Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

**про виконання лабораторних та практичних робіт блоку № 3**

На тему:  «Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції з змінною кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції.»

***з дисципліни:*** «Мови та парадигми програмування»

до:

ВНС Лабораторної Роботи № 2

ВНС Лабораторної Роботи № 3

ВНС Лабораторної Роботи № 7

Практичних Робіт № 3

**Виконала:**

Студентка групи ШІ-12

Виклюк Яна Ярославівна

# **Тема роботи:**

Цикли та їх види. Вкладені цикли. Функції. Перевантаження функції. Рекурсія

# **Мета роботи:**

Ознайомитись з циклами та їх видами у С++. Ознайомитись з вкладеними циклами. Ознайомитись з поняттям функцій, існуючими функціями, еліпсисом та навчитися писати власні функції. Ознайомитись та навчитись використовувати перевантаження функції, рекурсію.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли та їх види, оператор goto.
* Тема №2: Вкладені цикли.
* Тема №3: Функції, перевантаження функції.
* Тема №4: Еліпсис.
* Тема №5: Рекурсія

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли та їх види.
  + Джерела Інформації:
    - https://acode.com.ua/urok-72-tsykl-for/
    - <https://acode.com.ua/urok-70-tsykl-while/>
    - <https://acode.com.ua/urok-71-tsykl-do-while/>
    - <https://acode.com.ua/urok-69-operator-goto/>
  + Що опрацьовано:
    - Опрацьовано цикли while, do while, for, оператор goto.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 21.11
* Звершення опрацювання теми: 21.11
* Тема №2 : Вкладені цикли.
  + Джерела Інформації
    - <https://studfile.net/preview/5252553/page:3/>
  + Що опрацьовано:
    - Опрацьовано вкладені цикли
  + Статус: Ознайомлений
  + Початок опрацювання теми: 22.11
  + Звершення опрацювання теми: 22.11
* Тема №3: Функції, перевантаження функції.
  + Джерела Інформації:
    - <https://acode.com.ua/urok-15-funktsiyi-i-operator-return/#toc-2>
    - <https://acode.com.ua/urok-108-perevantazhennya-funktsij/>
  + Що опрацьовано:
    - Опрацьовано функції, перевантаження функції
  + Статус: Ознайомлений
  + Початок опрацювання теми: 23.11
  + Звершення опрацювання теми: 24.11
* Тема №4: Еліпсис.
  + Джерела Інформації:
    - <https://acode.com.ua/urok-117-elipsys/>
  + Що опрацьовано:
    - Опрацьовано еліпсис та особливості роботи з ним
  + Статус: Ознайомлений
  + Початок опрацювання теми: 24.11
  + Звершення опрацювання теми: 25.11
* Тема №5 Рекурсія.
  + Джерела Інформації:
    - <https://www.bestprog.net/uk/2019/01/07/recursion-examples-of-tasks-solving-advantages-and-disadvantages-of-recursion-ua-2/>
    - <https://acode.com.ua/urok-113-rekursiya-i-chysla-fibonachchi/>
  + Що опрацьовано:
    - Опрацьовано рекурсію
  + Статус: Ознайомлений
  + Початок опрацювання теми: 23.11
  + Звершення опрацювання теми: 24.11

# **Виконання роботи:**

## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 VNS Lab 2

* Варіант №8
* Деталі завдання
* Обчислити суму заданого ряду з точністю до ε=0.0001.
* Важливі деталі для врахування в імплементації програми
* Правильно вивести рекурентну формулу, зрозуміти як правильно знайти суму рядку з точністю.

Завдання №2 VNS Lab 3

* Варіант №8
* Деталі завдання
* Обчислити функції з використанням їхнього розкладу в степеневий ряд для заданого числа та із заданою точністю ε=0.0001, знайти точне значення функції.
* Важливі деталі для врахування в імплементації програми
* Правильно визначити за формулою загального члена його тип.

Завдання №3 VNS Lab 7 – Task 1

* Варіант №8
* Деталі завдання
* Потрібно написати функції зі змінною кількістю параметрів max та min, котрі визначають мінімальне та максимальне значення із чисел типу int. Написати викликаючу функцію main, що звертається до функцій не менше трьох разів з кількістю параметрів 5, 10, 12.
* Важливі деталі для врахування в імплементації програми
* Треба врахувати особливості роботи еліпсиса.

Завдання №4 VNS Lab 7 – Task 2

* Варіант №8
* Деталі завдання
* Написати перевантажені функції, одна з яких за номером року видає його назву за старояпонським календарем, інша - за назвою місяця видає знак Зодіаку, й основну програму, що їх викликає.
* Важливі деталі для врахування в імплементації програми
* Потрібно звернути увагу, що для того аби було перевантаження, назви функцій повинні бути однаковими, а тип даних з якими вони працюють різні (у першому випадку int, у другому string)

Завдання №5 Class Practice Work

* Варіант завдання відсутній
* Деталі завдання
* Створити просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

Завдання №6 Self Practice Work (Тренер слонів)

* Варіант завдання відсутній
* Умова
* Надивившись, як красиво та синхронно танцюють слони в Таїланді, Андрій вирішив, що він хоче стати дресувальником слонів. «Як повернусь в Україну, обіцяю стати найкращим тренером слонів у світі!» — вигукував Андрій.
* На жаль, на вулиці, де живе Андрій слонів немає — є лише коти. Однак Андрій не розгубився — він готовий працювати і з котами.
* Для простоти будемо вважати, що вулиця — це пряма лінія, а коти на ній — точки з певними координатами. Всього котів на вулиці є n, i-й кіт стоїть в точці xi. Для того, щоб розпочати свою діяльність, Андрію потрібно зібрати всіх котів в одну точку і дати їм тренерські настанови. За одну хвилину Андрій може вибрати довільну множину котів і наказати їм усім одночасно переміститись на одиницю відстані вліво або вправо (всі вибрані коти переміщаються в тому самому напрямку).
* Допоможіть Андрію знайти мінімальну кількість хвилин, за яку він зможе зібрати всіх котів в одній точці.
* Вхідні дані
* У першому рядку задано одне натуральне число n — кількість котів, що живуть на вулиці Андрія.
* У наступному рядку задано n цілих чисел, розділених пробілами — початкові координати всіх котів.
* Деталі завдання
* У єдиному рядку вивести одне ціле число — мінімальну кількість хвилин, за яку Андрій зможе зібрати всіх котів в одній точці.

Завдання №7 Self Practice Work (Депутатські гроші)

* Варіант завдання відсутній
* Умова
* Часто-густо громадяни намагаються з’ясувати, наскільки багатими є депутати. Дехто вірить, що матеріальні статки окремих депутатів є необмеженими.
* Тож уявіть собі депутата, у якого є необмежена кількість купюр усіх номіналів (1, 2, 5, 10, 20, 50, 100, 200 та 500 гривень). Він хоче придбати подарунок для своєї тещі, що коштує n гривень.
* Незважаючи на те, що наш герой-олігарх є нескінченно багатим, він також є нескінченно скупим та педантичним. Саме тому він хоче оплатити покупку готівкою без решти. Яка мінімальна кількість купюр йому для цього знадобиться?
* Вхідні дані
* У єдиному рядку задано одне натуральне число n — вартість подарунку.
* Деталі завдання
* Вивести одне ціле число — мінімальну кількість купюр, що необхідна для покупки подарунка.

