Cairo University
Faculty of Engineering

Data Structures Course - CMPN 102
Spring 2022

Assignment

Rules and regulations

- **This assignment consists of six problems related to the data structures discussed earlier in the course, which are (Linked Lists, Stacks, Queues).**
- **Assignment submission is due Friday 29th of April at 11:55PM, any submission after that wouldn't be considered.**
- **Plagiarism won't be tolerated. any sign of plagiarism would result in a zero grade for both students.**
- **You can use the code of (Linked List, Stack, Queue) classes that were discussed in the labs.**
- **For linked list problems, implement the required functions as member functions of the linked list class.**
- **For Stack and Queue problems, use Stack and Queue as a black box, don't modify anything in it.**
- **If you have any question, send an email to one of the following emails:**
  - **mohamed1989877@gmail.com**
  - **alyhassan62@yahoo.com**

Problem 1:

Problem definition:

Given the head of a linked list consists of **zeros and ones only**, which makes it looks like a binary number.

The head pointer points to **the most significant bit (the leftmost bit)**.

Convert the binary number represented by this linked list to its decimal representation.

Constraints:

$1 \ <= \ ListSize \ <= \ 60$

$ListElements \ \in \ \{0, \ 1\}$

Example:

Input: 0 0 1 1 1

Output: 7

Problem 2:

Problem definition:

Given the head of a linked list, rotate the list to the right by k places.

Note: k could be larger than the size of the list, which means you would rewind the list again.

Constraints:

$1 \ <= \ ListSize \ <= \ 10^5$

$1 \ <= \ k \ <= \ 10^9$

Example:

Input: $List$: $[1 \ -> \ 2 \ -> \ 3 \ -> \ 4 \ -> \ 5 \ -> \ NULL]$, $k \ = \ 3$

Output: $List$: $[3 \ -> \ 4 \ -> \ 5 \ -> \ 1 \ -> \ 2 \ -> \ NULL]$

Problem 3:

Problem definition:

Given the head of a linked list, split it into two new linked lists, the first one contains the first half of elements, the second one contains the rest of the linked list.

Note: if the number of nodes is **odd**, make the first list contain more elements.

Constraints:

$1 \ <= \ ListSize \ <= \ 10^6$

Examples:

Input: $[1 \ -> \ 2 \ -> \ 3 \ -> \ 4 \ -> \ 5 \ -> \ 6 \ -> \ NULL]$

Output: $[1 \ -> \ 2 \ -> \ 3 \ -> \ NULL]$, $[4 \ -> \ 5 \ -> \ 6 \ -> \ NULL]$

Input: $[1 \ -> \ 2 \ -> \ 3 \ -> \ 4 \ -> \ 5 \ -> \ 6 \ -> \ 7 -> \ NULL]$

Output: $[1 \ -> \ 2 \ -> \ 3 \ -> 4 \ -> \ NULL]$, $[4 \ -> \ 5 \ -> \ 6 \ -> \ NULL]$

Problem 4:

Problem definition:

Given a stack and a queue, exchange their content but keep the default retrieve order as it is.

For example, if the stack content is [1, 2, 3, 4, 5] , 1 is the top of the stack, then the first element of the queue must be 1, making the queue content is [1, 2, 3, 4, 5].

if the queue content is [1, 2, 3, 4, 5] , 1 is the start of the queue, then the top element of the stack must be 1, making the stack content is [1, 2, 3, 4, 5].

Constraints:

$1 \ <= \ StackSize \ <= \ 1000$

$1 \ <= \ QueueSize \ <= \ 1000$

Example:

Input: $Stack \ = \ [1, \ 2, \ 3, \ 4, \ 5], Queue \ = \ [6, \ 7, \ 8, \ 9, \ 10]$

Output: $Stack \ = \ [6, \ 7, \ 8, \ 9, \ 10], \ Queue \ = \ [1, \ 2, \ 3, \ 4, \ 5]$

Problem 5:

Problem definition:

Our college's cafeteria offers two types of sandwiches at lunch break, (beef and chicken) referred to by the numbers (0 and 1) respectively.

All students stand in a queue. Each student either prefers beef or chicken sandwiches.

The number of sandwiches in the cafeteria is equal to the number of students. The sandwiches are placed in a **stack**. At each step:

- If the student at the front of the queue prefers the sandwich on the top of the stack, they will **take it** and **leave the queue.**
- Otherwise, they will **leave it** and go to the **queue's end.**

This continues until none of the queue students want to take the top sandwich and are thus unable to eat.

You are given a stack representing the sandwiches in the cafeteria, and a queue representing the preferred sandwich of each student in the queue initially.

Calculate the number of students that are unable to have lunch.

Constraints:

$1 \ <= \ number\ of\ students \ <= \ 100$

$sandwich\ types\colon\ 0\ for\ beef,\ 1\ for\ chicken.$

$all\ students\ prefer\ either\ beef\ or\ chicken.$

Example:

Input: $students\colon\ [1, 1, 0, 0],\ sandwiches\colon\ [0, 1, 0, 1]$

Output: 0

Problem 6:

Problem definition:

Some of your friends are playing a basketball game with strange rules against a team from another university.You are responsible for recording the score of your friends' team. At the beginning of the game you have an **empty record**.

Given a list of strings representing the operations you would take to record the scores, where each operation is one of the following:

1. An integer x -> record a **new score** x.
2. $" + "$ -> record a **new score** that is the sum of the previous two scores. it is guaranteed there will always be two previous scores.
3. $"D"$ -> **Discard the previous score** and remove it from the record. it is guaranteed there will always be a previous score.

Calculate the sum of all scores on the record.

Constraints:

$1 <= number\ of\ operations <= 1000$

$-1000 <= x <= 1000$

Example:

Input: $operations = ["15", "10", " + ", "6", "D", "5"]$

Output: 55

Explanation:

- After operation 1, record = [15]
- After operation 2, record = [15, 10]
- After operation 3, record = [15, 10, 25]
- After operation 4, record = [15, 10, 25, 6]
- After operation 5, record = [15, 10, 25]
- After operation 6, record = [15, 10, 25, 5], so output = 15 + 10 + 25 + 5 = 55.