Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № (3)**

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

***з розділу:*** «Epic 3. Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія. Змінні»

***Виконав:***

студент групи ШІ-14

Чумаченко Дем’ян Сергійович

**Тема роботи:**

Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія Змінні

**Мета роботи:**

Навчитись використовувати прості цикли та вкладені цикли, застосовувати перевантаження функції і рекурсію, вміти створювати функції

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Theory Education Activities
* Тема №2:  Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)
* Тема №3: Lab# programming: VNS Lab 2
* Тема №4: Lab# programming: VNS Lab 3
* Тема №5  Lab# programming: VNS Lab 7
* Тема №6 Practice# programming: Class Practice Task
* Тема №7 Practice# programming: Self Practice Task

1. Індивідуальний план опрацювання теорії:

* **Тема №1: Theory Education Activities**

1. Джерела Інформації

<https://stackoverflow.com/questions/1338728/how-do-i-delete-a-commit-from-a-branch>

<https://confluence.atlassian.com/bitbucketserverkb/error-invalid-path-during-git-clone-to-windows-client-1085186345.html>

<https://learn.microsoft.com/ru-ru/cpp/cpp/goto-statement-cpp?view=msvc-170>

<https://acode.com.ua/urok-69-operator-goto/>

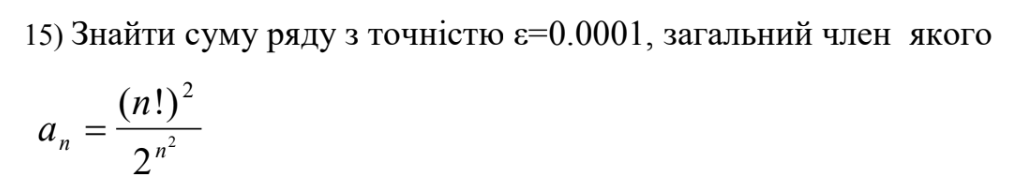
<https://drive.google.com/drive/u/1/folders/1RteNrVnAsO5jXanknOc2T7LesLT7Vspx>

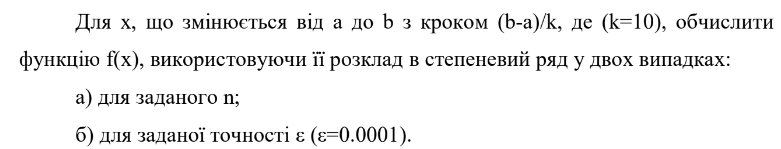
<https://learn.microsoft.com/ru-ru/cpp/c-runtime-library/reference/va-arg-va-copy-va-end-va-start?view=msvc-170>

1. Що опрацьовано: Поняття рекурсії, перевантаження функції, вкладені цикли
2. Статус: Ознайомлений

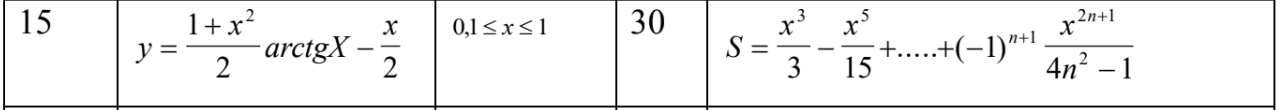
**Виконання роботи:**

* 1. **Опрацювання завдання та вимог до програм та середовища:**
* Task 2 - Requirements management and design activities with Draw.io and Google Docs
  + Створити блок-схеми до кодів
* Task 3: Lab# programming: VNS Lab 2

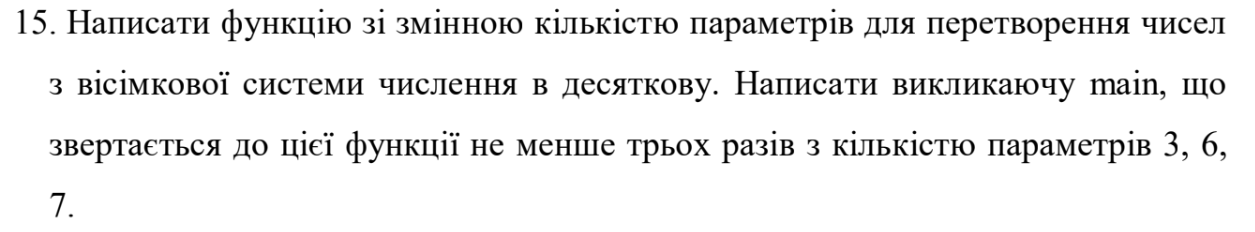


* Task 4: Lab# programming: VNS Lab 3





* Task 5: Lab# programming: VNS Lab 7 task 1



* Task 6: Practice# programming: Class Practice Task
  + Ви створюєте просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.
* Task 7: Practice# programming: Self Practice Task
  + Депутатські гроші

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?

