Data Structures and Algorithms
CS-F22
LAB-06

Issue Date: August 12,2024

Start Time: 2:15 PM
Completion Time: 04:15 PM

Total Marks : 60

Objective:

In this lab, students will understand about singly and doubly linked list working. The objective is to reinforce understanding of different functionalities performed by linked list, stack, queue and linear mapping of multi-directional arrays.

Instructions:

1) Follow the question instructions very carefully, no changes in function prototypes are allowed.
2) Your laptops must be on airplane mode.
3) Anyone caught in an act of plagiarism would be awarded an "F" grade in this Lab.

TASK 01: Linear Mapping [10 Marks]

Develop a row-major order formula for 2D array where underlaying 1D array index start from 0 but for a programmer the 1st dimension starts from 's' and 2nd dimension start from 'k'.
For Example: int a[3] [4]; // Assume s = - 1 k=5
The first element in matrix will be [-1][5]
For Programmer: array (row, col)
1st row will be: (-1,5), (-1,6), (-1,7), (-1, 8)
2nd row will be: (0,5), (0,6), (0,7), (0,8)
3rd row will be: (1,5), (1,6), (1,7), (1,8)
Underlaying Array: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.
So, (-1,5) maps to 0, (1,6) maps to 9

Function Prototype:

```
int 1Dmapping(int rows, int columns, int startRow, int startColumn )  //returns the index
```

TASK 02: Double a Number Represented as Linked List [15 Marks]

You are given the head of a non-empty linked list representing a non-negative integer without leading zeroes. Return the head of the linked list after doubling it.

Sample Run:

Input: head = [9,9,9] //999*2 =1998
Output: [1,9,9,8]

Function Prototype:

```
Node* doubleIt(Node* head)
```

TASK 03: Largest Number [15 Marks]

Given an integer N with D digits without any leading zeroes. Find the largest number which can be obtained by deleting exactly K digits from the number N.

Input format:

First line contains integer N.
Second line contains integer K.

Output format:

Return the largest number which can be obtained by deleting exactly K digits from the number N.

Function Prototype:

```
string getLargestNumber(string N, int K);
```

Sample Run:
Input:
1432219
3
Output:
4329
Note: Return the largest number without any leading zeros.

TASK 04: Queries On A Permutation With Key [20 Marks]

Given the array queries of positive integers between 1 and m, you have to process all queries[i] (from i=0 to i=queries.length-1) according to the following rules:

In the beginning, you have the permutation P=[1,2,3,...,m].

For the current i, find the position of queries[i] in the permutation P (indexing from 0) and then move this at the beginning of the permutation P. Notice that the position of queries[i] in P is the result for queries[i].

Return an array containing the result for the given queries.

Function Prototype:

int * processQueries(int * queries,int size, int m)

Sample Run:
Input:
queries = [3,1,2,1], m = 5
Output:
[2,1,2,1] // The queries are processed as follow:
For i=0: queries[i]=3, P=[1,2,3,4,5], position of 3 in P is 2, then we move 3 to the beginning of P resulting in P=[3,1,2,4,5].
For i=1: queries[i]=1, P=[3,1,2,4,5], position of 1 in P is 1, then we move 1 to the beginning of P resulting in P=[1,3,2,4,5].
For i=2: queries[i]=2, P=[1,3,2,4,5], position of 2 in P is 2, then we move 2 to the beginning of P resulting in P=[2,1,3,4,5].
For i=3: queries[i]=1, P=[2,1,3,4,5], position of 1 in P is 1, then we move 1 to the beginning of P resulting in P=[1,2,3,4,5].
Therefore, the array containing the result is [2,1,2,1].