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

***Виконав:***

студент групи ШІ-12

Воробець Максим Володимирович

# **Тема роботи:**

# Цикли. Вкладені Цикли. Функції. Перевантаження функцій. Рекурсія.

# **Мета роботи:**

Ознайомитися з концепцією циклів та їхнім використанням для повторення операцій у програмах. Вивчити концепцію функцій як засобу структуризації коду. Також ознайомитись із концепцією перевантаження функцій, де функція може мати кілька визначень з різними параметрами. Застосувати набуті знання на практиці, розв'язуючи практичні завдання.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Цикли for, while та do while.
* Тема №2: Функції.
* Тема №3: Перевантаження функцій.
* Тема №4: Рекурсія.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Цикли for, while та do while.
  + Джерела Інформації
    - Відео:

[Базовий курс С++ (Урок 5. Цикли: FOR, WHILE, DO WHILE) ітерація, тіло циклу, безкінченний цикл (youtube.com)](https://www.youtube.com/watch?v=RBqs1GGvmNU)

* + - Стаття:

[C++. Цикли. Оператори циклу for, while, do...while | BestProg](https://www.bestprog.net/uk/2017/09/04/cycles-operators-of-the-cycle-for-while-do-while_ua/)

* + Що опрацьовано:

Опрацьовано основи циклічного виконання операцій у програмах та використано їх для вирішення різноманітних задач.

* + Статус: Ознайомлений
  + Початок опрацювання теми: 11.11.2023
  + Звершення опрацювання теми: 12.11.2023
* Тема №2: Функції
  + Джерела Інформації:
    - Стаття:

[C++. Функції. Приклади використання. Параметри функції | BestProg](https://www.bestprog.net/uk/2017/04/15/%d1%84%d1%83%d0%bd%d0%ba%d1%86%d1%96%d1%97-%d1%87%d0%b0%d1%81%d1%82%d0%b8%d0%bd%d0%b0-1-%d0%be%d0%bf%d0%b8%d1%81-%d1%84%d1%83%d0%bd%d0%ba%d1%86%d1%96%d1%97-%d1%84%d0%b0%d0%ba%d1%82%d0%b8%d1%87/)

[Функції в С++ / aCode](https://acode.com.ua/urok-15-funktsiyi-i-operator-return/)

* + Що опрацьовано:

Опрацьовано матеріал та практичні навички з теми “Функції”

* + Статус: Ознайомлений
  + Початок опрацювання теми: 18.11.2023
  + Звершення опрацювання теми: 20.11.2023
* Тема №3: Перевантаження функцій.
  + Джерела Інформації:
    - Відео:

<https://www.youtube.com/watch?v=3KJfisev6SI&pp=ygUS0KTQo9Cd0JrQptCG0IcgYysr>

* + Що опрацьовано:

Опрацьовано матеріал по даній темі.

* + Статус: Ознайомлений
  + Початок опрацювання теми: 20.11.2023
  + Звершення опрацювання теми: 21.11.2023
* Тема №4 Рекурсія.
  + Джерела Інформації:
    - Відео:

<https://www.youtube.com/watch?v=V7q9w_s0nns>

* + - Стаття:

[C++. Приклади розв'язку задач на рекурсію | BestProg](https://www.bestprog.net/uk/2019/01/13/examples-of-tasks-solving-on-recursion-in-the-c-programming-language-ua/#q01)

[Рекурсія і Числа Фібоначчі в С++ / Уроки по С++ / aCode](https://acode.com.ua/urok-113-rekursiya-i-chysla-fibonachchi/)

* + Що опрацьовано:

Опрацьовано теоретичний матеріал та добре освоєні практичні навчики по даній темі.

* + Статус: Ознайомлений
  + Початок опрацювання теми: 25.11.2023
  + Звершення опрацювання теми: 27.11.2023

# **Виконання роботи:**

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

Завдання №1 **VNS Lab 2 - Task 1-N**

* Варіант завдання: 3
* Деталі завдання:

Знайти суму цілих додатніх непарних чисел, менших 200.