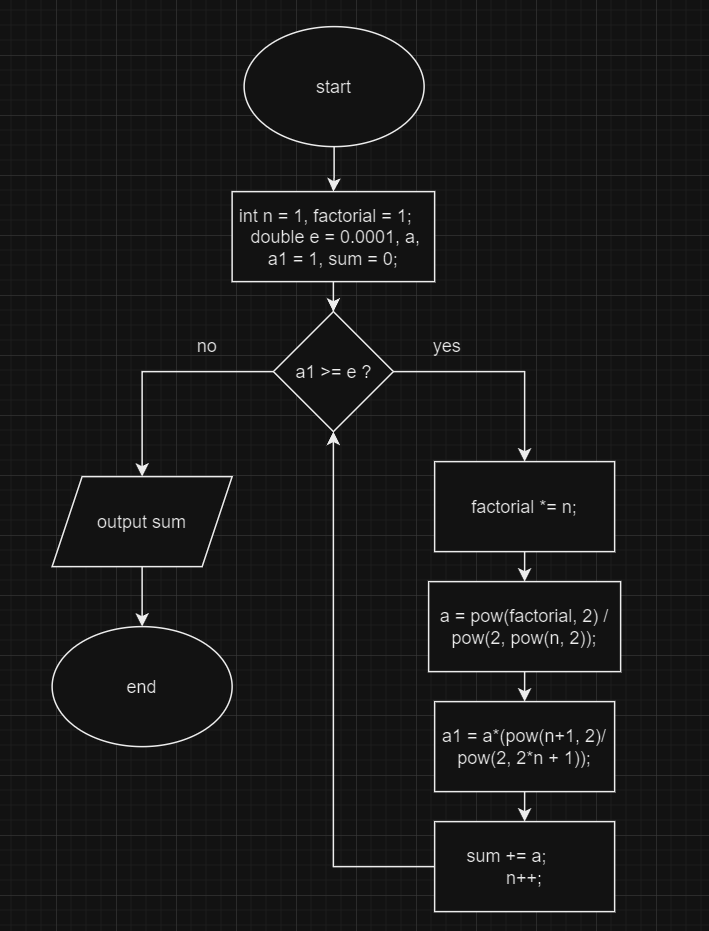
Marichka and cookies

Zenyk and Marichka decided to start hiking with scouts. Hike is very important thing. It is necessary to stock up on food products and distribute their consumption over the days so that there is enough for everyone. This time, Zenyk makes sure that there are enough cookies until the last day of the hike. Zenyk knows exactly how many packs of cookies should be left each day, and he counts them every evening. If Zenyk sees that there are fewer packs left than what should be left according to his calculations, he will definitely find the person who ate too many cookies and punish him.

Marichka likes cookies so much. Today, when all the scouts leave their tents and go swim in the river, Marichka plans to quietly eat some cookies. Of course, Marichka does not want to be punished and is very afraid that Zenyk will notice the loss. Marichka looked to see how many packs of cookies are in Zenyk’s backpack. She also knows how many cookies are in each pack. Marichka can’t wait to find out how many cookies she can eat without Zenyk noticing. Zenyk will notice the disappearance of cookies from a certain pack when and only when Marichka completely empties it.

