

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 ШІ-13

Михайло Білоусов

Львів -- 2024

**Тема:**

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

## **Мета:**

навчитися ефективно будувати програмну логіку та працювати з Динамічними структурами, створювати чіткі й структуровані програми, що вирішують різноманітні задачі з використанням стандартних інструментів програмування.

## **Теоретичні відомості:**

- 1) Вивчив/знав:
  - a) Struct
  - b) class
- 2) Джерела: власний досвід

## **Виконання роботи:**

- 1) Опрацювання завдання та вимог до програми та середовища

### **Завдання №1 VNS Lab 10 – Task 1**

Написати програму, у якій створюються динамічні структури й виконати їхню обробку у відповідності зі своїм варіантом.

Для кожного варіанту розробити такі функції:

1. Створення списку.
2. Додавання елемента в список (у відповідності зі своїм варіантом).
3. Знищення елемента зі списку (у відповідності зі своїм варіантом).
4. Друк списку.
5. Запис списку у файл.
6. Знищення списку.
7. Відновлення списку з файлу.

Записи в лінійному списку містять ключове поле типу `*char` (рядок символів). Сформувані двонаправлений список. Знищити елемент із заданим номером. Додати по *K* елементів перед і після елемента із заданим ключем.

### **Завдання №2 Algotester Lab 5 – Variant 3**

Lab 5v3

Limits: 1 sec., 256 MiB

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.

Input

У першому рядку 2 числа N  
та M

- розміри карти

у другому рядку 2 числа x  
та y

- координати піку гори

Output

N  
рядків по M елементів в рядку через пробіл - висоти карти.

### **Завдання №3 Algotester Lab 7-8 – Variant 1**

Ваше завдання - власноруч реалізувати структуру даних "Двов'язний список".  
Ви отримаєте Q

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

Вам будуть поступати запити такого типу:

Вставка:

Ідентифікатор - insert

Ви отримуєте ціле число index елемента, на місце якого робити вставку.

Після цього в наступному рядку рядку написане число N - розмір списку, який треба вставити.

У третьому рядку N цілих чисел - список, який треба вставити на позицію index

Видалення:

Ідентифікатор - erase

Ви отримуєте 2 цілих числа - index, індекс елемента, з якого почати видалення та n

- кількість елементів, яку треба видалити.

Визначення розміру:

Ідентифікатор - size

Ви не отримуєте аргументів.

Ви виводите кількість елементів у списку.

Отримання значення i

-го елемента

Ідентифікатор - get

Ви отримуєте ціле число - index

, індекс елемента.

Ви виводите значення елемента за індексом.

Модифікація значення i

-го елемента

Ідентифікатор - set

Ви отримуєте 2 цілих числа - індекс елемента, який треба змінити, та його нове значення.

Вивід списку на екран

Ідентифікатор - print

Ви не отримуєте аргументів.

Ви виводите усі елементи списку через пробіл.

Реалізувати використовуючи перегрузку оператора <<

Input

Ціле число Q

- кількість запитів.

У наступних рядках Q

запитів у зазначеному в умові форматі.

Output

Відповіді на запити у зазначеному в умові форматі.

### **Завдання №4 Algotester Lab 7-8 – Variant 2**

Ваше завдання - власноруч реалізувати структуру даних "Динамічний масив".

Ви отримаєте Q

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

Вам будуть поступати запити такого типу:

Вставка:

Ідентифікатор - insert

Ви отримуєте ціле число index елемента, на місце якого робити вставку.

Після цього в наступному рядку рядку написано число N - розмір масиву, який треба вставити.

У третьому рядку N цілих чисел - масив, який треба вставити на позицію index

.

Видалення:

Ідентифікатор - erase

Ви отримуєте 2 цілих числа - index, індекс елемента, з якого почати видалення та n

- кількість елементів, яку треба видалити.

Визначення розміру:

Ідентифікатор - size

Ви не отримуєте аргументів.

Ви виводите кількість елементів у динамічному масиві.

Визначення кількості зарезервованої пам'яті:

Ідентифікатор - capacity

Ви не отримуєте аргументів.

Ви виводите кількість зарезервованої пам'яті у динамічному масиві.

Ваша реалізація динамічного масиву має мати фактор росту (Growth factor) рівний 2.

Отримання значення і

-го елементу

Ідентифікатор - get

Ви отримуєте ціле число - index, індекс елемента.

Ви виводите значення елемента за індексом. Реалізувати використовуючи перегрузку оператора []

Модифікація значення і

-го елементу

Ідентифікатор - set

Ви отримуєте 2 цілих числа - індекс елемента, який треба змінити, та його нове значення. Реалізувати використовуючи перегрузку оператора []

Вивід динамічного масиву на екран

Ідентифікатор - print

Ви не отримуєте аргументів.

Ви виводите усі елементи динамічного масиву через пробіл.

Реалізувати використовуючи перегрузку оператора <<

Input

Ціле число Q

- кількість запитів.

У наступних рядках Q

запитів у зазначеному в умові форматі.

Output

Відповіді на запити у зазначеному в умові форматі.