Завдання №2 **VNS Lab 3 - Task 1-N**

* Варіант завдання: 3
* Деталі завдання:

Для х, що змінюється від a до b з кроком (b-a)/k, де (k=10), обчислити функцію f(x), використовуючи її розклад в степеневий ряд у двох випадках: а) для заданого n; б) для заданої точності ε (ε=0.0001). Для порівняння знайти точне значення функції.

Завдання №3 **VNS Lab 7 - Task 1-N**

* Варіант завдання: 3
* Деталі завдання:

Написати функцію sum зі змінною кількістю параметрів, що знаходить суму чисел типу int за даною формулою. Написати викликаючу функцію main, що звертається до функції sum не менше трьох разів з кількістю параметрів 5, 10, 12.

Завдання №4 **VNS Lab 7 - Task 2-N**

* Варіант завдання: 3
* Деталі завдання:

Написати перевантажені функції й основну програму, що їх викликає.

а) для множення цілих чисел;

б) для множення комплексних чисел.

Завдання №5 **Class Practice Work**

* Деталі завдання:

Cтворити програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути. Програма повинна вмітиперерахувати всі книги, дозволити взяти книгу (за наявності), **д**озволити повернення книги.

* Важливі деталі для врахування в імплементації програми:

Використати масив або вектор для зберігання назв книг. Використати інший масив або вектор для збереження стану доступності кожної книги.

Завдання №6 **Self Practice Work**

* Деталі завдання:

<https://algotester.com/en/ArchiveProblem/DisplayWithEditor/2>

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

Програма №1 **VNS Lab 2 - Task 1-N**

* Планований час на реалізацію: 30хв

Програма №2 **VNS Lab 3 - Task 1-N**

* Блок-схема

Зображення, що містить текст, знімок екрана, Шрифт, графічний дизайн

Автоматично згенерований опис

Рисунок . Блок-схема до завдання №2

* Планований час на реалізацію: 1год 30хв

Програма №3 **VNS Lab 7 - Task 1-N**

* Планований час на реалізацію: 45хв

Програма №4 **VNS Lab 7 - Task 2-N**

* Планований час на реалізацію: 45хв

Програма №5 **Class Practice Work**

* Планований час на реалізацію: 1год 45хв

Програма №6 **Self Practice Work**

* Планований час на реалізацію: 40хв

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

Завдання №1 **VNS Lab 2 - Task 1-N** Код написаний за допомогою циклу for(i=1), який дає нам змогу до значення додавати 2, допоки не дойде до числа 200. Програма обчислює суму всіх непарних цілих додатніх чисел, які менше за 200, і виводить цю суму на екран.

#include <stdio.h>

int main(){

    int x=0;

    for(int i=1; i<200; i+=2)

     {

        x+=i;

    }

     printf("Сума цілих додатніх непарних чисел, менших за 200: %d\n", x);

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/1d0eab0e3eb9de221fda4cc3aef1ed71d62a98c1/ai_12/maksym_vorobets/Epic3/vns_lab_2_task_maksym_vorobets.cpp>

Завдання №2 **VNS Lab 3 - Task 1-N** Цей код написания для апроксимації значення синус-функції за допомогою ряду Тейлора. Я використав функцію fact для знаходження факторіалу, цикл for та while для знаходження значення. Ця програма працює коректно та виконує математичну дію за даною формулою.

#include <iostream>

#include <math.h>

#include<iomanip>

using namespace std;

int fact(int n){

 if(n==0||n==1){return 1;}

    return n\*fact(n-1);

}

int main(){

    double x, s, a=0.1, b=1.0, SN, SE, Y, E=0.0001;

    cout << fixed << setprecision(5);

    int n=10, k=10;

        for(double i=a;i<=b;i+=(b-a)/k){

        x=i;

        cout <<"X="<< x << "    ";

        SN=x;

    for(int i=0; i<=n; i++){

        SN+=pow((-1),i)\*pow(x,2\*i+1)/fact(2\*i+1);