## **2. Дизайн та планована оцінка часу виконання завдань:**

Програма №1 VNS Lab 2

* Блок-схема

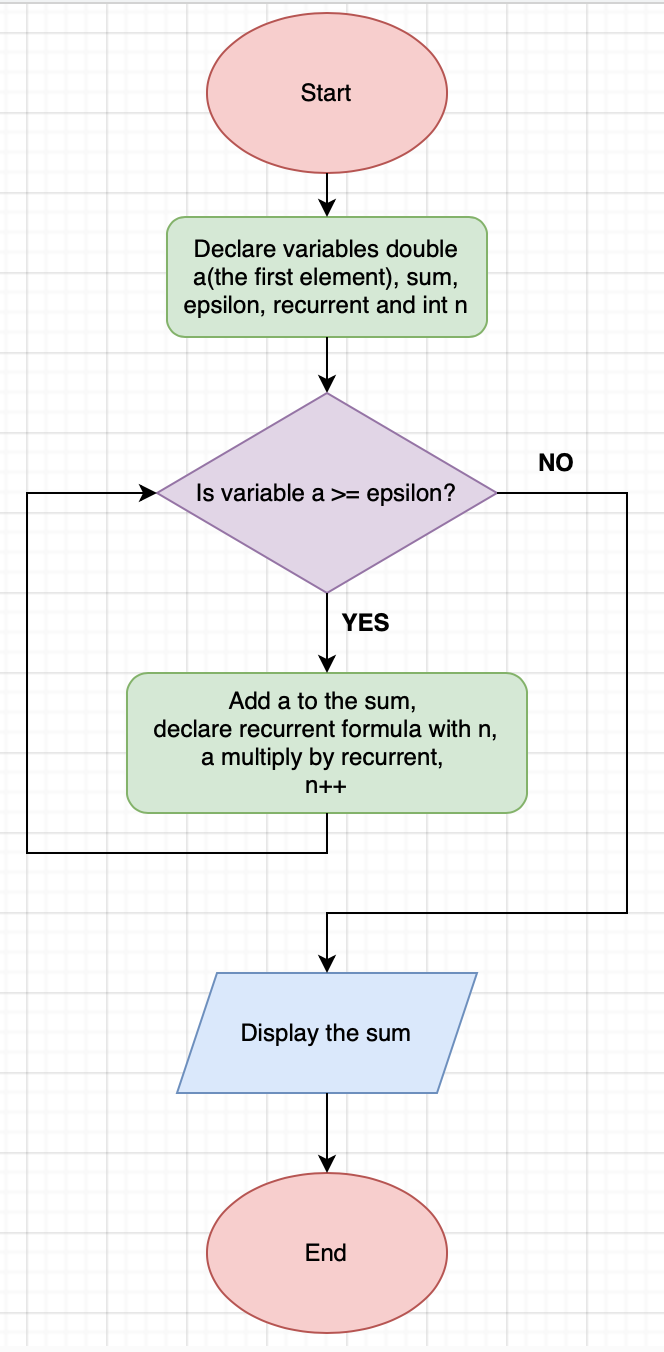


Рисунок Блок-схема до програми №1

* Планований час на реалізацію 40 хв
* Важливі деталі для врахування в імплементації
* Правильно вивести рекурентну формулу, обчислити перший член суми, уважно врахувати точність.

Програма №2 VNS Lab 3

* Блок-схема

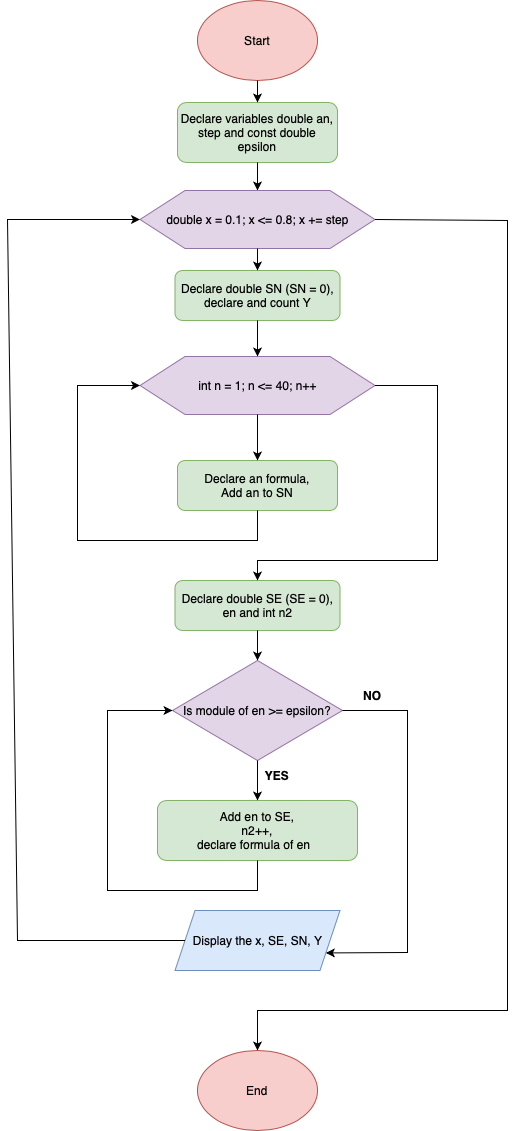


Рисунок Блок-схема до програми №2

* Планований час на реалізацію 90 хв
* Важливі деталі для врахування в імплементації
* Правильно визначити за формулою загального члена його тип. Обчислити правильний крок.

