


MIN

1+0 = 1/17

	National University of Computer and Emerging Sciences (Lahore)		
	Course:	Applied Programming	
	Section:	MCS-1A	Semester: Spring 2024
	Duration:	40 minutes	Total Marks: 20/17
	Name:	Ali Amrighan	Roll no: 23L-8056

Question 1: Write a C++ function `printFibonacci(int n)` which prints the Fibonacci numbers till the  $n$ th term inclusively by using FIFO Queue data structure. You may assume that Queue data structure is already implemented.

Example:

Input:  $n = 7$

Output: 0 1 1 2 3 5 8

```
void printFibonacci(int n) {
```

```
    Count = 0;
```

```
    push(0);
```

```
    print(0);
```

```
    Count++;
```

```
    push(1);
```

```
    print(1);
```

```
    Count++;
```

```
    while (Count <= n) {
```

```
        i = pull(); j = pull();
```

```
        K = i + j;
```

```
        push(j);
```

```
        push(K);
```

```
        print(K);
```

```
        Count++;
```

Implementation of print?

for loop

Invalid function calls w/o object initialization/callable.

- No printing done

- not given

Question 2: Write a C++ function `evalRPN(char tokens[], int n)`, where  $n$  is the size of the 1-D character array `tokens` of tokens that represents an arithmetic expression in Reverse Polish Notation (RPN) a.k.a. Postfix Notation. The function evaluates the given RPN expression and then returns an integer that represents the value, that is the answer, of that expression. You may use any number of helper functions and may assume the implementation of the LIFO Stack data structure.

Note:

- The valid operators are '+', '-', '\*' and '/'.
- Each operand may be a single-digit integer or another expression.
- The division between two integers always truncates toward zero.
- There will not be any division by zero.
- The input represents a valid arithmetic expression in a reverse polish notation.

Example:

Input: `tokens[5] = ["2", "1", "+", "3", "*"]`,  $n = 5$

Output: 9

Explanation:  $((2 + 1) * 3) = 9$

not attempted. zero. (0)