    }

     cout <<"SN="<< SN << "    ";

        s=x;

        int m=1;

   while (E<=abs(s)){

    s=pow((-1),m)\*pow(x,2\*m+1)/fact(2\*m+1);

    SE+=s;

    m++;

   }

   cout <<"SE="<< SE << "    ";

   Y=sin(x);

   cout <<"Y="<< Y << "    "<<endl;

}

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/1d0eab0e3eb9de221fda4cc3aef1ed71d62a98c1/ai_12/maksym_vorobets/Epic3/vns_lab_3_task_maksym_vorobets.cpp>

Завдання №3 **VNS Lab 7 - Task 1-N** У цій програмі ми використали дві функції: sum та main. Фунція sum приймає кількість параметрів **k** та використовує вказівник **p**, щоб пройти через аргументи, таким чином ми обчислимо суму добутків пар сусідніх елементів, використовуючи цикл for. Функція main викликає функцію sum тричі з різною кількістю аргументів та виводить результат в термінал.

#include <stdio.h>

#include <stdarg.h>

int sum(int k, ...){

   int s = 0;

   int \*p=&k;

   int a=\*(++p);

   for(int i = 0; i < k-1; i++){

   int a1=\*(++p);

   int a2=\*(++p+1);

   s+=a1\*a2;

   }

   return s;

}

int main() {

    int s5 = sum(5, 2, 3, 4, 5, 1);

    printf("S5=%d\n", s5);

    int s10 = sum(10, 1, 2, 3, 4, 2, 5, 1, 3, 7, 1);

    printf("S10=%d\n", s10);

    int s12 = sum(12, 2, 3, 4, 5, 6, 7, 8, 9, 2, 1, 2, 3);

    printf("S12=%d\n", s12);

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/1d0eab0e3eb9de221fda4cc3aef1ed71d62a98c1/ai_12/maksym_vorobets/Epic3/vns_lab_7_task_1_variant_3_maksym_vorobets.cpp>

Завдання №4 **VNS Lab 7 - Task 2-N** У даному коді є дві функції comp: одна для роботи з комплексними числами, інша для роботи з цілими числами. У функції main ці функції викликаються з конкретними аргументами. Функції виводять результати на екран. Функція для комплексних чисел використовує бібліотеку <complex>, яка дозволяє працювати з комплексними числами.

#include <iostream>

#include <complex>

using namespace std;

   double comp(double x1, double x2, double y1, double y2);

   int comp(int a, int b);

int main(){

    comp(1.0, 6.0, 2.0, 4.0);

    comp(23, 5);

 return 0;

}

    double comp(double x1, double x2, double y1, double y2){

    complex< double > z1(x1, x2); //1+6i//

    complex< double > z2(y1, y2);//2+4i//

    cout<<z1\*z2<<endl;

    return 0;

}

    int comp(int a, int b){

    int s;

    s=a\*b;

    cout<<s<<endl;

    return s;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/1d0eab0e3eb9de221fda4cc3aef1ed71d62a98c1/ai_12/maksym_vorobets/Epic3/vns_lab_7_task_2_variant_3_maksym_vorobets.cpp>

Завдання №5 **Class Practice Work** Це проста програма керування бібліотекою. Програма вміє перечислювати всі книги, дає користувачу змогу вибрати книгу та віддати книгу. Код написаний за допомогою масивів для зберігання назв книг та для збереження стану доступності кожної книги. Також для написання коду використано умовні оператори if-else, цикли for, do while та goto, щоб перенаправити користувача до головного меню.