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

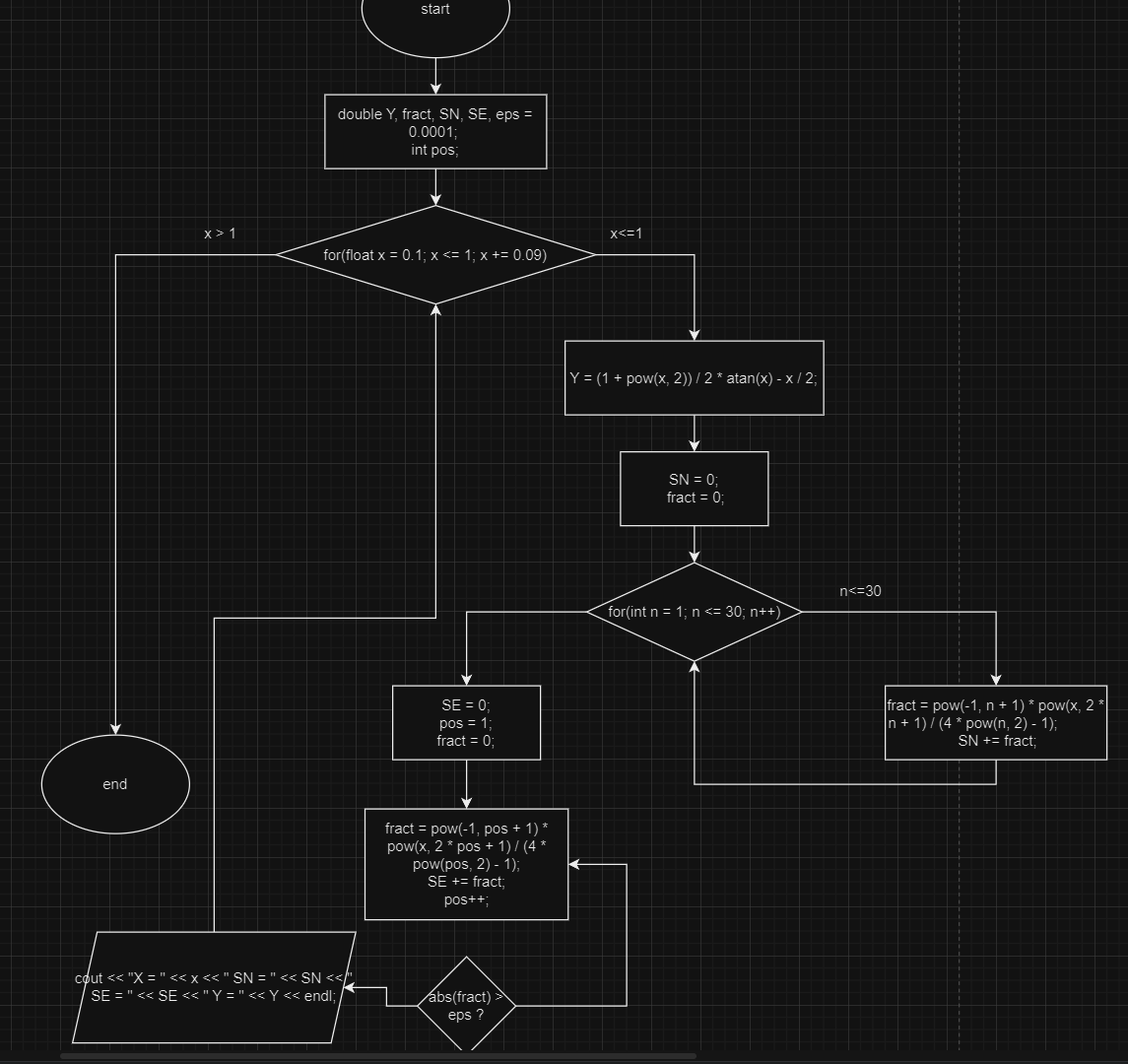
* Task 3  Lab# programming: VNS Lab 2



*Рисунок 1: Блок схема до програми №1*

Запланований час: година

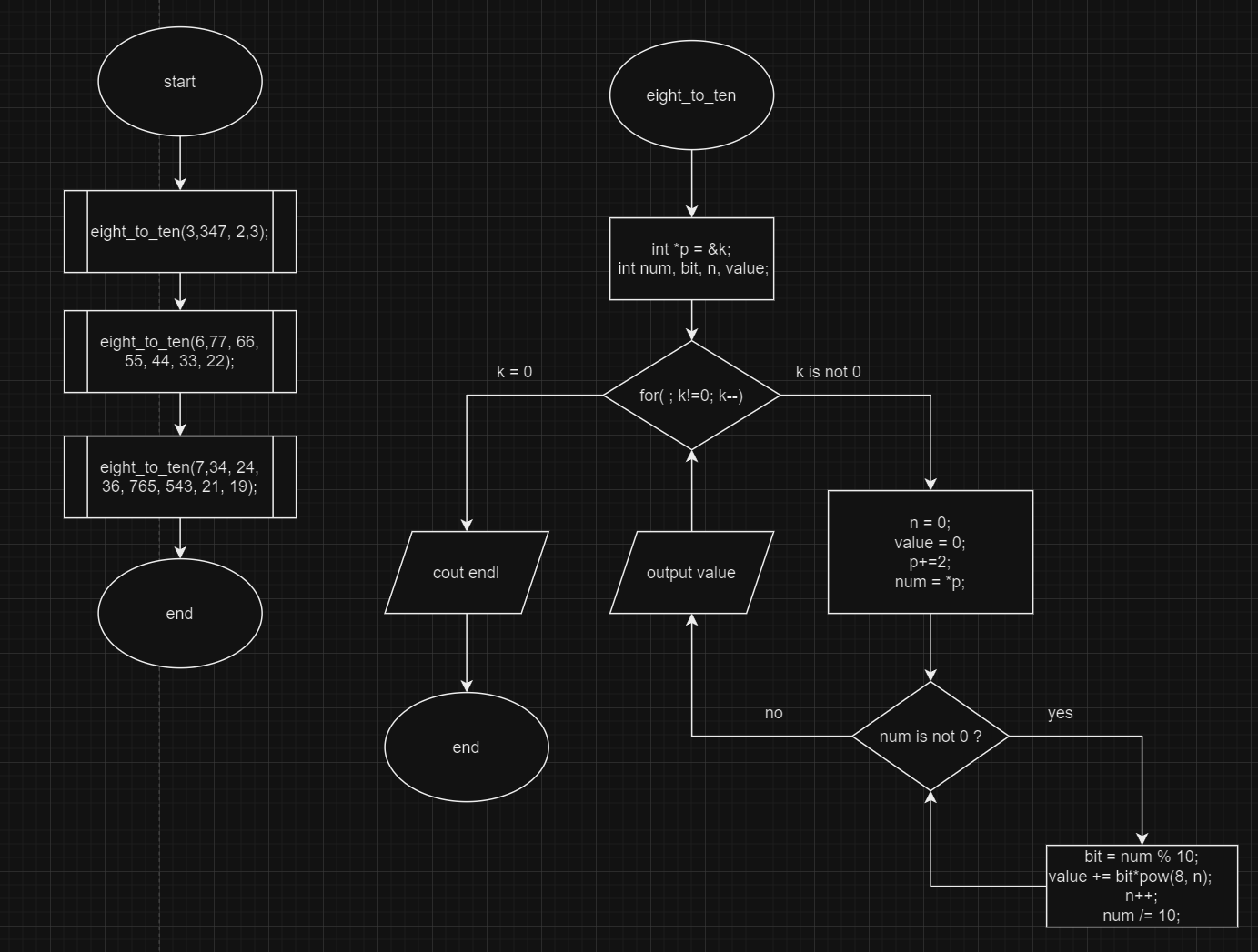
* Task 4: Lab# programming: VNS Lab 3



*Рисунок 2: Блок схема до програми №2*

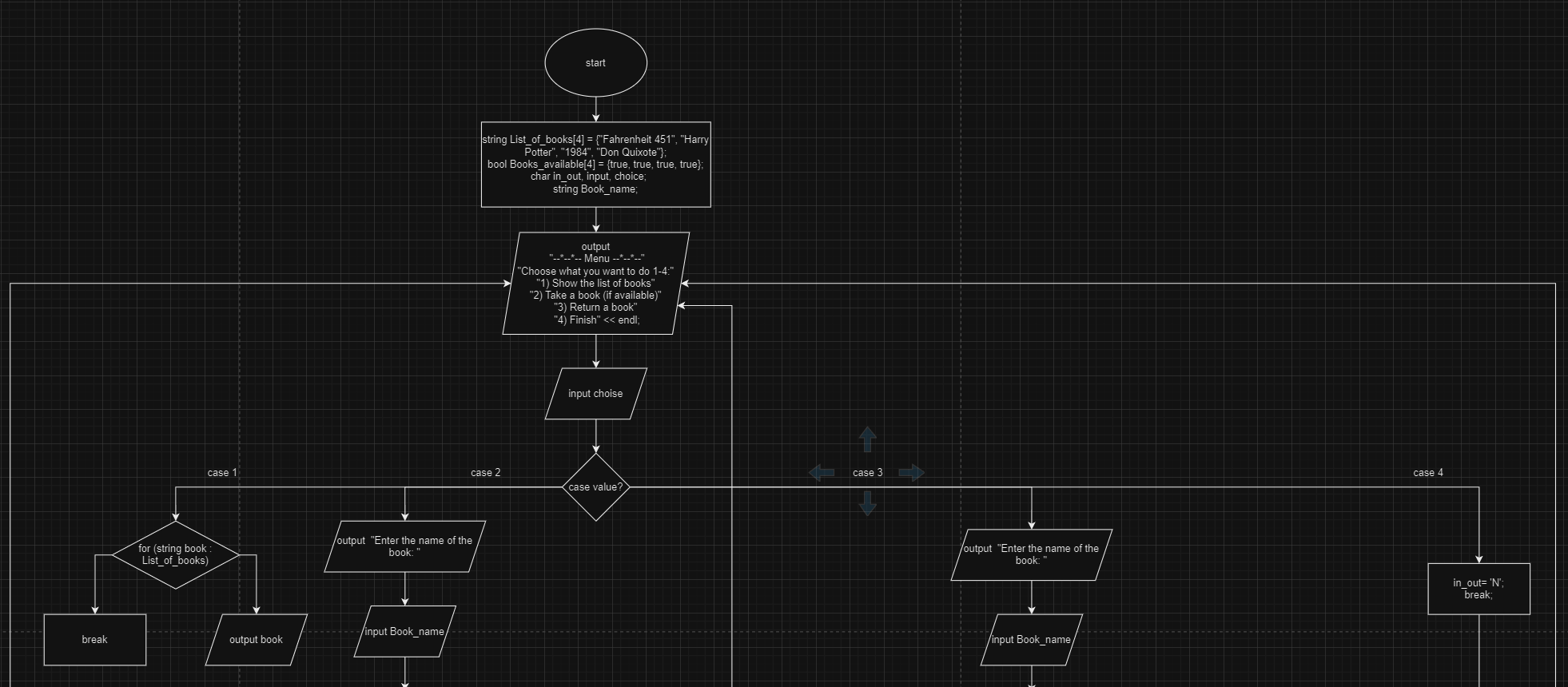