## **Завдання №5 Class Practice Work**

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 проходиться 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);

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 вітку 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

**До завдання 5 (Class Practice Work) є файл з блоксхемою  
practice\_work\_team\_tasks\_mykhailo\_bilousov.drawio**

3) Код програми з посиланням на зовнішні ресурси

До завдання №1 код

**vns\_lab\_10\_task\_1\_variant\_24\_mykhailo\_bilousov.cpp**

До завдання №2 код

**algotester\_lab\_5\_variant\_3\_mykhailo\_bilousov.cpp**

До завдання №3 код

**algotester\_lab\_78\_no\_template\_variant\_1\_mykhailo\_bilousov.cpp**

До завдання №4 код

**algotester\_lab\_78\_template\_variant\_2\_mykhailo\_bilousov.cpp**

До завдання №5 код  
**practice\_work\_team\_tasks\_mykhailo\_bilousov.cpp**

5) Результати виконаних завдань, тестування і затрачений час  
**Завдання №1**

abcde  
abcde1  
abcde2  
abcde3  
abcde4

abcde1  
abcde2  
abcde3  
abcde4

abcde1  
abcde2  
aaa  
aaa  
aaa  
abcde3  
aaa  
aaa  
aaa  
abcde4

Element with index 18446744073709551615 not found!

abcde1  
abcde2  
aaa  
aaa  
aaa  
abcde3  
aaa  
aaa  
aaa  
abcde4

List is empty!



abcde1  
abcde2  
aaa  
aaa  
aaa  
abcde3  
aaa  
aaa  
aaa  
abcde4

## Завдання №2

```
5 5
3 3
0 1 2 1 0
1 2 3 2 1
2 3 4 3 2
1 2 3 2 1
0 1 2 1 0
```

~/ai\_programming\_playground\_2024 / epic\_6\_pract...ilo\_bilousov ?3

## Завдання №3

```
9
insert
0
5
1 2 3 4 5

insert
2
3
7 7 7

print

e1 2 7 7 7 3 4 5
rase
1 2

print

size

get
3

set
3 13

print
1 7 7 3 4 5
6
3
1 7 7 13 4 5
```

```
~/ai_programming_playground_2024 / epic_6_pract...ilo_bilousov ?3
```

## Завдання №4

```
12
size
capacity

insert 0 2
100 100

size
capaci0
1
2
ty

insert 0 2
102 102

size
capacity

insert 0 2
4
4
8
103 103

size
capacity

print
6
8
103 103 102 102 100 100
```

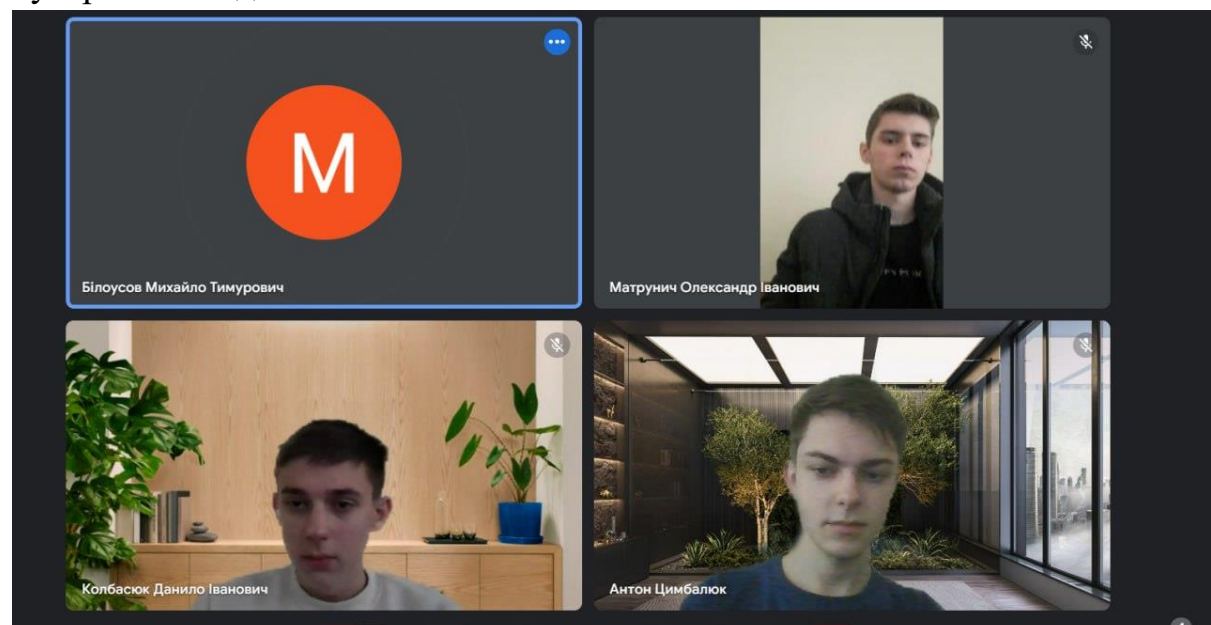
```
[ ~/ai_programming_playground_2024 / epic_6_pract...ilo_bilousov ?4
```

## Завдання №5

```
1 2 3
3 2 1
1 2 3
3 2 1
1 0 0 1
123
927
1050
[ 0, [ 1, [ 5 ], [ 6 ] ], [ 2, [ 3 ], [ 4 ] ] ]
[ 0, [ 2, [ 4 ], [ 3 ] ], [ 1, [ 6 ], [ 5 ] ] ]
[ 18, [ 7, [ 4 ], [ 3 ] ], [ 11, [ 6 ], [ 5 ] ] ]

~/ai_programming_playground_2024 / epic_6_pract..ilo_bilousov ?5
```

## Зустріч команди



## Висновки:

У цьому епіку я глибше ознайомився з C++, ознайомився з роботою з структурами та класами. Також я їх застосував для написання задач в Алготестері та лабораторних робіт у ВНС.

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/new/epic\\_6\\_practice\\_and\\_1\\_abs\\_mykhailo\\_bilousov](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/new/epic_6_practice_and_1_abs_mykhailo_bilousov)