Програма №3 VNS Lab 7 – Task 1

* Блок-схема

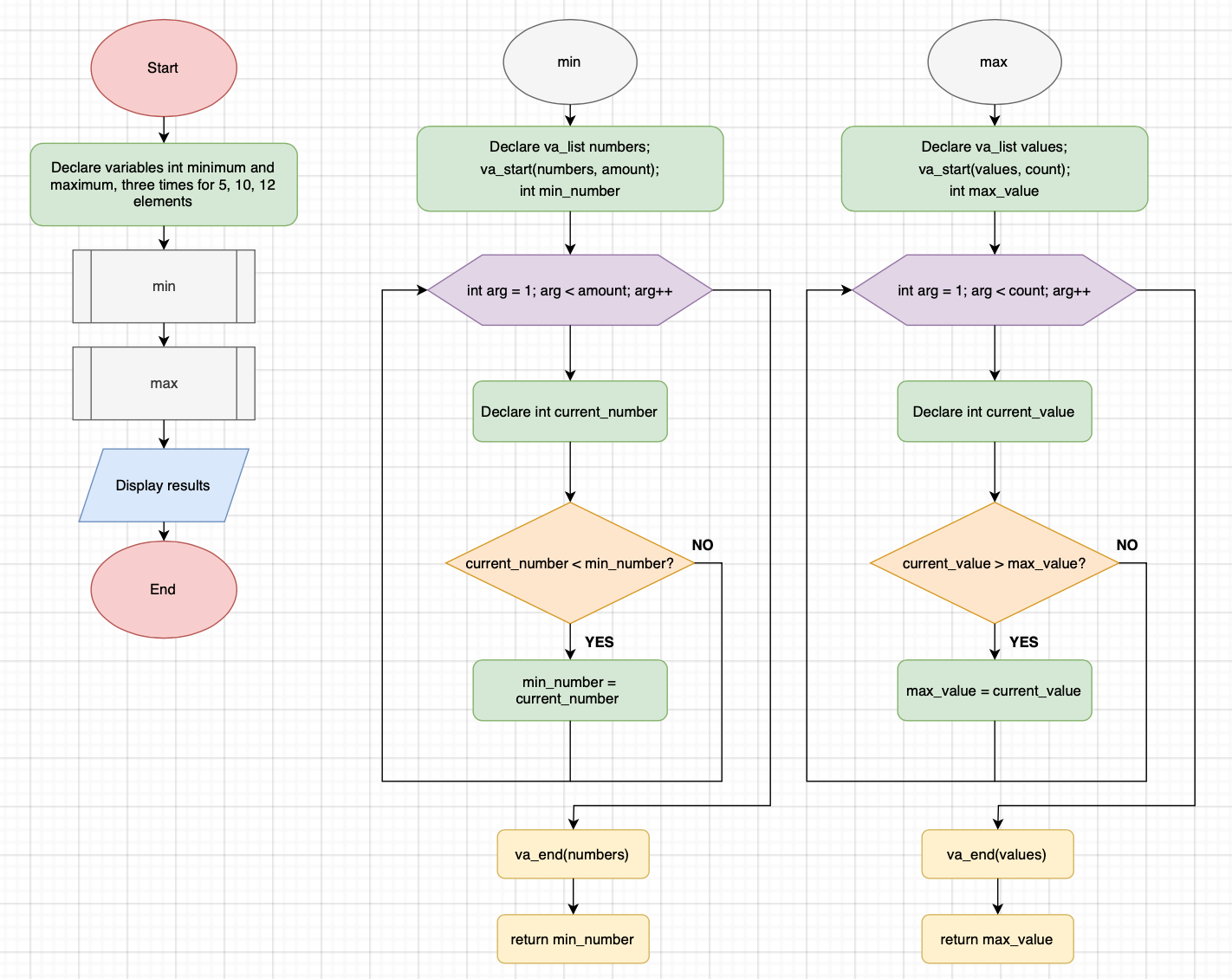


Рисунок Блок-схема до програми №3

* Планований час на реалізацію 90 хв
* Важливі деталі для врахування в імплементації
* Уважно написати функції зі змінними параметрами, для кращої та безпечнішої роботи використати va\_list.

Програма №4 VNS Lab 7 – Task 2

* Блок-схема

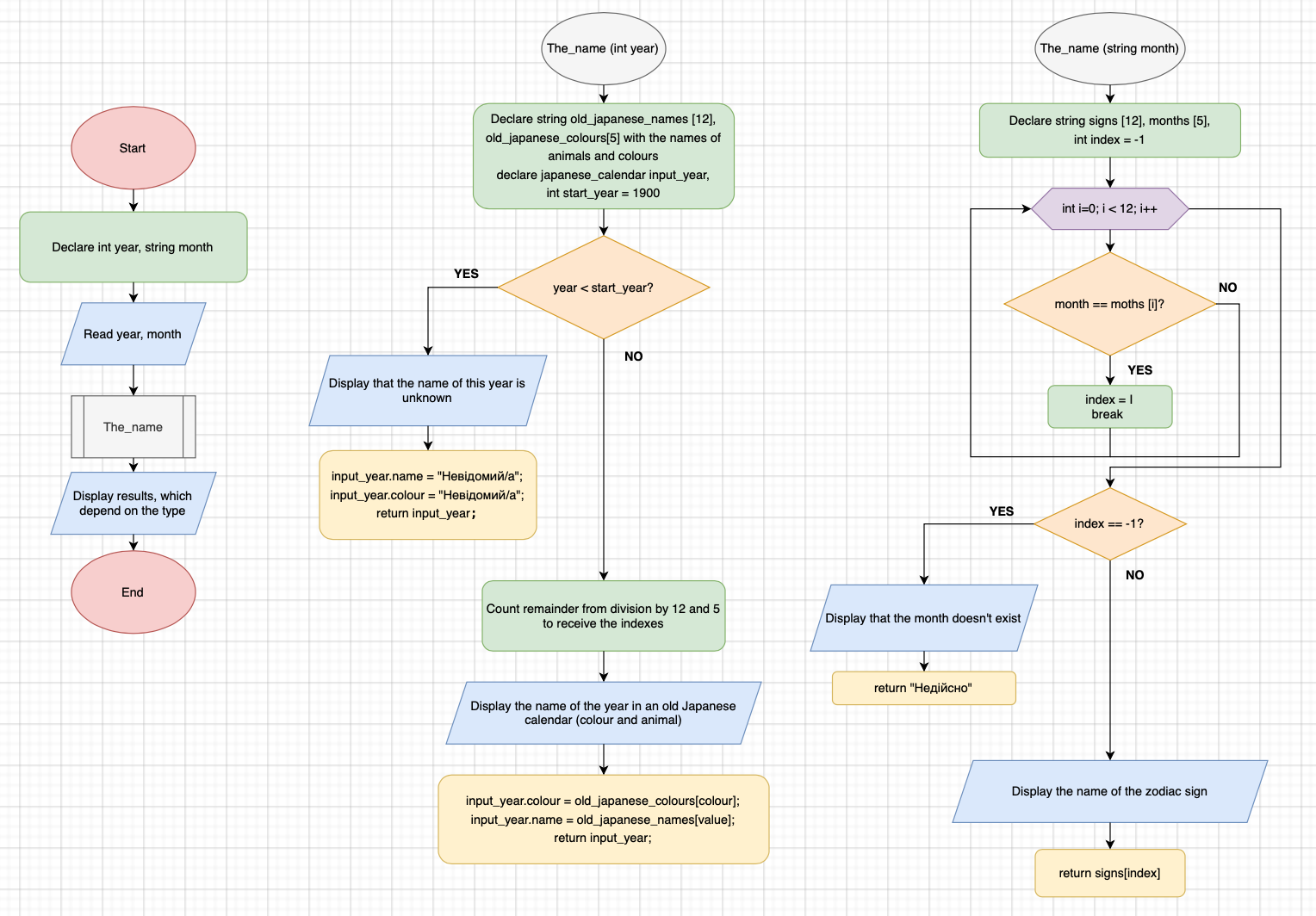


Рисунок Блок-схема до програми №4

* Планований час на реалізацію 90 хв
* Важливі деталі для врахування в імплементації
* Дати однакову назву функціям, котрі працюють з різними типами, аби зробити перевантаження.

