Stack and Queue Problems

1: Balanced Parentheses

Given a string containing only parentheses `(` and `)`, check if the string is balanced. A string is balanced if each opening parenthesis has a matching closing one in the correct order.

Input Format:
A single string `S` containing parentheses.

Output Format:
Print "YES" if balanced, otherwise "NO".

Sample Input:
(()())

Sample Output:
YES

Sample Input:

(()))()

Sample Output:
NO

2: Next Greater Element

Given an array of integers, for each element find the next greater element to its right. If none exists, output `-1`.

Input Format:
First line: integer `N` (number of elements).
Second line: `N` integers separated by spaces.

Output Format:
Print `N` integers, each representing the next greater element.

Sample Input:
4
4 5 2 25

Sample Output:
5 25 25 -1

Sample Input:

3
6 4 5

Sample Output:

-1 5 -1

3: Queue Using Two Stacks

You are given queries to perform enqueue and dequeue operations on a queue implemented using two stacks.

Input Format:

First line: integer `Q` (number of queries).
Next `Q` lines contain one of the following:
- `1 X`   Enqueue element `X`.
- `2`   Dequeue element from front.
- `3`   Print the front element.

Output Format:

For each query type `3`, print the front element.

Sample Input:

6
1 10
1 20
3
2
3
2

Sample Output:

10
20

4: Reverse a Queue Using Stack

You are given `N` integers in a queue. Reverse the queue using a stack and print the elements.

Input Format:

First line: integer `N`.
Second line: `N` integers (queue elements from front to rear).

Output Format:

Print the reversed queue elements.

Sample Input:

5
1 2 3 4 5

Sample Output:

5 4 3 2 1

5: Evaluate Postfix Expression

You are given a postfix expression consisting of integers and operators `+ - * /`. Evaluate the expression and print the result.

Input Format:

A single string containing the postfix expression with space-separated tokens.

Output Format:

Print the evaluated result as an integer.

Sample Input:

6 3 2 + * 2 /

Sample Output:

15