19AIE111 - DATA STRUCTURES-1 – 2020 batch

Program List

1. Write a java program to read the text from “in.dat” file and copy the contents into another file by replacing all ‘o’s or ‘0’s to *

2. Write a java program to read 10 numbers from “in.dat” file and sort them and store into another file.

3. Write a program to read the numbers from a file into two 2-dim arrays(matrices), find the sum, rowsum, colsum, diagonalsum and print it with appropriate c aptions.

4. Display the upper triangular matrix and lower triangular matrix.

5. Display the frequency(count) of each number in the given matrix.

1 3 2
2 4 1
2 3 5

Count(1)=2 Count(2)=3 Count(3)=2 Count(4)=1 Count(5)=1

6. Challenging question: (OPTIONAL)

You have a matrix of size N * N with rows numbered through 1 to N from top to bottom and columns through 1 to N from left to right. It contains all values from 1 to N^2, i.e. each value from 1 to N^2 occurs exactly once in the matrix. Now, you start from the cell containing value 1, and from there visit the cell with value 2, and then from there visit the cell with value 3, and so on till you have visited cell containing the number N^2. In a single step, you can move from a cell to one of its adjacent cells. Two cells are said to be adjacent to each other if

they share an edge between them. Find out minimum number of steps required. (atleast number of steps)

For example, if matrix is

```
1 3

2 4
```

You start from cell containing value 1 (i.e. (1,1)) and you want to visit cell with value 2 (i.e. (2,1)). Now, from cell (2,1)

you have to visit cell (1,2), which can be done is 2 steps (First we go from (2, 1) to (1, 1) and then to (1, 2), total 2 steps).

Finally you move to cell where value 4 is present in 1 step. So, total number of steps required is 4.

Input

The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows.

The first line of each test case contains a single integer N denoting the size of matrix. Each of the next N lines contain N integers denoting the values in the rows of the matrix.

Output

- For each test case, output in a single line the required answer.

Constraints

- **1 ≤ T ≤ 5**
- **1 ≤ N ≤ 500**

Subtasks

- **Subtask 1** (30 points) : **1 ≤ N ≤ 20**
- **Subtask 2** (70 points) : Original constraints

Example

```
Input:

2

2

1 3

2 4

3

1 7 9

2 4 8

3 6 5

Output:

4

12
```

7. Singly Linked list (SLL), Circular Linked Lists(CLL) and Doubly linked List(DLL) Operations

- insertfront, insertlast, (insertafter, insertbefore – optional)
- deletelast, deletefirst, deleteelement, delete_pos_element(particular position element)
- display()
- search
- replace
- Remove a particular element
- Replace an element
- Find the min, max element
- Remove duplicates
- Merging two sorted lists
- Split a list into two halves
- Insert a list into another list ….

8. Implementation of stack using arrays, SLL, DLL

9. Applications of stack (5 programs)

Infix to postfix

Evaluation of postfix

Delimiter checking(parenthesis checking)

Recursion- factorial,

Tower of Hanoi

10. Implementation of QUEUE and its types

Queue USING ARRAYS AND LINKED LIST(ANY ONE SLL OR CLL OR DLL),

Circular queue USING ARRAYS AND LINKED LIST(ANY ONE SLL OR CLL OR DLL),

Priority queue USING ARRAYS AND LINKED LIST(ANY ONE SLL OR CLL OR DLL),

Deque USING ARRAYS AND LINKED LIST(ANY ONE SLL OR CLL OR DLL),

11. Binary search tree – operations and traversals