

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

Виконав:

Студент групи ШІ-13

Бойко Роман Андрійович

Львів 2024

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

Мета роботи: Навчитися користуватися різними видами динамічних структур.

Теоретичні відомості:

- Лекції, практичні.
- W3schools
- ChatGPT
- YouTube

Виконання роботи

Завдання № 1 : Practice Work – 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 списка (напр. 379 \Rightarrow 9 \rightarrow 7 \rightarrow 3);
- функція повертає новий список, передані в функцію списки не модифікуються.

Бінарні дерева

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

Результат виконання програми:

```
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 ] ] ]
```

Час виконання завдання ~ 2.5 години

Завдання № 2 : VNS Lab 10 - Task 1-14

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.

Результат виконання програми:

```
Список після створення:
Елемент 1 Елемент 2 Елемент 3
Введіть номер елемента, який хочете видалити: 2
Елемент з номером 2 видалено.
Список після видалення елемента:
Елемент 1 Елемент 3
Список після додавання нового елемента:
Елемент 1 Елемент 3 Елемент 4
Список збережено у файл list.txt.
Список знищено.
Список після знищення:
Список порожній.
Список знищено.
Список відновлено з файлу list.txt.
Список після відновлення з файлу:
Елемент 1 Елемент 3 Елемент 4
Список знищено.
```

Час виконання завдання ~ 1.5 години

Завдання № 3 : 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.

Input

У першому рядку 2 числа NN та MM - розміри карти

у другому рядку 2 числа xx та yy - координати піку гори

Output

NN рядків по MM елементів в рядку через пробіл - висоти карти.

Результат виконання програми:

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

Час виконання завдання ~ 50хв

Завдання № 4 : Algotester Lab 7-8v1

Ваше завдання - власноруч реалізувати структуру даних "Двотвірний 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 <<<<

Input

Ціле число QQ - кількість запитів.

У наступних рядках QQ запитів у зазначеному в умові форматі.

Output

Відповіді на запити у зазначеному в умові форматі.

Результат виконання програми:

```

9
insert
0
5
1 2 3 4 5
insert
2
3
7 7 7

print
1 2 7 7 7 3 4 5

erase
1 2

print
1 7 7 3 4 5

size
6
get
3
3
set
3 13
print
1 7 7 13 4 5

```

Час виконання завдання ~ 1.5 години

Завдання № 5 : Self Practice Work

Lab 5v2

В пустелі існує незвичайна печера, яка є двохвимірною. Її висота це NN, ширина - MM.

Всередині печери є пустота, пісок та каміння. Пустота позначається буквою OO , пісок SS і каміння XX;

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 NN та MM - висота та ширина печери

У NN наступних рядках стрічка rowіrowі яка складається з NN цифер - і-й рядок матриці, яка відображає стан печери до землетрусу.

Output

NN рядків, які складаються з стрічки розміром MM - стан печери після землетрусу.

Результат виконання програми:

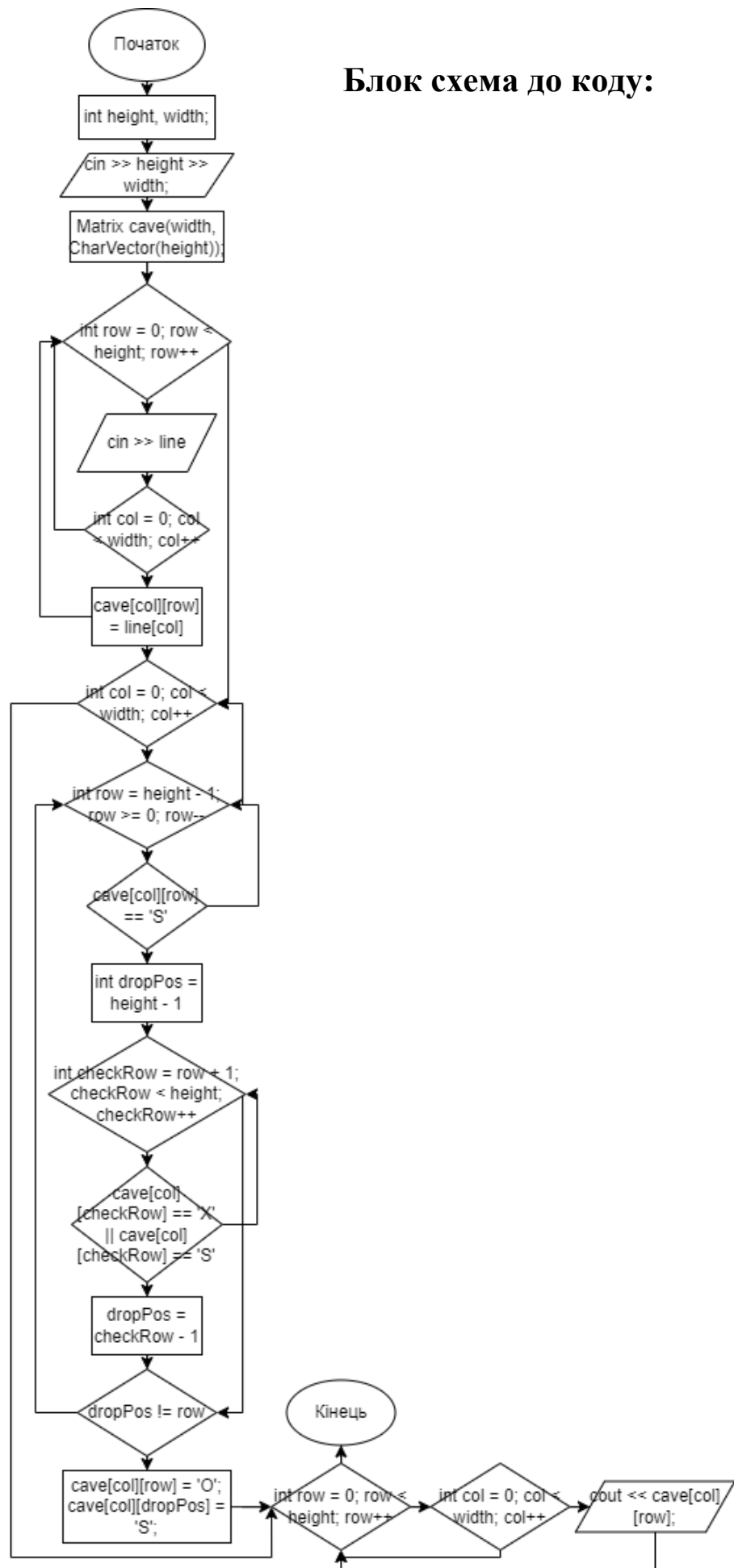
```

5 5
SSSSS
00000
S00XX
0000S
00S00

00000
000SS
000XX
S0000
SSS0S

```

Блок схема до коду:



Час виконання завдання ~ 2 години

Висновки:

Я навчився застосовувати динамічні структури для ефективного зберігання та обробки даних в програмах. Також отримав розуміння алгоритмів для роботи з чергою, стеком, списками та деревами, що дозволяє вирішувати складні обчислювальні задачі.