

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



## **Звіт**

**про виконання лабораторних та практичних робіт блоку № 6**

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 № 6

**Виконав:**

Студент групи ІІІ-11  
Корнілов Артем Сергійович

Львів 2024

**Тема:** Динамічні структури (Черга, Стек, Списки, Дерево).  
Алгоритми обробки динамічних структур.

**Мета:** Засвоїти основи роботи з динамічними структурами даних, такими як черга, стек, списки та дерева. Ознайомитися з алгоритмами їх обробки для розв'язання різноманітних задач.

### **Теоретичні відомості та джерела:**

#### **Динамічні структури**

- [Урок №89. Динамічне виділення пам'яті;](#)
- [Урок №90. Динамічні масиви;](#)

-Стек;

[C++ • Теорія • Урок 58 • Стек, Куча, Статична пам'ять](#)  
<https://www.youtube.com/watch?v=B3VHHfMW0Pg>

-Черга;

[#4](#)  
[#5](#)

-Списки:

[#1](#)  
[#2](#)  
[Урок #133](#)  
[Урок #134](#)  
[Урок #135](#)

-Дерева:

[#3](#)  
[C++ • Теорія • Урок 144 • ADT • Бінарне дерево](#)

## **Виконання роботи:**

### **VNS Lab 10v2:**

1. Написати функцію для створення списку. Функція може створювати порожній список, а потім додавати в нього елементи.
2. Написати функцію для друку списку. Функція повинна передбачати вивід повідомлення, якщо список порожній.
3. Написати функції для знищення й додавання елементів списку у відповідності зі своїм варіантом.
4. Виконати зміни в списку й друк списку після кожної зміни.
5. Написати функцію для запису списку у файл.
6. Написати функцію для знищення списку.
7. Записати список у файл, знищити його й виконати друк (при друці повинне бути видане повідомлення "Список порожній").
8. Написати функцію для відновлення списку з файлу.
9. Відновити список і роздрукувати його.
10. Знищити список.

Записи в лінійному списку містять ключове поле типу `int`.

Сформувані однонаправлений список. Знищити з нього елемент із заданим ключем, додати елемент перед елементом із заданим ключем;

### **Algotester Lab 5v3:**

У вас є карта гори розміром  $N \times M \times M$ .

Також ви знаєте координати  $\{x, y\}$ , у яких знаходиться вершина гори.

Ваше завдання - розмалювати карту таким чином, щоб найнижча точка мала число 0, а пік гори мав найбільше число.

Клітинки які мають суміжну сторону з вершиною мають висоту на один меншу, суміжні з ними і не розфарбовані мають ще на 1 меншу висоту і так далі.

### **Algotester Lab 7-8 v1:**

Ваше завдання - власноруч реалізувати структуру даних "Двоzv'язний список".

Ви отримаєте  $QQ$  запитів, кожен запит буде починатися зі слова-ідентифікатора, після якого йдуть його аргументи.

Вам будуть поступати запити такого типу:

- Вставка:  
Ідентифікатор - insertinsert  
Ви отримуєте ціле число indexindex елемента, на місце якого робити вставку.  
Після цього в наступному рядку рядку написане число NN - розмір списку, який треба вставити.  
У третьому рядку NN цілих чисел - список, який треба вставити на позицію indexindex.
- Видалення:  
Ідентифікатор - eraseerase  
Ви отримуєте 2 цілих числа - indexindex, індекс елемента, з якого почати видалення та nn - кількість елементів, яку треба видалити.
- Визначення розміру:  
Ідентифікатор - sizesize  
Ви не отримуєте аргументів.  
Ви виводите кількість елементів у списку.
- Отримання значення ii-го елемента  
Ідентифікатор - getget  
Ви отримуєте ціле число - indexindex, індекс елемента.  
Ви виводите значення елемента за індексом.
- Модифікація значення ii-го елемента  
Ідентифікатор - setset  
Ви отримуєте 2 цілих числа - індекс елемента, який треба змінити, та його нове значення.
- Вивід списку на екран  
Ідентифікатор - printprint  
Ви не отримуєте аргументів.  
Ви виводите усі елементи списку через пробіл.  
Реалізувати використовуючи перегрузку оператора <<

## Class Practice Task:

## Задача №1 - Реверс списку (Reverse list)

**Реалізувати метод реверсу списку:** `Node* reverse(Node *head);`

*Умови задачі:*

- використовувати цілочисельні значення в списку;
- реалізувати метод реверсу;
- реалізувати допоміжний метод виведення вхідного і обернутого списків;

## Задача №2 - Порівняння списків

`bool compare(Node *h1, Node *h2);`

*Умови задачі:*

- використовувати цілочисельні значення в списку;
- реалізувати функцію, яка ітеративно проходить по обох списках і порівнює дані в кожному вузлі;
- якщо виявлено невідповідність даних або якщо довжина списків різна (один список закінчується раніше іншого), функція повертає **false**.