Програма №5 Class Practice Work

* Блок-схема

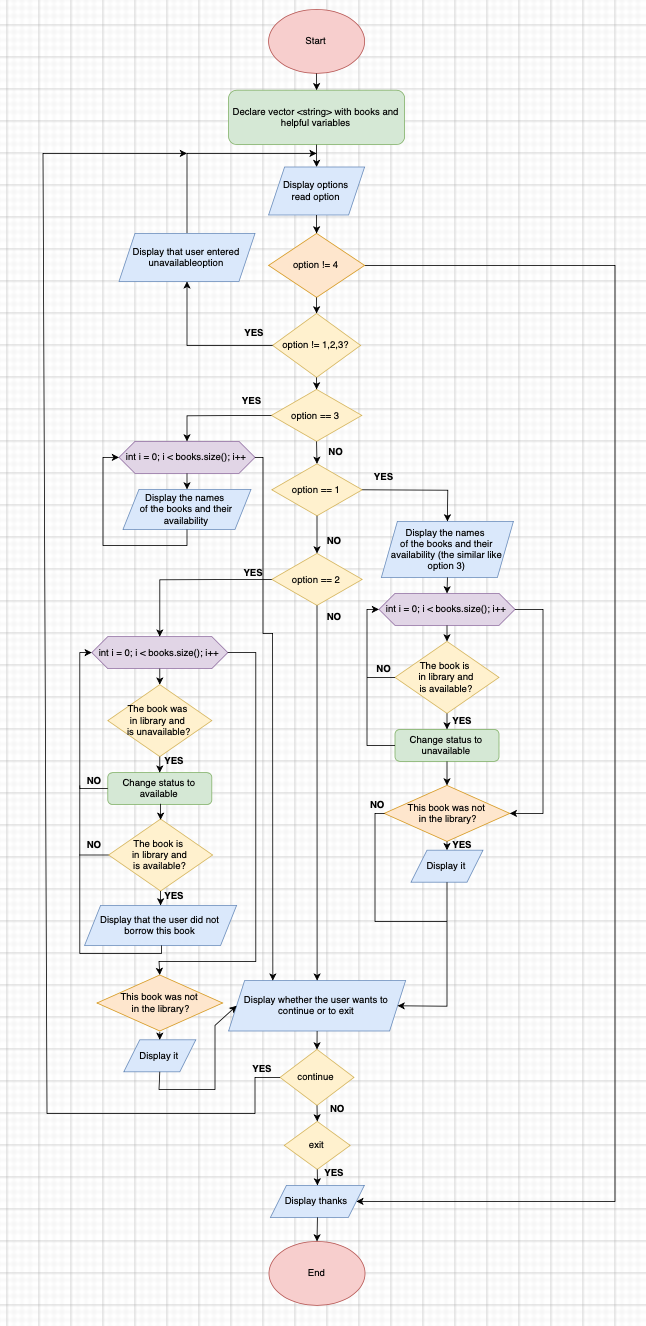


Рисунок Блок-схема до програми №5

* Планований час на реалізацію 2 год
* Важливі деталі для врахування в імплементації
* Уважно писати цикли, аби не залізти у нескінченний цикл.

Програма №6 Self Practice Work (Тренер слонів)

* Блок-схема

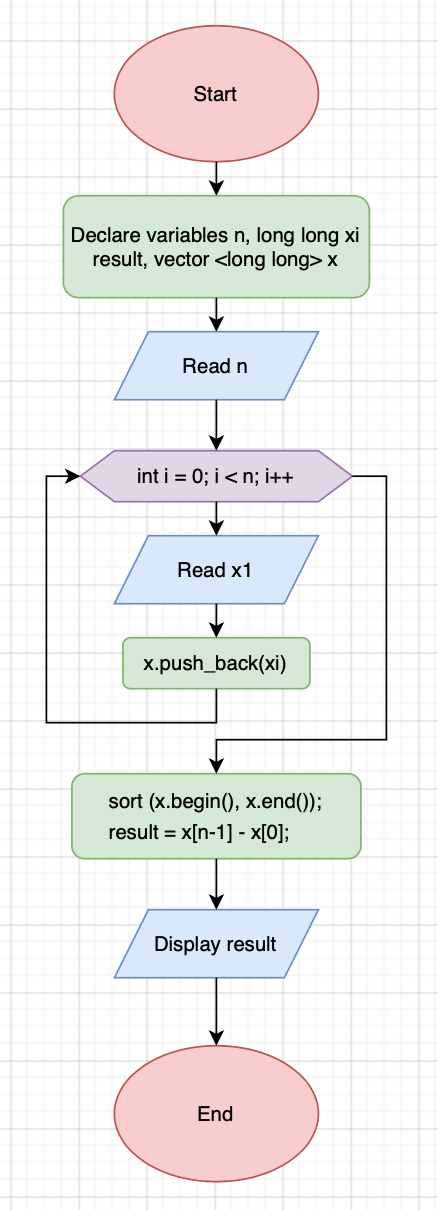


Рисунок 6 Блок-схема до програми №6

* Планований час на реалізацію 40 хв
* Важливі деталі для врахування в імплементації
* Визначити, що мінімальна кількість хвилин дорівнює різниці між мінімальною та максимальною координатами.

Програма №7 Self Practice Work (Депутатські гроші)

* Блок-схема

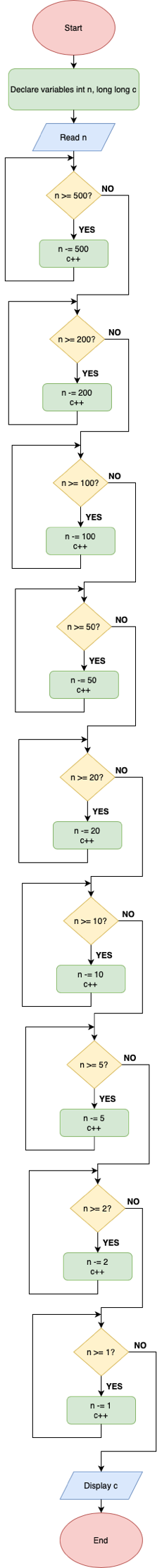


Рисунок 7 Блок-схема до програми №7

* Планований час на реалізацію 30 хв
* Важливі деталі для врахування в імплементації
* Врахувати, що коли рахую купюру потрібно відняти від загальної суми номінал купюри.

