Data Structures and Algorithms Lab

SE-F22

LAB-07

Issue Date: March 14, 2024

Start Time: 8:15AM

Total Marks: 50

The objective of this lab is to:

Play with queues.

Instructions:

1) Follow the question instructions very carefully, no changes in function prototypes are allowed.
2) Make separate header files for ADTs. (.h_for function prototypes) (.cpp for function implementations)

Task 01(Stack Using Queues) [25 Marks]

In a toy manufacturing unit, the workers need to organize the toys into stacks for packaging. However, they only have access to queues for arranging the toys. Your task is to help them implement a stack data structure using queues.

`push(x)`: Pushes element x onto the stack.
`pop()`: Removes the element on top of the stack and returns it.
`top()`: Returns the element on the top of the stack without removing it.
`empty()`: Returns true if the stack is empty, false otherwise.

You should implement the stack using standard queue operations such as **enqueue** and **dequeue**.

Template: (Follow it, else marks will be deducted!)

```
class Stack {
private:
public:
    Stack();  // Constructor

    void push(int x);  // Push element x onto stack
    int pop();  // Removes the element on top of the stack and returns that element
    int top();  // Get the top element
    bool empty();  // Returns whether the stack is empty
};
To get full marks use only 1 queue to simulate the stack, any other working approach involving
queue will get a maximum of 18 marks. Approach not involving queue will get 0
```

Task 02 (Toy Collection Queue) [25 Marks]

At a toy store, there are two types of toys available: **toy cars and toy dolls**, represented by numbers **0 and 1** respectively.

Children wait in line to collect their preferred toy. The toys are arranged in a stack, and each child at the front of the line decides whether to take the toy at the top of the stack based on their preference. **If the toy type matches their preference, they take it and leave the queue; otherwise, they move to the end of the line.**

This process continues until no child in the queue wants to take the toy at the top of the stack.

Your task is to determine how many children are **unable to collect their preferred toy**.

Input:

An integer array children representing the preferences of each child in the queue. Each element is either 0 or 1, indicating whether the child prefers a toy car or a toy doll respectively.

An integer array toys representing the types of toys stacked up. Each element is either 0 or 1, indicating whether the toy at that position is a toy car or a toy doll respectively.

Examples:

```
children = [1, 1, 1, 0, 0, 1]
toys = [1, 0, 0, 0, 1, 1]

Output: 3
```