Запланований час: година

* Task 5: Lab# programming: VNS Lab 7

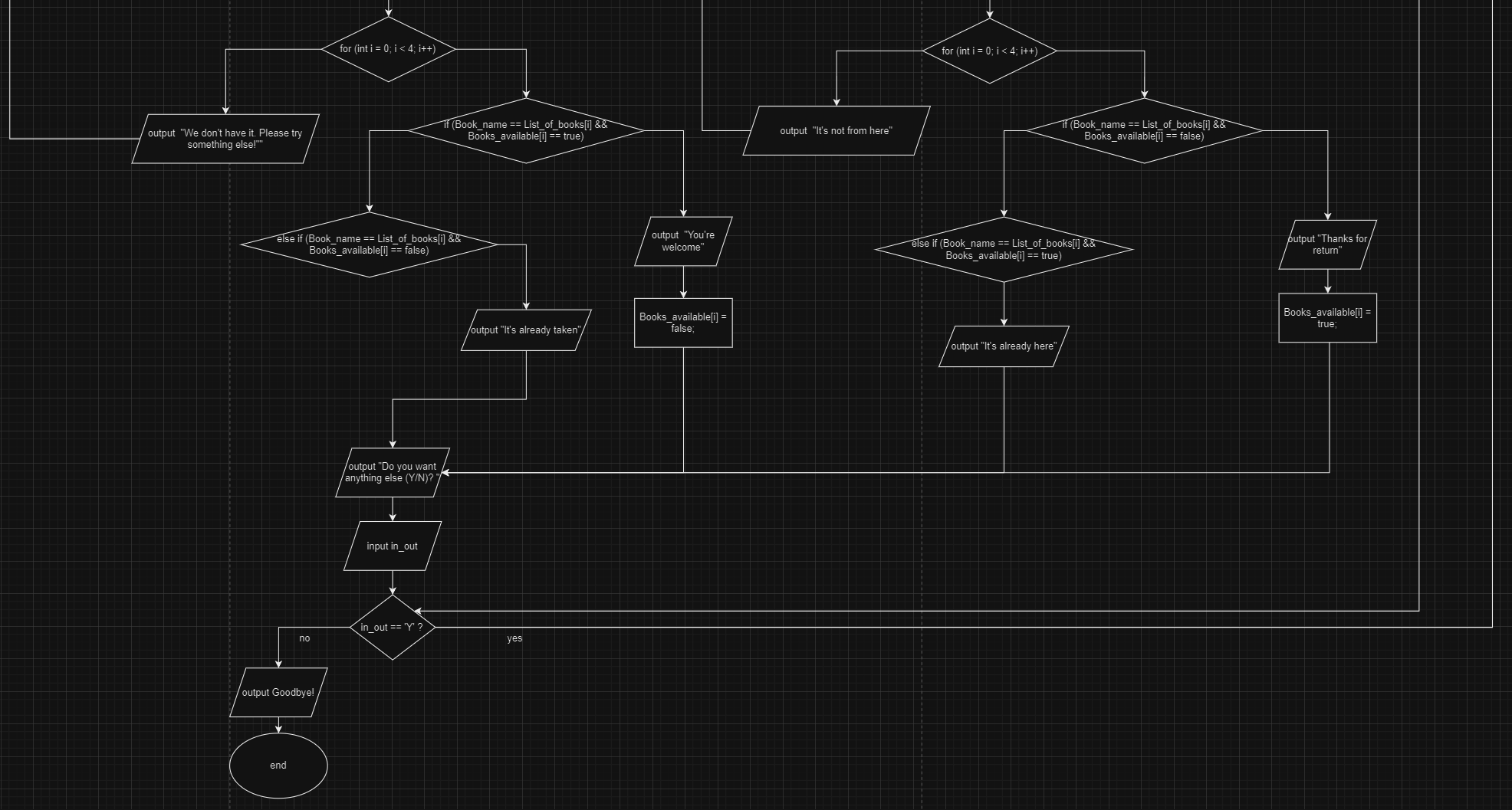


*Рисунок 3: Блок схема до програми №3*

* Task 6: Practice# programming: Class Practice Task



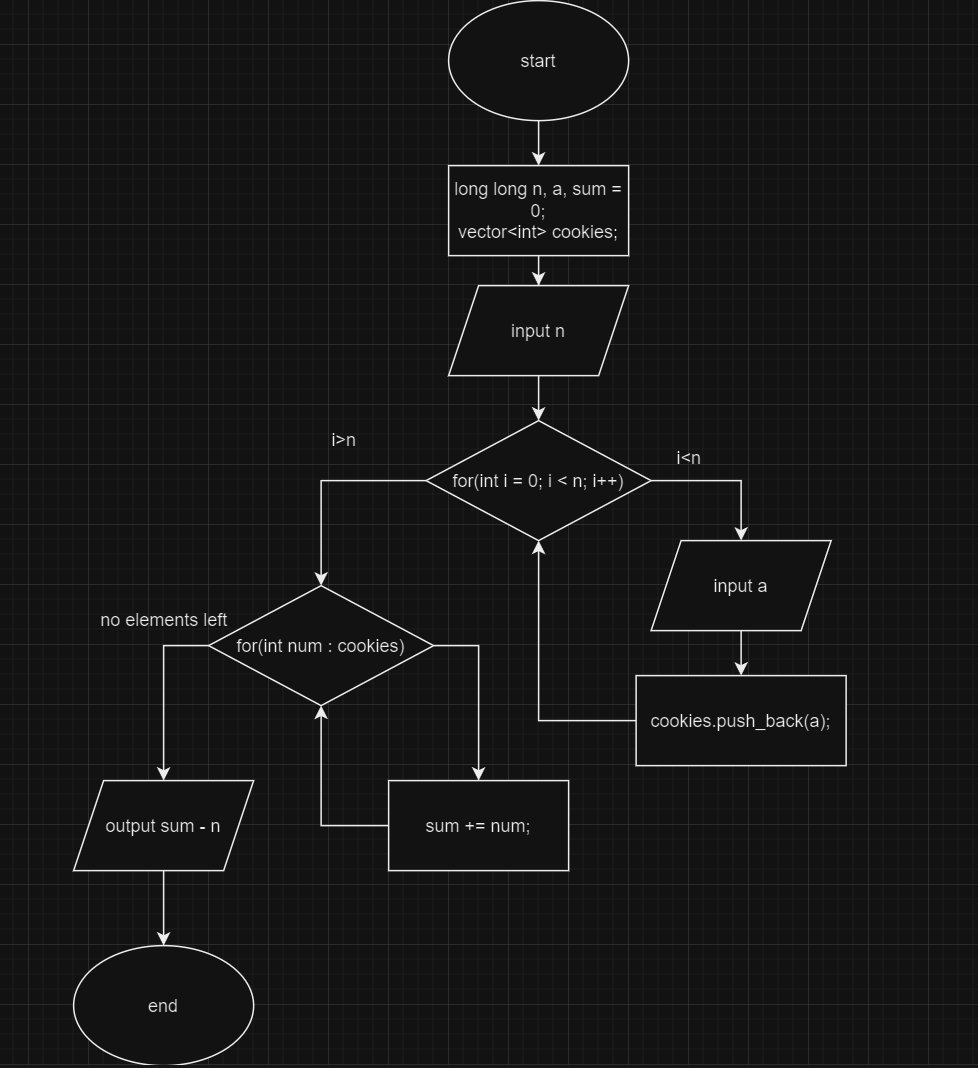
*Рисунок 4: Блок схема до програми № 4*

**

*Рисунок 5: Блок схема до програми № 4*

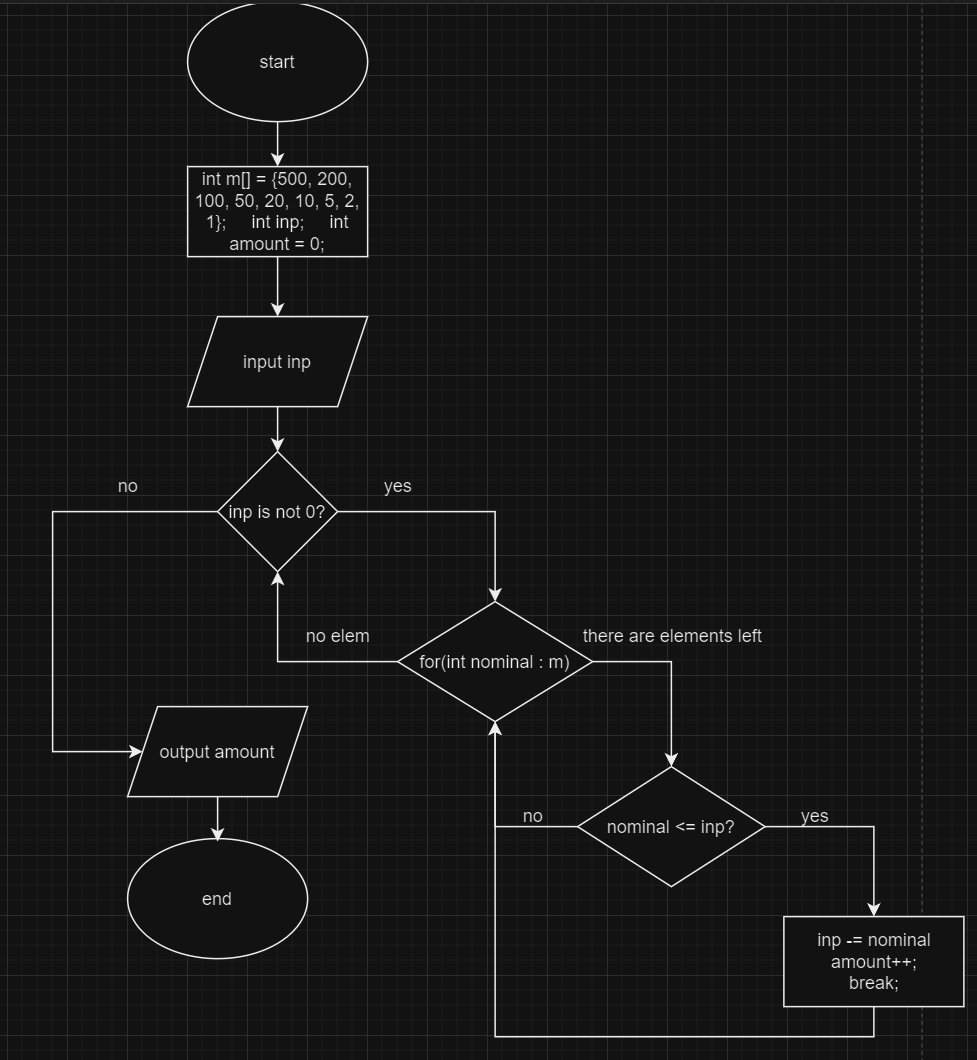
Запланований час: 2 години

* Task 7: Practice# programming: Self Practice Task



Запланований час: пів години

*Рисунок 6: Блок схема до програми № 5*

**

*Рисунок 7: Блок схема до програми № 6*

Запланований час: пів години

* 1. **Код програм з посиланням на зовнішні ресурси:**
* Task 3  Lab# programming: VNS Lab 2
* #include <iostream>
* #include <cmath>
* using namespace std;
* int main()
* {
* int n = 1, factorial = 1;
* double e = 0.0001, a, a1 = 1, sum = 0;
* while(a1 >= e){
* factorial \*= n;
* a = pow(factorial, 2) / pow(2, pow(n, 2));
* a1 = a\*(pow(n+1, 2)/ pow(2, 2\*n + 1));
* sum += a;
* n++;
* }
* cout << sum << endl;
* }
* <https://drive.google.com/drive/u/1/folders/1RteNrVnAsO5jXanknOc2T7LesLT7Vspx>
* Task 4: Lab# programming: VNS Lab 3