## Задача №3 – Додавання великих чисел

`Node* add(Node *n1, Node *n2);`

*Умови задачі:*

- використовувати цифри від 0 до 9 для значень у списку;
- реалізувати функцію, яка обчислює суму двох чисел, які збережено в списку; молодший розряд числа записано в голові списку (напр. 379  $\Rightarrow$  9 $\rightarrow$ 7 $\rightarrow$ 3);
- функція повертає новий список, передані в функцію списки не модифікуються.

## Задача №4 - Віддзеркалення дерева

`TreeNode *create_mirror_flip(TreeNode *root);`

*Умови задачі:*

- використовувати цілі числа для значень у вузлах дерева
- реалізувати функцію, що проходить по всіх вузлах дерева і міняє місцями праву і ліву вітки дерева
- функція повертає нове дерево, передане в функцію дерево не модифікується

Задача №5 - Записати кожному батьківському вузлу суму підвузлів

```
void tree_sum(TreeNode *root);
```

*Умови задачі:*

- використовувати цілочисельні значення у вузлах дерева;
- реалізувати функцію, яка ітеративно проходить по бінарному дереві і записує у батьківський вузол суму значень підвузлів
- вузол-листок не змінює значення
- значення змінюються від листків до кореня дерева

## **Self Practice**

### **Task Algotester Lab 5v2:**

У вас є карта гори розміром  $N \times M$ .

Також ви знаєте координати  $\{x, y\}$ , у яких знаходиться вершина гори.

Ваше завдання - розмалювати карту таким чином, щоб найнижча точка мала число 0, а пік гори мав найбільше число.

Клітинки які мають суміжну сторону з вершиною мають висоту на один меншу, суміжні з ними і не розфарбовані мають ще на 1 меншу висоту і так далі.

**Код програми з посиланням на зовнішні ресурси**

### **VNS Lab 10v2:**

**затратність ~3 години**

### **Algotester Lab 5v3:**

**затратність ~ 2 години**

### **Algotester Lab 7-8 v1:**

**затратність ~ 3 години**

## **Class Practice Task:**

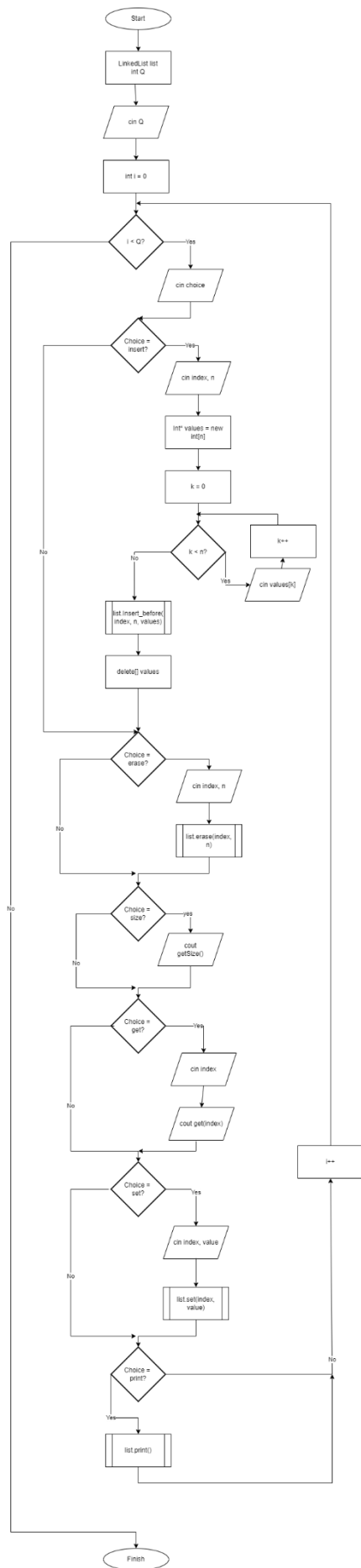
**Practice task 1**

**Practice task 2-3**

**Practice task 4-5**

**затратність ~ 4 години**

**Block-scheme for task Algotester Lab 7-8 v1**





**Результати виконаних завдань, тестування та фактично  
затрачений час**

**VNS Lab 10v1: ~4 години**

```
1 2 3 4 5
1 2 3 1 4 5
1 2 3 1 4
List succesfully inputed in file: "List.txt".
List destroyed.
List empty
List destroyed.
List succesfully restored from: "List.txt".
1 2 3 1 4
List destroyed.
PS C:\VS\projects>
```

