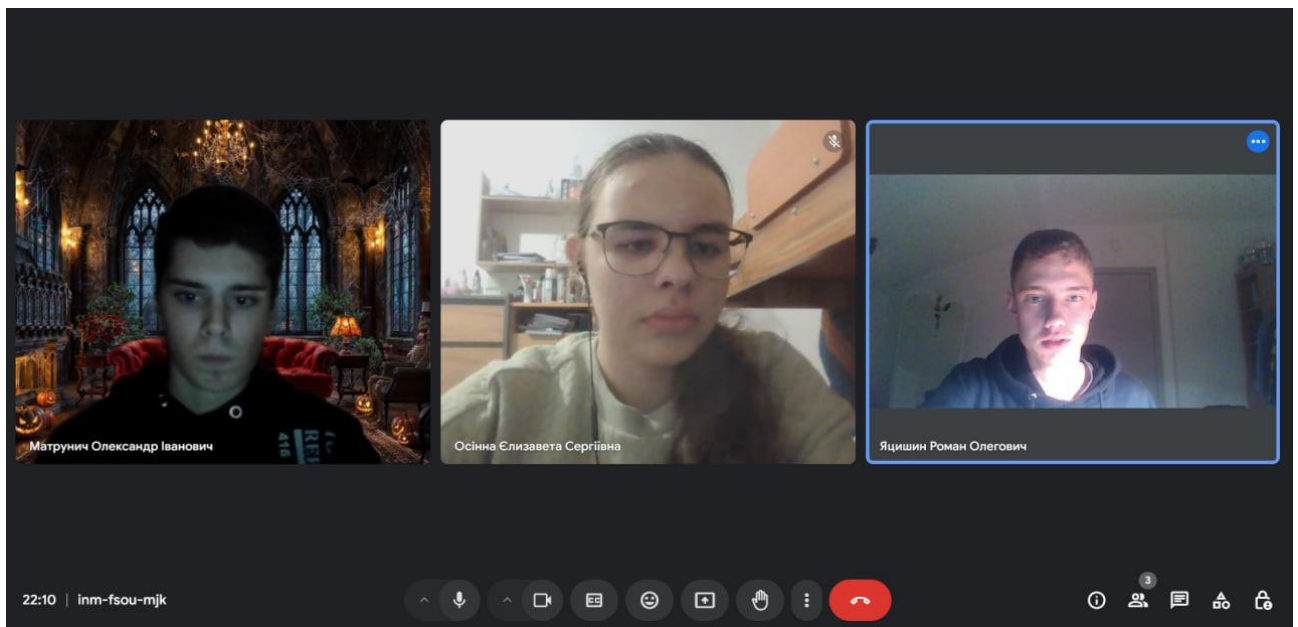
PS D:\Epic 3> & 'c:\Users\1\.v
ine-In-farnvyvd.dwa' '--stdout=
3u.c23' '--dbgExe=C:\msys64\ucr
30
VVVVKKKKVKKVVKVKKKKVKKVVVKVKV
1:0
2:5
PS D:\Epic 3>

```

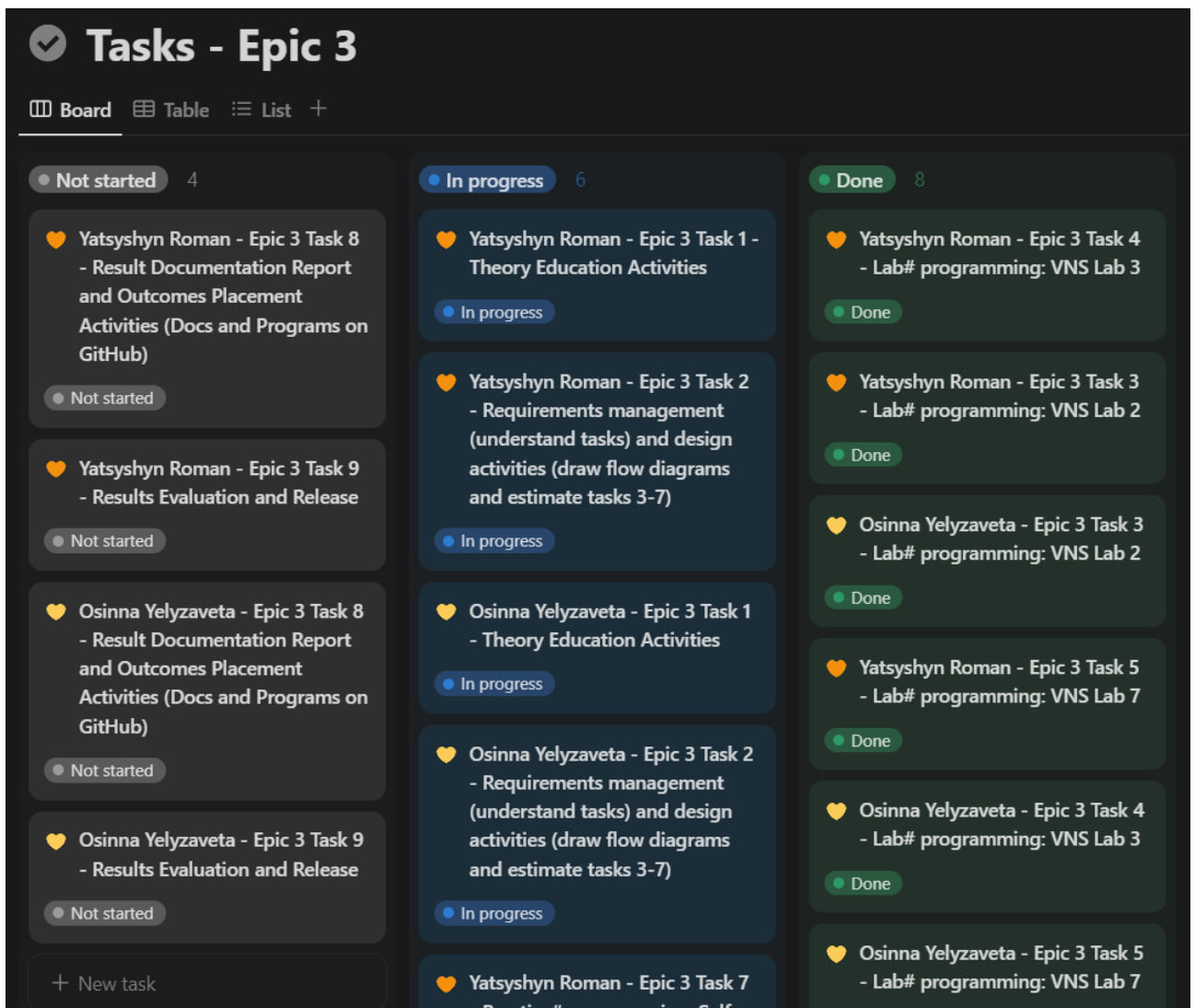
| Розв'язки | | |
|--------------|--------------------|------------|
| | Вихідний код | Компілятор |