#include <iostream>

#include <cmath>

using namespace std;

int main(){

double Y, fract, SN, SE, eps = 0.0001;

int pos;

for(float x = 0.1; x <= 1; x += 0.09){

Y = (1 + pow(x, 2)) / 2 \* atan(x) - x / 2;

SN = 0;

fract = 0;

for(int n = 1; n <= 30; n++){

fract = pow(-1, n + 1) \* pow(x, 2 \* n + 1) / (4 \* pow(n, 2) - 1);

SN += fract;

}

SE = 0;

pos = 1;

fract = 0;

do{

fract = pow(-1, pos + 1) \* pow(x, 2 \* pos + 1) / (4 \* pow(pos, 2) - 1);

SE += fract;

pos++;

} while(abs(fract) > eps);

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

}

return 0;

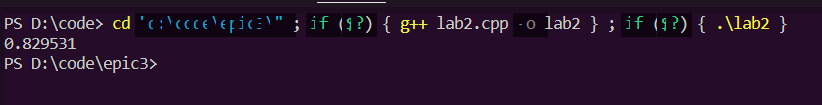
}

<https://drive.google.com/drive/u/1/folders/1RteNrVnAsO5jXanknOc2T7LesLT7Vspx>

* Task 5: Lab# programming: VNS Lab 7
* #include <iostream>
* #include <cmath>
* using namespace std;
* void eight\_to\_ten(int k, ...){
* int \*p = &k;
* int num, bit, n, value;
* for( ; k!=0; k--){
* n = 0;
* value = 0;
* p+=2;
* num = \*p;
* while(num != 0){
* bit = num % 10;
* value += bit\*pow(8, n);
* n++;
* num /= 10;
* }
* cout << value << " ";
* }
* cout << endl;
* }
* int main(){
* eight\_to\_ten(3,347, 2,3);
* eight\_to\_ten(6,77, 66, 55, 44, 33, 22);
* eight\_to\_ten(7,34, 24, 36, 765, 543, 21, 19);
* return 0;
* }

<https://drive.google.com/drive/u/1/folders/1RteNrVnAsO5jXanknOc2T7LesLT7Vspx>

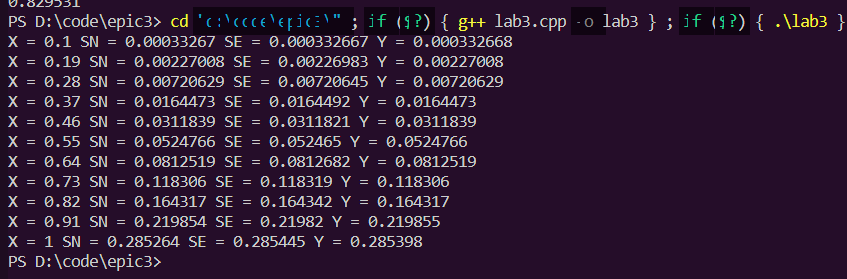
* Task 6: Practice# programming: Class Practice Task
* #include <iostream>
* #include <string>
* using namespace std;
* int main() {
* string List\_of\_books[4] = {"Fahrenheit 451", "Harry Potter", "1984", "Don Quixote"};
* bool Books\_available[4] = {true, true, true, true};
* char in\_out, choice;
* string Book\_name;
* do {
* menu:
* cout << "--\*--\*-- Menu --\*--\*--" << endl;
* cout << "Choose what you want to do 1-4:" << endl;
* cout << "1) Show the list of books" << endl;
* cout << "2) Take a book (if available)" << endl;
* cout << "3) Return a book" << endl;
* cout << "4) Finish" << endl;
* cin >> choice;
* switch (choice) {
* case '1':
* for (string book : List\_of\_books) {
* cout << book << endl;
* }
* break;
* case '2':
* cout << "Enter the name of the book: ";
* cin.ignore();
* getline(cin, Book\_name);
* for (int i = 0; i < 4; i++) {
* if (Book\_name == List\_of\_books[i] && Books\_available[i] == true) {
* cout << "You're welcome" << endl;
* Books\_available[i] = false;
* goto continue\_menu;
* }
* else if (Book\_name == List\_of\_books[i] && Books\_available[i] == false) {
* cout << "It's already taken" << endl;
* goto continue\_menu;
* }
* }
* cout << "We don't have it. Please try something else!" << endl;
* goto menu;
* case '3':
* cout << "Enter the name of the book: ";
* cin.ignore();
* getline(cin, Book\_name);
* for (int i = 0; i < 4; i++) {
* if (Book\_name == List\_of\_books[i] && Books\_available[i] == false) {
* cout << "Thanks for return" << endl;
* Books\_available[i] = true;
* goto continue\_menu;
* }
* else if (Book\_name == List\_of\_books[i] && Books\_available[i] == true) {
* cout << "It's already here" << endl;
* goto continue\_menu;
* }
* }
* cout << "It's not from here" << endl;
* goto menu;
* case '4':
* in\_out = 'N';
* break;
* default:
* cout << "Invalid choice. Please enter a number between 1 and 4." << endl;
* break;
* }
* continue\_menu:
* cout << "Do you want anything else (Y/N)? ";
* cin >> in\_out;
* } while (in\_out == 'Y');
* cout << "Goodbye!" << endl;
* return 0;
* }
* <https://docs.google.com/document/d/16b6EwQ-uoTBiPCZ2XhK42GSX2l5mnqd3hAZO78P9jtQ/edit>
* <https://acode.com.ua/urok-69-operator-goto/>
* Task 7: Practice# programming: Self Practice Task
* #include <iostream>
* #include <vector>
* using namespace std;
* int main(){
* long long n, a, sum = 0;
* vector<int> cookies;
* cin >> n;
* for(int i = 0; i < n; i++){
* cin >> a;
* cookies.push\_back(a);
* }
* for(int num : cookies){
* sum += num;
* }
* cout << sum - n;
* return 0;
* }
* #include <iostream>
* using namespace std;
* int main(){
* int m[] = {500, 200, 100, 50, 20, 10, 5, 2, 1};
* int inp;
* int amount = 0;
* cin >> inp;
* while(inp != 0){
* for(int nominal : m){
* if(nominal <= inp){
* inp -= nominal;
* amount++;
* break;
* }
* }
* }
* cout << amount;
* return 0;
* }
  1. **Результати виконання завдань, тестування та фактично затрачений час:**
* Task 3  Lab# programming: VNS Lab 2



