Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

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

**На тему:** “Динамічні структури (Черга, Стек, Списки, Дерево). Алгоритми обробки динамічних структур.”

***з дисципліни:*** «Основи програмування»

до:

ВНС Лабораторної Роботи №10

Алготестер Лабораторної Роботи №5

Алготестер Лабораторної Роботи №7-8

Практичних Робіт до блоку №6

**Виконала:**

Студентка групи ШІ-13

Ходацька Аліна Віталіївна

Львів 2024

**Тема роботи:**

Динамічні структури. Алгоритми обробки динамічних структур.

**Мета роботи:**

Навчитись працювати з динамічними структурами даних, такими як списки, черги, стеки та дерева, а також удосконалити навички застосування алгоритмів їх ефективної обробки для розв’язання практичних задач.

**Теоретичні відомості:**

* динамічні структури
* стек
* черга
* зв’язні списки
* дерева
* алгоритми обробки динамічних структур

**Використані джерела:**

* <https://acode.com.ua/urok-90-dynamichni-masyvy/>
* <https://acode.com.ua/urok-111-stek-i-kupa/>
* https://www.youtube.com/watch?v=0BE0yxidMdE

**Виконання роботи**

**Завдання №1** VNS Lab 10 Task 1 Variant 12

*Постановка завдання*

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.

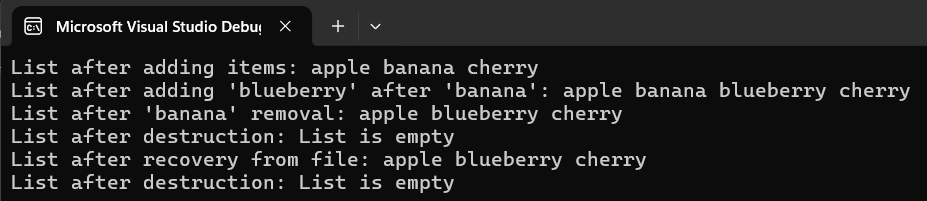
*Завдання*

Записи в лінійному списку містять ключове поле типу \*char (рядок символів). Сформувати двонаправлений список. Знищити з нього елементи, з однаковими ключовими полями. Додати елемент після елемента із заданим ключовим полем.

Зображення, що містить текст, знімок екрана

Автоматично згенерований описЗображення, що містить текст, знімок екрана, комп’ютер, монітор

Автоматично згенерований опис



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

У вас є карта гори розміром N×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.

Клітинці 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.

Зображення, що містить текст, схема, знімок екрана, ряд

Автоматично згенерований описЗображення, що містить схема, План, Креслення, ряд

Автоматично згенерований опис





**Завдання №3** Algotester Lab 7-8 Variant 2

*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 написане 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](https://en.wikipedia.org/wiki/Dynamic_array#Growth_factor)) рівний 2.
* Отримання значення i-го елементу  
  Ідентифікатор - get  
  Ви отримуєте ціле число - index, індекс елемента.  
  Ви виводите значення елемента за індексом. Реалізувати використовуючи перевантаження оператора []
* Модифікація значення i-го елементу  
  Ідентифікатор - set  
  Ви отримуєте 2 цілих числа - індекс елемента, який треба змінити, та його нове значення. Реалізувати використовуючи перевантаження оператора []
* Вивід динамічного масиву на екран  
  Ідентифікатор - print  
  Ви не отримуєте аргументів.  
  Ви виводите усі елементи динамічного масиву через пробіл.  
  Реалізувати використовуючи перевантаження 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.

Зображення, що містить текст, знімок екрана, монітор

Автоматично згенерований опис

Зображення, що містить знімок екрана, текст, монітор

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, програмне забезпечення

Автоматично згенерований опис

**Завдання №5** 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 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 списка (напр. 379 ⟹

9→7→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



Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

**Завдання №6** Algotester Self Practice work Lab 5 Variant 2

В пустелі існує незвичайна печера, яка є двовимірною. Висота - N, ширина - M.

Всередині печери є пустота, пісок та каміння. Пустота позначається буквою О , пісок S і каміння X;

Одного дня стався землетрус і весь пісок посипався вниз. Він падає на найнижчу клітинку з пустотою, але він не може пролетіти через каміння.

Ваше завдання сказати як буде виглядати печера після землетрусу.

*Input*

У першому рядку 2 цілих числа N та M - висота та ширина печери

У N наступних рядках стрічка  яка складається з N цифр - i-й рядок матриці, яка відображає стан печери до землетрусу.

*Output*

N рядків, які складаються з стрічки розміром M - стан печери після землетрусу.

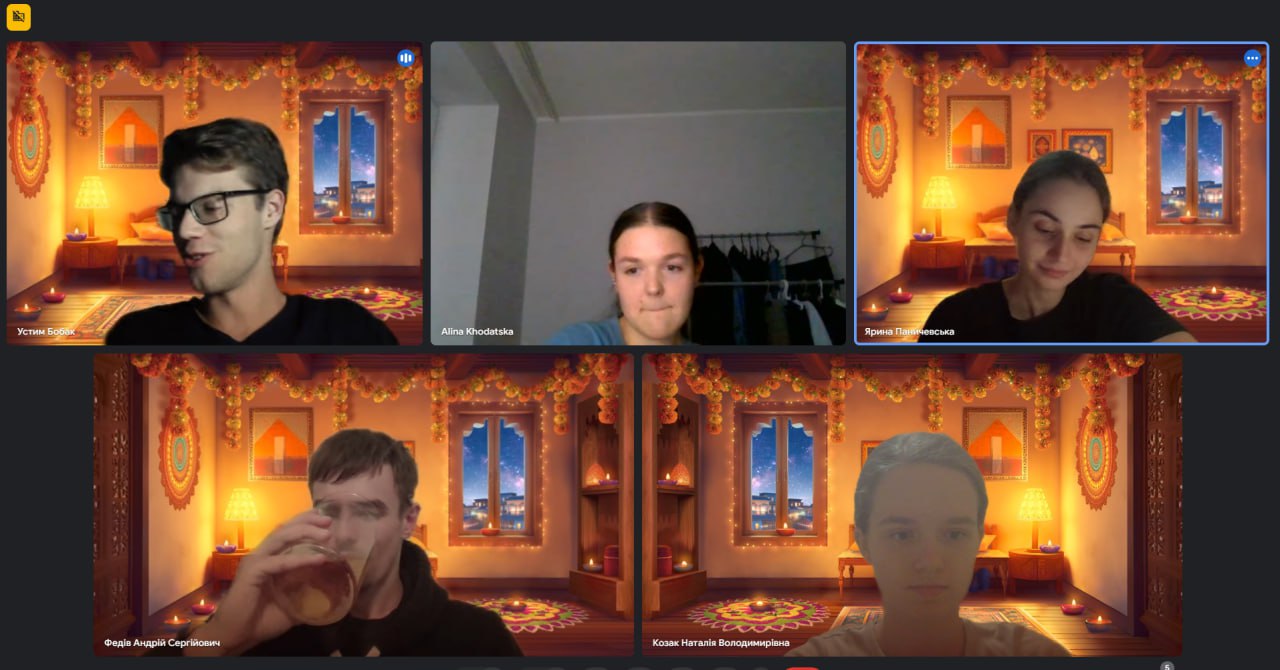
Зображення, що містить текст, електроніка, монітор, знімок екрана

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

**Зустріч з командою та дошка в Trello**

****

**Зображення, що містить текст, знімок екрана, число, Операційна система

Автоматично згенерований опис**

**Pull request:** https://github.com/artificial-intelligence-department/ai\_programming\_playground\_2024/pull/662