#include <iostream>

#include <string.h>

using namespace std;

int main(){

    string books[]={"Harry\_Potter", "Aftershocks", "Misery", "Icebreaker", "The\_Committed"};

    bool available[] = {true , true,true, true, true};

    string operation, availability, bookName, rewiew, exit;

    Menu:

        do{

        cout << "Введіть операцію:" << endl;

        cout << "1. Огляд книг" << endl;

        cout << "2. Вибрати книгу" << endl;

        cout << "3. Вернути книгу"<< endl;

        cout << "4. Вийти" << endl;

        cin >> operation;

    if(operation=="1")

    {

        for(int i=0; i<5; i++)

            cout << books[i] << " "<<endl;

          }

          else if(operation=="2")

        {

             cout << "Виберіть:" <<endl;

             cin >> bookName;

           for (int i = 0; i < 5; i++)

              {

            if(bookName==books[i])

            {

                if(available[i])

                {

                    cout << "Вибрано " << endl;

                    available[i]=false;

               }

                else{

                cout <<"Немає в наявності"<< endl;

              break;

                }

            }

              }

        }

             else if(operation=="3")

             {

                cout<<"Яку Ви хочете повернути книгу"<<endl;

                cin>>bookName;

            for (int i = 0; i < 5; i++)

           if(bookName==books[i])

        {

                available[i]=true;

            break;

        }

             }

             else if (operation=="4")

             {

            cout<<"Ви точно хочете вийти?"<<endl;

            cin>>exit;

            if(exit=="No"){

                goto Menu;

            }else;

             }

           }

        while (exit!="Yes");

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/1d0eab0e3eb9de221fda4cc3aef1ed71d62a98c1/ai_12/maksym_vorobets/Epic3/practice_work_team_tasks_maksym_vorobets.cpp>

Завдання №6 **Self Practice Work** Код написаний для того, щоб програма вивела максимальну кількість штук печива, яку зможе з’їсти Марічка. Для цього використано масив через цикл for та умовний оператор if, щоб розв’язати дану задачку.

#include <iostream>

using namespace std;

int main(){

    int n;

    long s=0;

    cin>>n;

int a[n];

    for(int i=0; i<n; i++){

        cin>>a[i];

        if(a[i]>0){

        s +=a[i]-1;

        }else

        {s=0;}

    }

     cout<<s<<endl;

return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/blob/1d0eab0e3eb9de221fda4cc3aef1ed71d62a98c1/ai_12/maksym_vorobets/Epic3/practice_work_self_algotester_tasks_maksym_vorobets.cpp>

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

Завдання №1

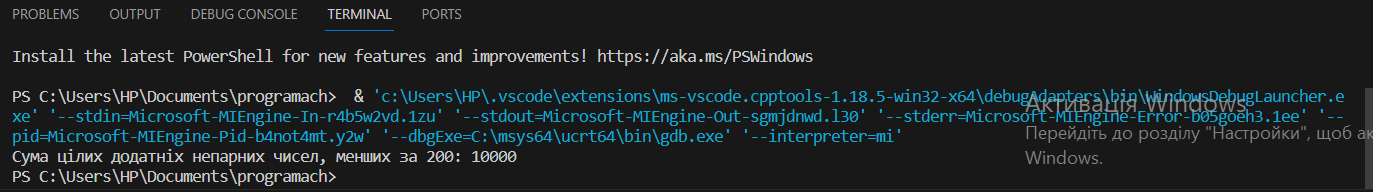


Рисунок . Результат програми №1

Час затрачений на виконання завдання: 30хв

Завдання №2

Зображення, що містить текст, знімок екрана, Мультимедійне програмне забезпечення, програмне забезпечення

