

### Programming Exercise

Write a program that uses linked lists in order to support the following operations:

1. PUSH (S, x) - pushes a value x into a stack S
2. POP (S, i) - gets a number i (positive integer) and pops i numbers of S. If S contains less than i values, the operation is impossible to execute (program prints ERROR in this case – see below).
3. REVERSE (S) - reverse the order of the elements in S (you might want to apply recursion). If, for example, S is a stack and x was the last inserted, from now on x is treated as the first inserted element.
4. QUEUE (S) - declares that from this moment S becomes and acts like a queue. Nothing is printed after this operation.
5. ENQUEUE(S, x) - adds x to a queue
6. DEQUEUE(S) - removes element when S is a queue.
7. STACK(S) - makes S into a stack. Nothing is printed after this operation is executed.
8. AVERAGE(S) - returns the average of the numbers in S

The program reads a sequence of strings, names of the operations, and additional value (when required). The program prints the values in S after each operation is executed unless it's stated otherwise. If S is empty, program prints EMPTY. The program prints ERROR in case and the operation is impossible to execute. Assume, the program starts with empty stack. Assume, the values are positive integers. The program ends when the string END is entered. Assume, the operation AVERAGE returns 0.0 if S is empty.

For every operation the program must state its running time – write this in comments.

Each important statement in your program must be documented in comments. Programs without documentation will not receive full credit.

Example 1:

Input:

```
AVERAGE
PUSH 4
PUSH 5
PUSH 7
PUSH 9
POP 2
PUSH 11
PUSH 6
REVERSE
```

QUEUE  
ENQUEUE 10  
ENQUEUE 12  
DEQUEUE  
DEQUEUE  
AVERAGE  
DEQUEUE  
DEQUEUE  
DEQUEUE  
DEQUEUE  
DEQUEUE  
REVERSE  
END

Output:

0.0  
4  
5 4  
7 5 4  
9 7 5 4  
5 4  
11 5 4  
6 11 5 4  
4 5 11 6  
4 5 11 6 10  
4 5 11 6 10 12  
5 11 6 10 12  
11 6 10 12  
9.75  
6 10 12  
10 12  
12  
EMPTY  
ERROR  
EMPTY

Example 2:

POP 1  
PUSH 2  
PUSH 4

QUEUE  
ENQUEUE 10  
ENQUEUE 12  
PUSH 3  
STACK  
POP 2  
REVERSE  
AVERAGE  
DEQUEUE  
STACK  
POP 2  
END

Output:

ERROR  
2  
4 2  
4 2 10  
4 2 10 12  
ERROR  
10 12  
12 10  
11.0  
10  
ERROR

Example 3:

ENQUEUE 10  
DEQUEUE  
QUEUE  
ENQUEUE 1  
ENQUEUE 2  
ENQUEUE 3  
ENQUEUE 7  
DEQUEUE  
REVERSE  
STACK  
POP 2  
REVERSE

AVERAGE  
POP 2  
END

Output:

ERROR  
ERROR  
1  
1 2  
1 2 3  
1 2 3 7  
2 3 7  
7 3 2  
2  
2  
2.0  
ERROR

Run your program on THREE inputs listed below.

**TEST INPUT 1:**

PUSH 3  
PUSH 4  
PUSH 6  
AVERAGE  
POP 4  
POP 2  
QUEUE  
ENQUEUE 1  
ENQUEUE 2  
DEQUEUE  
ENQUEUE 8  
ENQUEUE 9  
STACK  
PUSH 12  
POP 1  
PUSH 6  
REVERSE  
AVERAGE  
END

### **TEST INPUT 2:**

POP 1  
QUEUE  
ENQUEUE 1  
ENQUEUE 2  
DEQUEUE  
ENQUEUE 8  
ENQUEUE 9  
AVERAGE  
STACK  
PUSH 3  
PUSH 4  
PUSH 6  
QUEUE  
ENQUEUE 1  
ENQUEUE 2  
DEQUEUE  
DEQUEUE  
AVERAGE  
STACK  
PUSH 1  
POP 2  
REVERSE  
END

### **TEST INPUT 3:**

AVERAGE  
REVERSE  
PUSH 3  
POP 1  
AVERAGE  
PUSH 4  
PUSH 5  
PUSH 6  
POP 4  
POP 2  
QUEUE  
ENQUEUE 1  
DEQUEUE

ENQUEUE 2  
DEQUEUE  
ENQUEUE 8  
STACK  
POP 1  
PUSH 4  
REVERSE  
AVERAGE  
END

Submit on Canvas the following files:

1. Source file (txt, pdf or Word format ONLY)
2. Output file (txt, pdf or Work format ONLY)