National University of Computer and Emerging Sciences, Lahore Campus

Course:	Applied Programming (AP)	Course Code:	CS-0319
Program:	MSCS	Semester:	Spring 2024
Deadline:	18th April 2024	Total Marks:	40
Section:	MSCS-2A		

Important Instructions:

1. *Submit one .cpp file for each question (three in total). Format: <rollno_A1_Qno.cpp>*
2. *You are not allowed to copy solutions from other students. If any sort of cheating / plagiarism is found, negative marks will be given to all students involved.*
3. *Late submission of your solution will result in a penalty.*
4. *Do not use the builtin <stack> and <queue> data type. Create your own data structure classes.*

Question 1: ('Invertible' Stack) [10 marks]

Add the following methodology to the Stack Class (You must create your own Stack Class).
Add a method called **flipStack**. This method should work in **O(1)**. Its effect should be such that the whole stack should be logically inverted, i.e., the oldest element becomes the newest and vice versa. So the next pop will remove the element that was at the bottom of the stack before it had been flipped. Notice that the stack may be flipped again and again by using the flipStack method repeatedly. Make sure that no slot of the array is wasted.

Question 2: ('Undo' Stack) [10 marks]

Create a class called **UndoStack**. This stack is intended to be used for the undo operation in another application. It should have the property that it remembers at most the last 100 elements (max size) pushed into it. So, for example, if there were 101 push operations it would "forget" the oldest of these and if a pop happens the 2nd pushed element will be popped, etc. Write complete C++ code for this class.

Question 3: (Radix Sort) [10 + 10 marks]

Radix sort is a sorting algorithm, which sorts the keys based on the values of digits in keys. It takes a queue containing n keys to be sorted, where each key consists of k number of digits, and there could be m possible values for each digit 0 through m-1. Radix sort uses an array consisting of (m) queues for sorting of these keys. For example, if each key contains k = 3 digits and each individual digit has m=10 possible values 0 to 9, then it will use an array consisting of 10 queues 0-9 in the sorting process of considering all digits one by one as follows.
(use radix sort option given here to visualize: https://visualgo.net/en/sorting?slide=6-4)

You have to **implement two Radix sort algorithms**, one for integer and one for string keys with any value of n and k.

All the Best