*Рисунок 8: Результат програми № 1*

Затрачено пів години

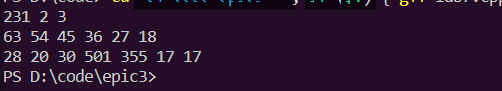
* Task 4: Lab# programming: VNS Lab 3



*Рисунок 9: Результат програми № 2*

Затрачено пів години

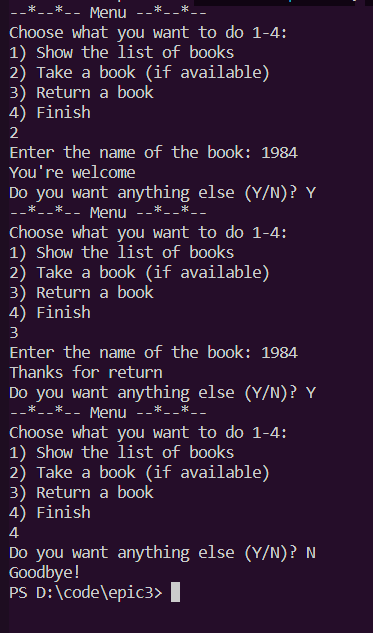
* Task 5: Lab# programming: VNS Lab 7



*Рисунок 10: Результат програми № 3*

Затрачено пів години

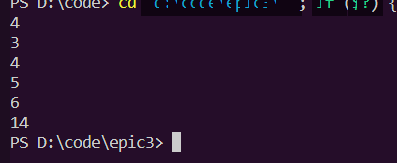
* Task 6: Practice# programming: Class Practice Task



*Рисунок 11: Результат програми № 4*

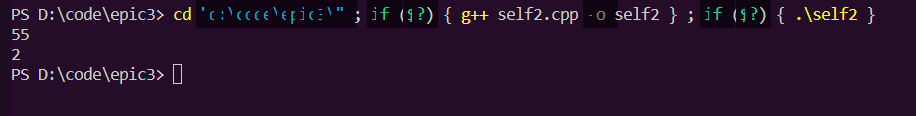
Затрачено півтори години

* Task 7: Practice# programming: Self Practice Task



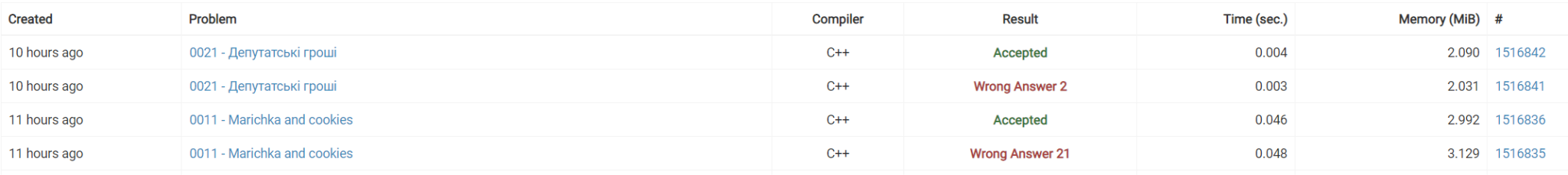
*Рисунок 12: Результат програми № 5*

Затрачено 15 хв



*Рисунок 13: Результат програми № 6*

Затрачено 15 хв



*Рисунок 14: Результат програм № 5 і 6*

**Посилання на pull request**

[**https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/522**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/522)

**Висновки:** Навчився використовувати функції зі змінною кількістю параметрів, тобто перевантаження функції, рекурсії, глибше ознайомився з поінтерами та референсами. Використовував на практиці goto, вкладені цикли.