Автоматично згенерований опис

Рисунок . Результат програми №2

Час затрачений на виконання завдання: 1год 30хв

Завдання №3

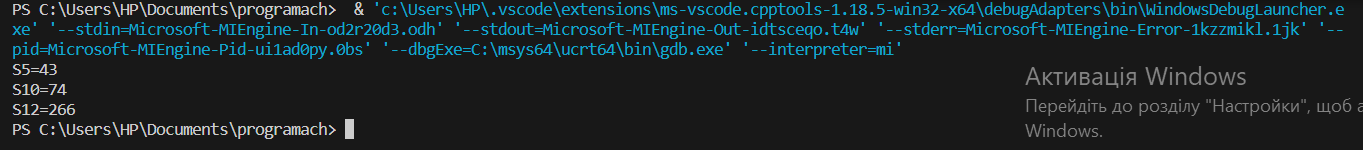


Рисунок . Результат програми №3

Час затрачений на виконання завдання: 45хв

Завдання №4

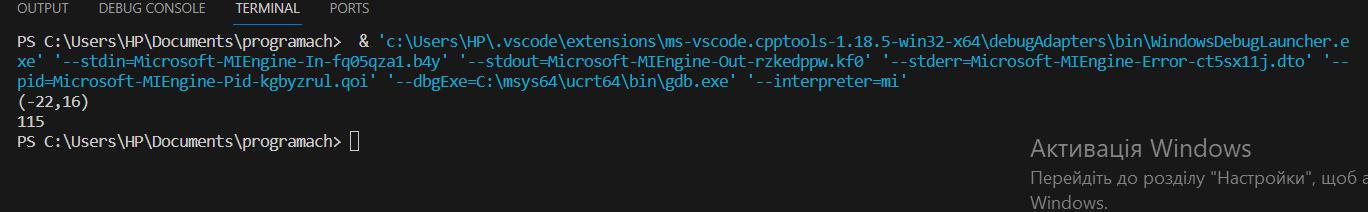


Рисунок . Результат програми №4

Час затрачений на виконання завдання: 45хв

Завдання №5

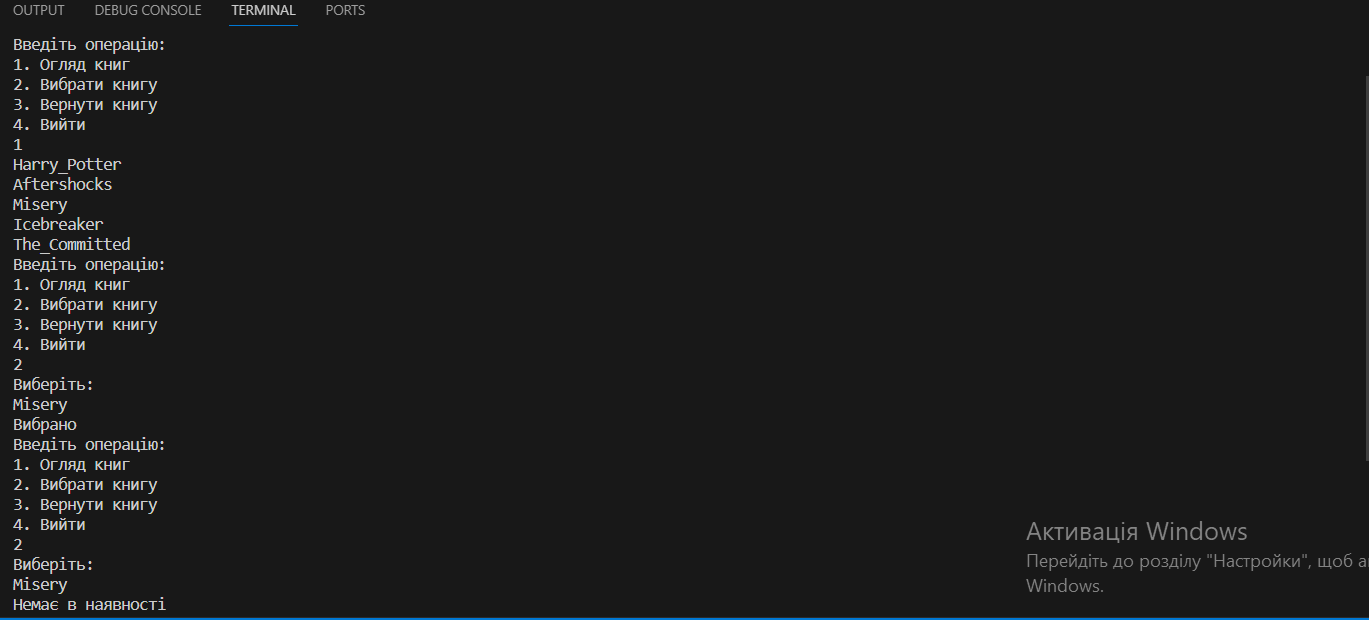


Рисунок . Результат програми №5

Час затрачений на виконання завдання: 1год 45хв

Завдання №6

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рисунок . Результат програми №6