**Algotester Lab 5v3: ~2 години**

```
7 5
3 1
6 5 4 3 2
7 6 5 4 3
8 7 6 5 4
7 6 5 4 3
6 5 4 3 2
5 4 3 2 1
4 3 2 1 0
```

Created	Compiler	Result	Time (sec.)	Memory (MiB)
a minute ago	C++ 23	Accepted	0.857	7.859

**Algotester Lab 7-8 v1: ~4 годин**

Created	Compiler	Result	Time (sec.)	Memory (MiB)
3 hours ago	C++ 23	Accepted	0.008	1.277

**Class Practice Task: ~4 години**

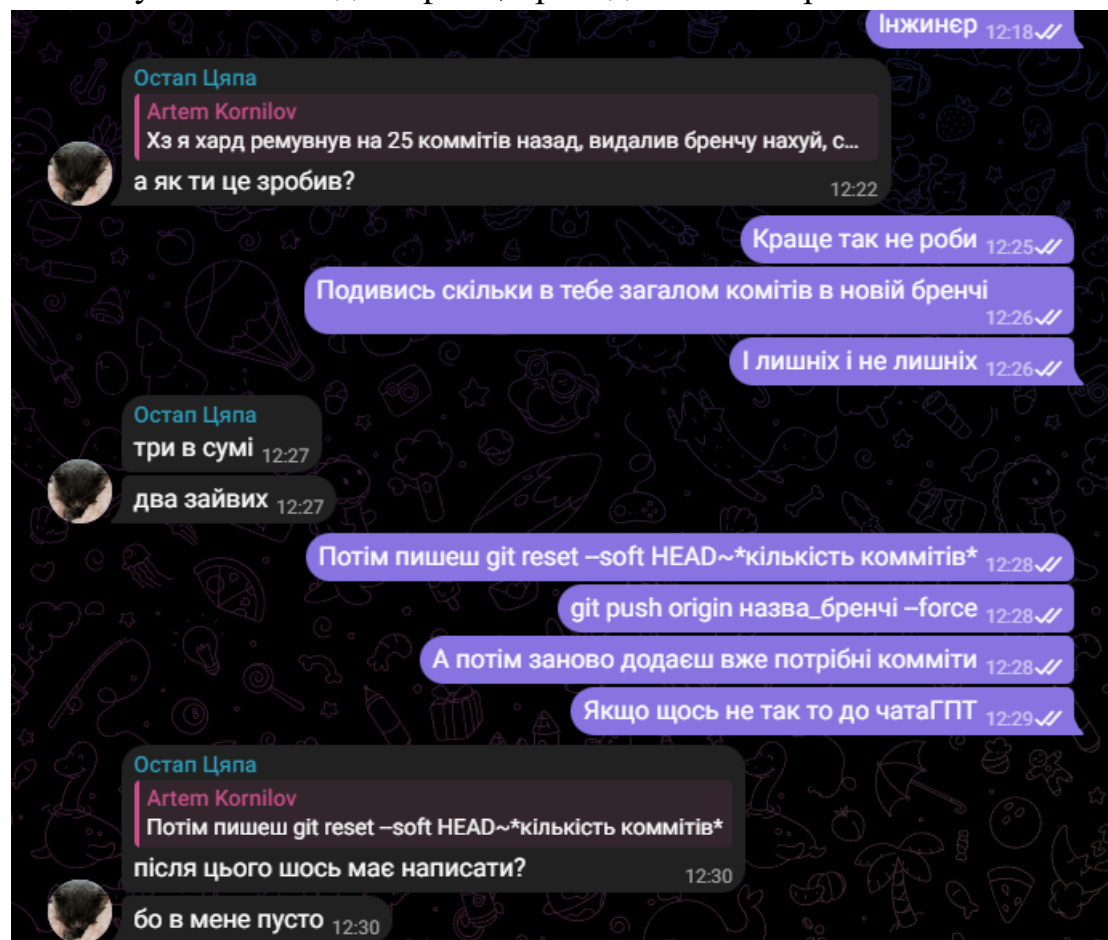
```
1 2 3 4 5
List reversed successfully.
5 4 3 2 1
```

```
List 1: 1 2 3 4 5
List 2: 1 2 3 4 5
List 3: 1 2 3
Compare List 1 and List 2: Equal
Compare List 1 and List 3: Not Equal
First number: 9 7 3
Second number: 6 5 4
Sum: 5 3 8
```

```
Original tree (Inorder): 1 2 3 4 5 6
Does the Tree contain '5' ? Yes
Mirrored tree (Inorder): 6 5 4 3 2 1
Tree after applying tree_sum (Inorder): 1 4 3 10 6 6
```

## Робота з командою:

Налаштували Trello для Еріс 6, проводили обговорення



## Висновки:

Отже, в межах цього епіка я старався зрозуміти, що таке списки, дерева та як їх реалізовувати в коді. Практикувався з записом даних у файли, а також покращив роботу з масивами та алгоритмами.

[Посилання на pull request](#)