## **3. Конфігурація середовища до виконання завдань:**

Деталі по конфігурації середовища

Достатньо базових розширень та налаштувань Visual Studio Code, C++, нічого додаткового не потрібно для кожного завдання.

## **4. Код програм з посиланням на зовнішні ресурси:**

Завдання №1 VNS Lab 2

Деталі по програмі

Потрібно обчислити суму заданого ряду з точністю до ε=0.0001, попередньо вивести рекурентну формулу.

#include <iostream>

using namespace std;

double sum = 0;

double a = 0.5;

double epsilon = 0.0001;

int n = 1;

double reccurent;

int main()

{

while(a >= epsilon)

{

sum += a;

reccurent = ((2.0\*n)+1.0)/((4.0\*n)-2.0);

a \*= reccurent;

n++;

}

cout << sum << endl;

return 0;

}

Блок з кодом до завдання №1

Завдання №2 VNS Lab 3

Деталі по програмі

Потрібно обчислити функції з використанням їхнього розкладу в степеневий ряд для заданого числа та із заданою точністю ε=0.0001, знайти точне значення функції.

#include <iostream>

#include <cmath>

using namespace std;

const double epsilon = 0.0001;

double an = 0.0;

double step = (0.8-0.1)/10;

int main()

{

for (double x = 0.1; x <= 0.8; x += step)

{

double SN = 0;

double Y = (x\*sin(M\_PI/4))/(1 - (2\*x\*cos(M\_PI/4))+ (pow(x,2)));

for(int n = 1; n <= 40; n++)

{

an = pow(x,n)\*sin(n \* (M\_PI/4));

SN += an;

}

double SE = 0;

int n2 = 1;

double en = pow(x,n2)\*sin(n2 \* (M\_PI/4));

while (fabs(en) >= epsilon)

{

SE += en;

n2++;

en = pow(x,n2)\*sin(n2 \* (M\_PI/4));

}

cout << "X = " << x << " SN = " << SN << " SE = " << SE << " Y = " << Y << endl;

}

return 0;

}

Блок з кодом до завдання №2

Завдання №3 VNS Lab 7 Task 1

Деталі по програмі

Потрібно написати функції зі змінною кількістю параметрів max та min, котрі визначають мінімальне та максимальне значення із чисел типу int. Написати викликаючу функцію main, що звертається до функцій не менше трьох разів з кількістю параметрів 5, 10, 12

#include <iostream>

#include <cstdarg>

int min(int amount, ...)

{

va\_list numbers;

va\_start(numbers, amount);

int min\_number = va\_arg(numbers, int);

for (int arg = 1; arg < amount; arg++)

{

int current\_number = va\_arg(numbers, int);

if (current\_number < min\_number)

{

min\_number = current\_number;

}

}

va\_end(numbers);

return min\_number;

}

int max(int count, ...)

{

va\_list values;

va\_start(values, count);

int max\_value = va\_arg(values, int);

for (int arg = 1; arg < count; arg++)

{

int current\_value = va\_arg(values, int);

if (current\_value > max\_value)

{

max\_value = current\_value;

}

}

va\_end(values);

return max\_value;

}

int main()

{

int minimum1 = min(5, 12, 45, 3, 78, 68);

int maximum1 = max(5, 12, 45, 3, 78, 68);

int minimum2 = min(10, -6, 23, 1, -69, 114, 13, 26, -2, 43, 5);

int maximum2 = max(10, -6, 23, 1, -69, 114, 13, 26, -2, 43, 5);

int minimum3 = min(12, 321, 0, 87, -101, 100, 307, 6, 33, 562, 10, -7, -101);

int maximum3 = max(12, 321, 0, 87, -101, 100, 307, 6, 33, 562, 10, -7, -101);

printf("Minimal number: %d\n", minimum1);

printf("Maximal number: %d\n\n", maximum1);

printf("Minimal number: %d\n", minimum2);

printf("Maximal number: %d\n\n", maximum2);

printf("Minimal number: %d\n", minimum3);

printf("Maximal number: %d\n", maximum3);

return 0;

}

Блок з кодом до завдання №3

Завдання №4 VNS Lab 7 Task 2

Деталі по програмі

Потрібно написати перевантажені функції, одна з яких за номером року видає його назву за старояпонським календарем, інша - за назвою місяця видає знак Зодіаку, й основну програму, що їх викликає.

#include <iostream>

#include <string>

using namespace std;

struct japanese\_calendar

{

string name;

string colour;

};

japanese\_calendar The\_name(int year)

{

string old\_japanese\_names[12] = {"Щур", "Бик","Тигр", "Кролик", "Дракон", "Змія", "Кінь", "Коза", "Мавпа", "Півень", "Собака", "Свиня"};

string old\_japanese\_colours[5] = {"Білий/а", "Блакитний/а", "Зелений/а", "Червоний/а", "Жовтий/а"};

japanese\_calendar input\_year;

int start\_year = 1900;

if (year < start\_year)

{

cout << "Невідома назва заданого року за старояпонським календарем." << endl;

input\_year.name = "Невідомий/а";

input\_year.colour = "Невідомий/а";

return input\_year;

}

int value = (year - start\_year) % 12;

int colour = (year - start\_year) % 5;

cout << "Навза Вашого року за старояпонським календарем: " << old\_japanese\_colours[colour] << " " << old\_japanese\_names[value] << endl;

input\_year.colour = old\_japanese\_colours[colour];

input\_year.name = old\_japanese\_names[value];

return input\_year;

}

string The\_name(string month)

{

string signs[12] = {"Овен", "Телець", "Близнюки", "Рак", "Лев", "Діва", "Терези", "Скорпіон", "Стрілець", "Козеріг", "Водолій", "Риби"};

string months[12] = {"Квітень", "Травень", "Червень", "Липень", "Серпень", "Вересень", "Жовтень", "Листопад", "Грудень", "Січень", "Лютий", "Березень"};

int index = -1;

for (int i=0; i < 12; i++)

{

if(month == months[i])

{

index = i;

break;

}

}

if(index == -1)

{

cout << "Введіть дійсний місяць!" << endl;

return "Недійсно";

}

cout << "Ваш знак зодіаку: " << signs[index] << endl;

return signs[index];

}

int main()

{

int year;

string month;

cout << "Введіть рік за григоріанським календарем: " << endl;

cin >> year;

cout << "Введіть місяць з великої букви " << endl;

cin >> month;

The\_name(year);

The\_name(month);

return 0;

}

Блок з кодом до завдання №4

Завдання №5 Class Practice Work

Деталі по програмі

Потрібно створити просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

#include <iostream>

#include <string>

#include <vector>

using namespace std;

int main()

{

vector <string> books;

books.push\_back("Sherlock Holmes");

books.push\_back("1984");

books.push\_back("It");

books.push\_back("The Picture of Dorian Gray");

vector <string> is\_available;

for (int i = 0; i < books.size(); i++)

{

is\_available.push\_back("available");

}

string borrow\_book, return\_book, next\_option;

int option;