| Хто | Роман Яцишин | |
| Задача | Коля, Вася і Теніс | |
| Стан | Processed | |
| Результат | Зараховано | |
| Файл | main.cpp | |
| Розмір файлу | 1056 | |
| Компілятор | C++ 23 | |

Фактичний час виконання завдання: 1.5 год

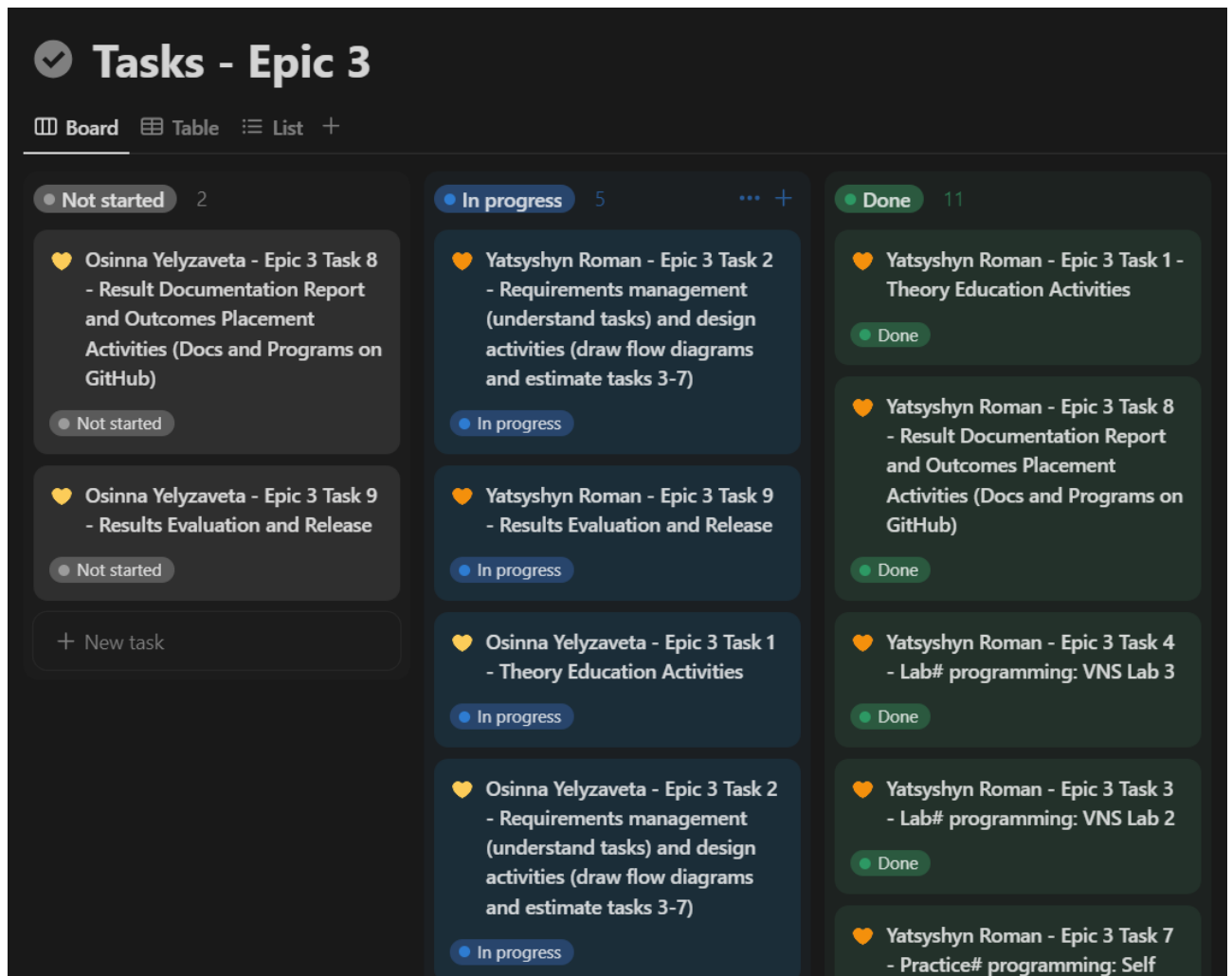
6. Кооперація з командою:



Before



After



Висновок:

У лабораторній роботі було розглянуто основи роботи з циклами, зокрема вкладеними циклами та способами завершення їх виконання, функції, перевантаження функцій, що дозволяє використовувати однакові імена для функцій з різними параметрами, еліпсис для роботи зі змінною кількістю аргументів і рекурсією для самовикликів.