

## Lab-9

Name: Abu Fardous Shauon

ID: 22301320

Section: 24

Task 1a): Firstly I made a multidimensional array of length  $n \times n$ . where all the values are 00. Then I updated the value of the index with respect to the input.

Task 1b): First I made an array of length  $n+1$  containing 0 values. Then I append the values in the respective index from the input file. Then I showed the output.

Task 2): Firstly I made a dictionary, where I put the values from as key and value from the input. Then I created three empty array. And I called BFS. It will check and append the first value in visited and queue. And gradually it will check every value until the queue is empty. Then finally we get traverse array and showed the output.

Task 3): I made a dictionary and put the values as key, value pairs. I called DFS function. It will check every values whether



it is present in visited array or not.  
and append in traversers..

Task 4: I created the <sup>dictionary</sup> ~~created~~ with key pair values. And called is cyclic function. Here visit the values of visited and helper is False. And It will DFS function. And gradually the value of visited will update. ~~The~~ If the index value ~~of~~ is already true then it will return cyclic.

Task 5: Here I have implement BFS. After DFS traversal we will find the shortest path by tracking the parent vertices. The length of path - 1 is the time and the list of ~~path~~ shortest path backwards is the path.

Task 6: Firstly I created 2D Matrix. We implement DFS to traverse the grid and count the number of connected regions marked by 'D'. The traverse function iterates through the grid from each "0" to find the size of connected parts and returns the maximum 'D' found throughout the traversal.