Start\_menu:

cout << "Choose your option: \n1 - borrow\n2 - return\n3 - recalculate\n4 - exit" << endl;

cin >> option;

while(option != 4)

{

if (option != 1 && option != 2 && option != 3)

{

cout << "Enter available option!" << endl;

goto Start\_menu;

}

if (option == 3)

{

for (int i = 0; i < books.size(); i++)

{

cout << "The book " << books[i] << " is " << is\_available[i] << endl;

}

}

if (option == 1)

{

cout << "Choose the book you wanna borrow: \n";

for (int i = 0; i < books.size(); i++)

{

cout << books[i] << " is " << is\_available[i] << endl;

}

cin.ignore();

getline(cin, borrow\_book);

bool av\_book = false;

for (int i = 0; i < books.size(); i++)

{

if(borrow\_book == books[i] && is\_available[i] == "available")

{

cout << "You borrow: " << borrow\_book << endl;

is\_available[i] = "unavailable";

av\_book = true;

break;

}

}

if (av\_book != true)

{

cout << "Unfortunately, this book is unavailable :(" << endl;

}

}

if(option == 2)

{

cout << "Oh, you finally decided to return the book you borrowed. Please name it!" << endl;

cin.ignore();

getline(cin, return\_book);

bool rt\_book = false;

for (int i = 0; i < books.size(); i++)

{

if (return\_book == books[i] && is\_available[i] == "unavailable")

{

cout << "Sadly, you won't pay any fine" << endl;

is\_available[i] = "available";

rt\_book = true;

break;

}

else if(return\_book == books[i] && is\_available[i] == "available")

{

cout << "You did not borrow this book!" << endl;

rt\_book = true;

break;

}

}

if(rt\_book != true)

{

cout << "This book was not in our library, but you can present it to us)" << endl;

}

}

bool nxt = false;

do

{

cout << "Do you want to do any other option? (yes or no)" << endl;

cin.clear();

cin >> next\_option;

if (next\_option == "yes")

{

nxt = true;

cout << "Choose your option: \n1 - borrow\n2 - return\n3 - recalculate\n4 - exit" << endl;

cin.ignore();

cin >> option;

break;

}

else if (next\_option == "no")

{

option = 4;

nxt = true;

break;

}

cout << "Enter correct answer!" << endl;

}

while (nxt != true);

}

cout << "Thanks)" << endl;

return 0;

}

Блок з кодом до завдання №5

Завдання №6 Self Practice Work (Тренер слонів)

Деталі по програмі

Програма виводить одне ціле число — мінімальну кількість хвилин, за яку Андрій зможе зібрати всіх котів в одній точці.

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int n;

long long xi, result;

vector <long long> x;

int main()

{

cin >> n;

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

{

cin >> xi;

x.push\_back(xi);

}

sort (x.begin(), x.end());

result = x[n-1] - x[0];

cout << result << endl;

return 0;

}

Блок з кодом до завдання №6

Завдання №7 Self Practice Work (Депутатські гроші)

Деталі по програмі

Програма виводить одне ціле число — мінімальну кількість купюр, що необхідна для покупки подарунка.

#include <iostream>

using namespace std;

long long n;

int c;

int main ()

{

cin >> n;

while(n >= 500)

{

n -= 500;

c++;

}

while(n >= 200)

{

n -= 200;

c++;

}

while (n >= 100)

{

n -= 100;

c++;

}

while (n >= 50)

{

n -= 50;

c++;

}

while (n >= 20)

{

n -= 20;

c++;

}

while (n >= 10)

{

n -= 10;

c++;

}

while (n >= 5)

{

n -= 5;

c++;

}

while (n >= 2)

{

n -= 2;

c++;

}

while (n >= 1)

{

n -= 1;

c++;

}

cout << c << endl;

return 0;

}

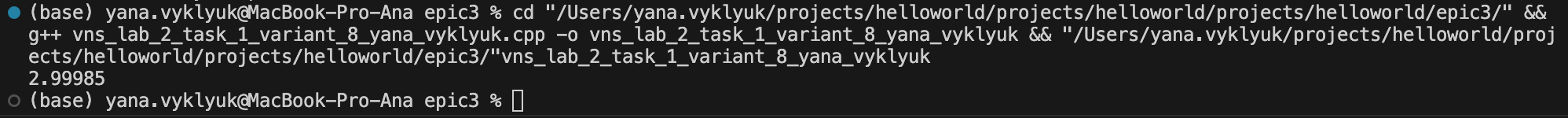
Блок з кодом до завдання №7

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

Завдання №1 VNS Lab 2

Деталі по виконанню і тестуванню програми

При виконанні цієї програми сума ряду знаходиться за допомогою рекурентної формули, як тільки член ряду стане меншим за точність цикл перестане працювати і буде виведено суму ряду з заданою точністю.



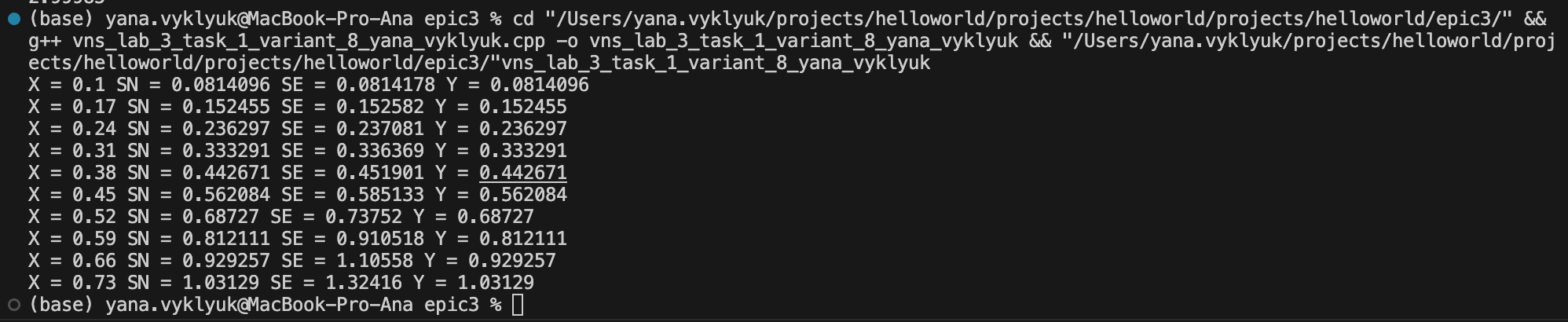
Блок з виконанням та тестуванням програми до завдання №1

Час затрачений на виконання завдання 40 хв.

Завдання №2 VNS Lab 3

Деталі по виконанню і тестуванню програми

Програма виводить спочатку значення Х (межі якого, та крок були задані в умові) щодо якого ми шукаємо значення функції, потім виводить SN, тобто суму степеневого ряду для заданого n, SE з точністю до епсилон та точне значення функції Y. SN та Y збігаються, а SE має певну похибку, котра з кожним наступним членом зростає, тож можна зробити висновок, що розклад у степеневий ряд є більш точним, аніж пошук значення функції з певною точністю, котра і дає похибку.