Час затрачений на виконання завдання: 40хв

## **5. Кооперація з командою:**

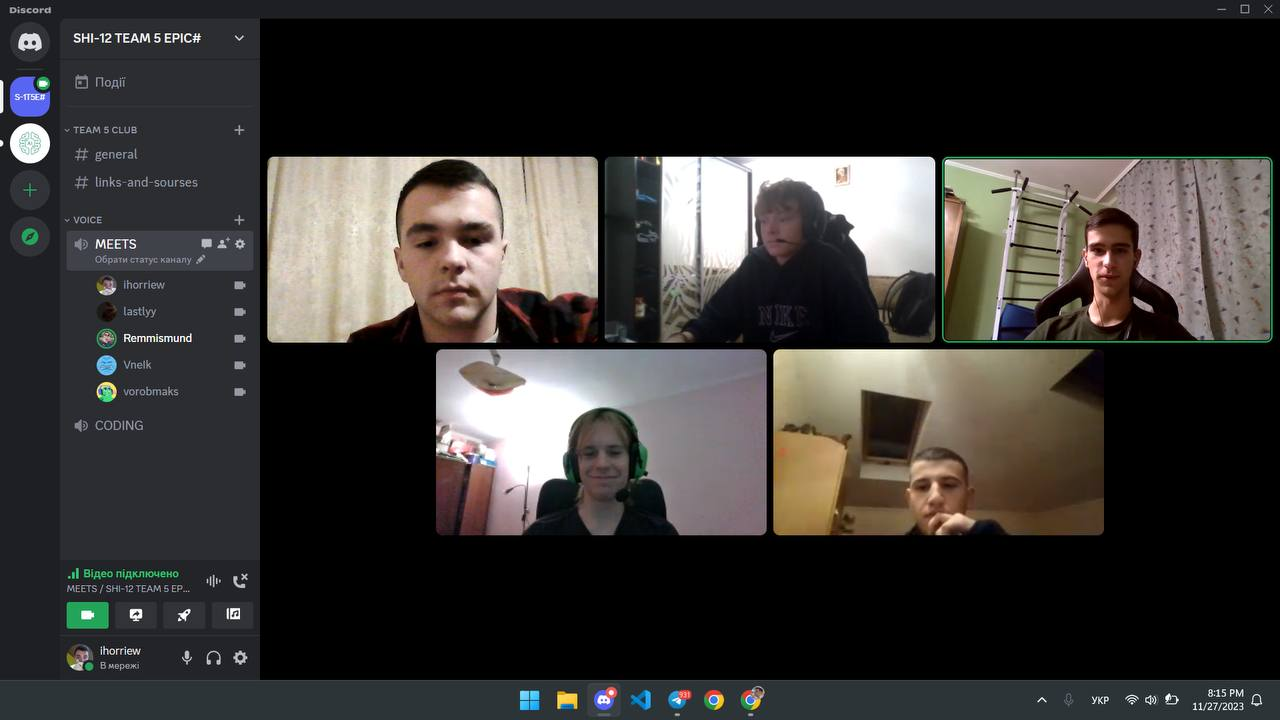


Рисунок . Обговорення з командою

Зображення, що містить текст, програмне забезпечення, Мультимедійне програмне забезпечення, Графічний редактор

Автоматично згенерований опис

Рисунок . Trello

Зображення, що містить Мультимедійне програмне забезпечення, програмне забезпечення, знімок екрана

Автоматично згенерований опис

Рисунок . Коментар

# **Висновки:**

Я ознайомився з концепцією циклів та їхнім використанням для повторення операцій у програмах, вивчив концепцію функцій як засобу структуризації коду, ознайомився із концепцією перевантаження функцій, де функція може мати кілька визначень з різними параметрами. Також команда допомогла розібратися з рекурсивними функціями. Розвинув практичні навички за допомогою лабораторних завдань, що допомогли закріпити теоретичний матеріал.