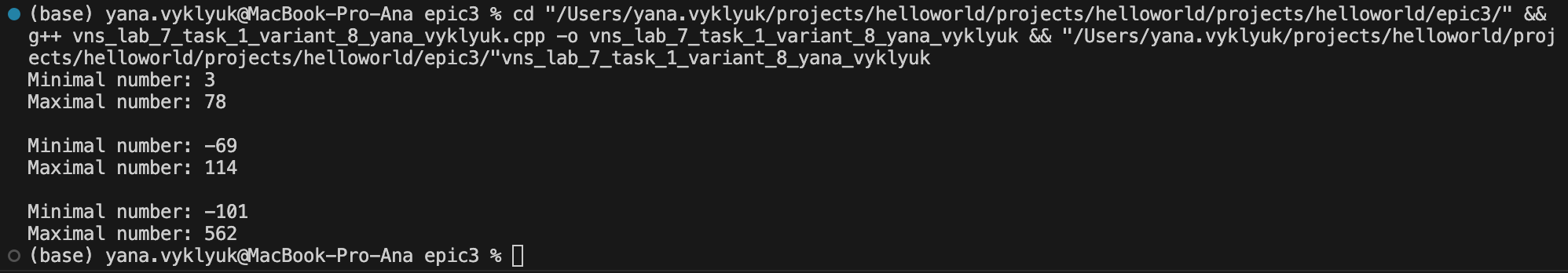
Блок з виконанням та тестуванням програми до завдання №2

Час затрачений на виконання завдання 80 хв

Завдання №3 VNS Lab 7 Task 1

Деталі по виконанню і тестуванню програми

Програма виводить 3 результати з мінімальними та максимальними елементами (кількість 5, 10, 12). Для цього написано відповідно дві функції, котрі перший елемент при їх виклику вважають сталим та розміром масиву. Потрібно звернути увагу, що необхідно задавати лише значення типу, вказаного у функціях, адже у іншому випадку результат буде неочікуваним, проте програма скампілюється.



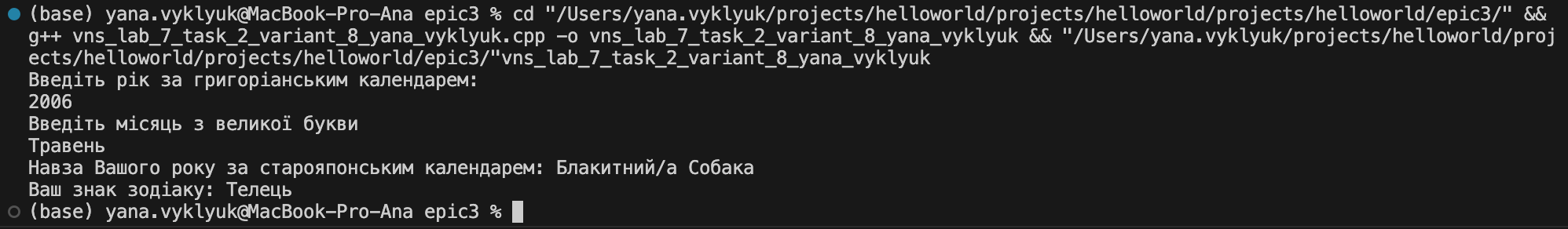
Блок з виконанням та тестуванням програми до завдання №3

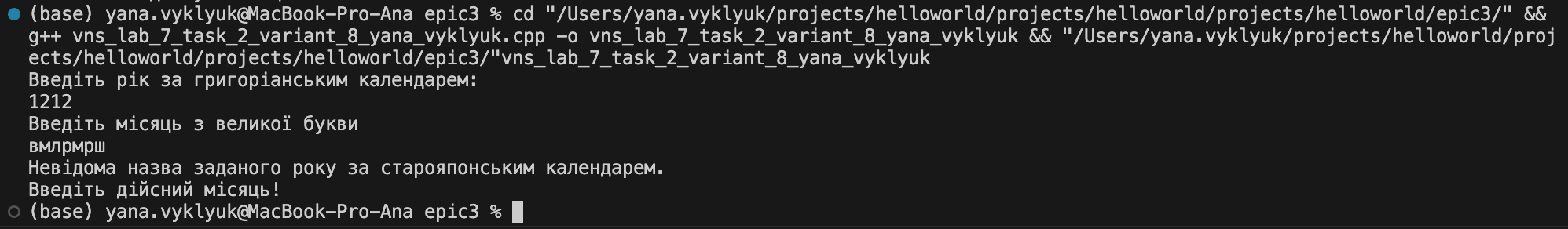
Час затрачений на виконання завдання 90 хв

Завдання №4 VNS Lab 7 Task 2

Деталі по виконанню і тестуванню програми

При введені року меншого за 1900 та незаданих місяців програма просить ввести дійсні умови. Ввівши останні, залежно від року та місяцю програма виводить назву за старояпонським календарем та знак зодіаку. Функції названі однаково, але приймають різні типи, тож тут є перевантаження, та в залежності від введеного типу програма викличе ту чи іншу функцію.



****

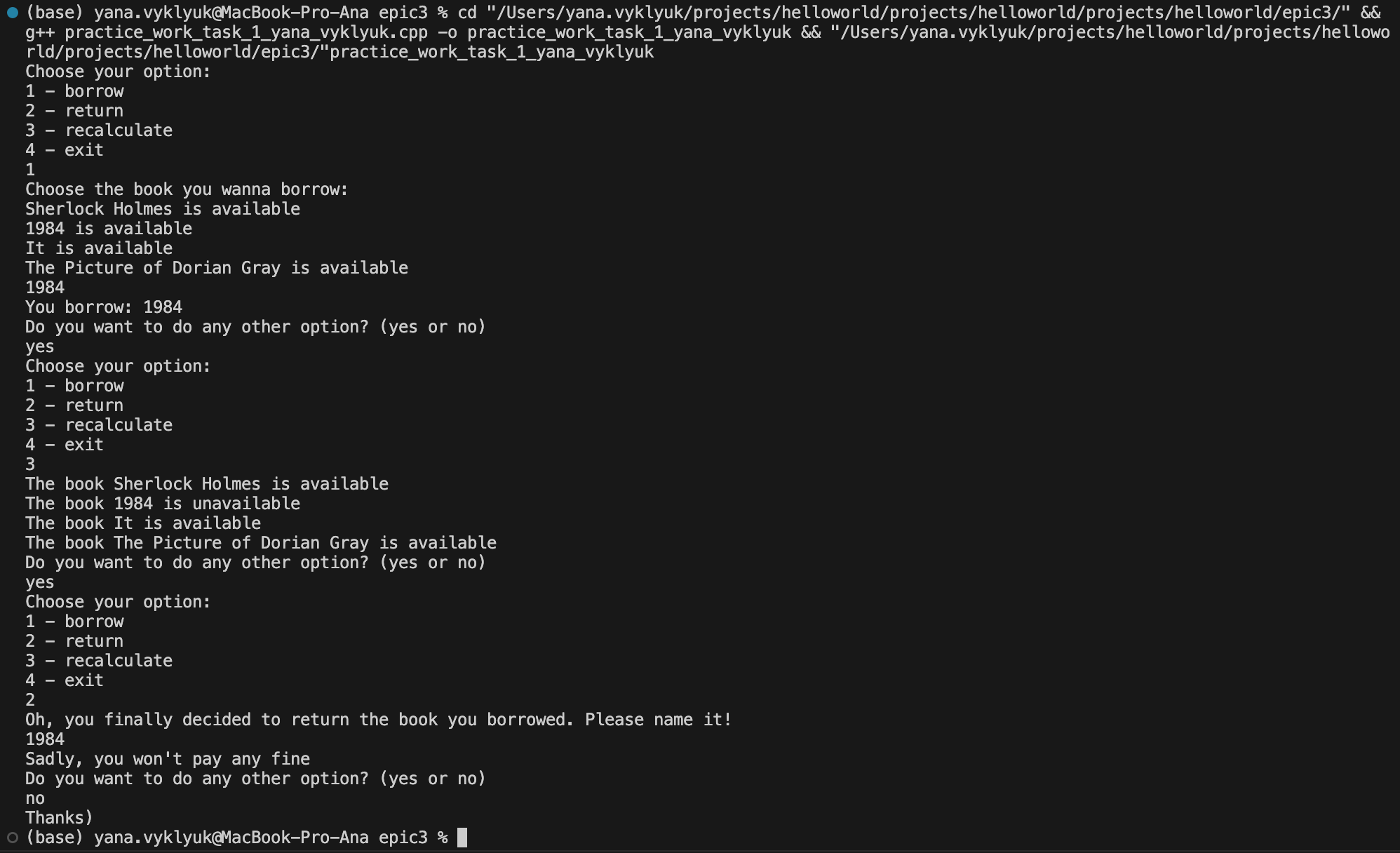
Блок з виконанням та тестуванням програми до завдання №4

Час затрачений на виконання завдання 90 хв.

Завдання №5 Class Practice Work

Деталі по виконанню і тестуванню програми

Програма пропонує обрати опцію, позичити, повернути книги, перерахувати їх та вийти. Після кожної операції програма перепитує чи хоче користувач продовжити далі. У разі, якщо користувач позичив книгу її статус змінюється на не доступний, і він не можу позичити такі книги, у разі повернення статус книги змінюється на доступний, якщо ж користувач хоче повернути книгу котрої нема, то програма про це повідомляє.



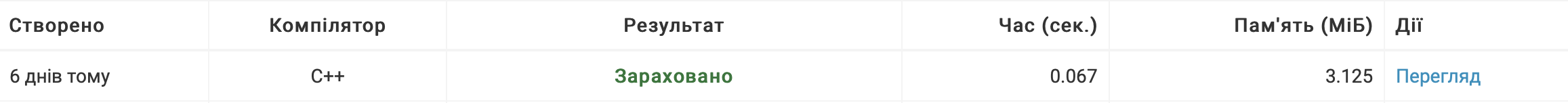
Блок з виконанням та тестуванням програми до завдання №5

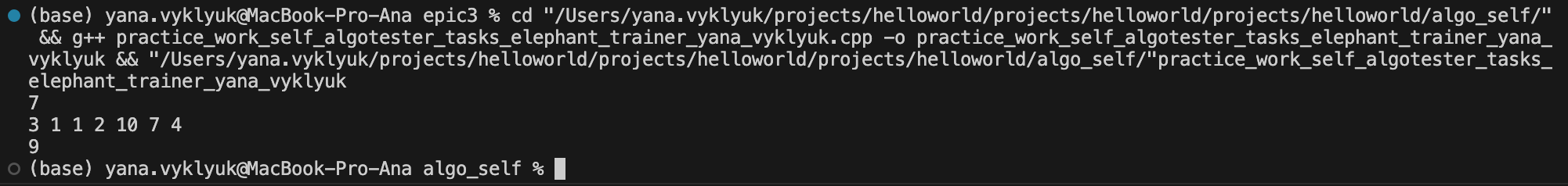
Час затрачений на виконання завдання 2 год.

Завдання №6 Self Practice Work (Тренер слонів)

Деталі по виконанню і тестуванню програми

Код програми загружено на Algotester. Програма виводить мінімальну кількість хвилин, за яку Андрій зможе зібрати всіх котів в одній точці.





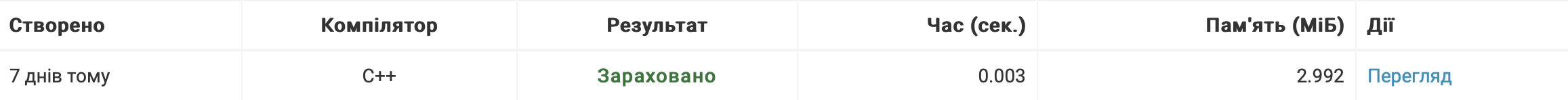
Блок з виконанням та тестуванням програми до завдання №6

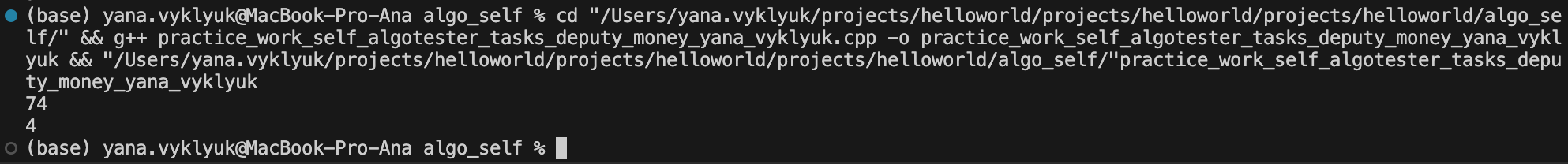
Час затрачений на виконання завдання 40 хв.

Завдання №7 Self Practice Work (Депутатські гроші)

Деталі по виконанню і тестуванню програми

Код програми загружено на Algotester. Програма виводить мінімальну кількість купюр, що необхідна для покупки подарунка.





Блок з виконанням та тестуванням програми до завдання №7

Час затрачений на виконання завдання 30 хв.

# **Висновки:**

Ознайомлено з циклами та їх видами у С++. Ознайомлено з вкладеними циклами. Застосовано їх у програмах. Ознайомлено з поняттям функцій, існуючими функціями, еліпсисом(фукціями для змінних параметрів), особливостями роботи з ним. Навчилася писати власні функції. Ознайомилась та навчилась використовувати перевантаження функції, рекурсію.