SSS – 13– 9 – 2020

(C – any 3, S – any 4 , E – all 5)

1. Given a stack of integers, sort it in ascending order using another temporary stack.

Examples:

Input : [34, 3, 31, 98, 92, 23]
Output : [3, 23, 31, 34, 92, 98]

Input : [3, 5, 1, 4, 2, 8]
Output : [1, 2, 3, 4, 5, 8]

2. Given a balanced parentheses string S, write code to compute the score of the string based on the following rule: 2.5

- () has score 1
- AB has score A + B, where A and B are balanced parentheses strings.
- (A) has score 2 * A, where A is a balanced parentheses string.

Examples:

Input: "()"

Output: 1

Input: "(())"

Output: 2

Input: "()()"

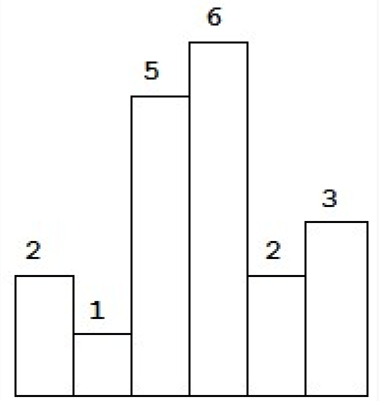
Output: 2

Input:
"(()(()))"

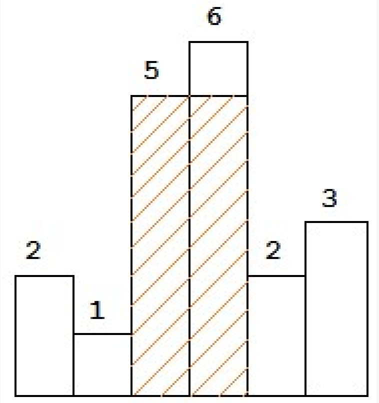
Output: 6

3. Largest Rectangle in Histogram

Given an array of integers **A** of size **N**. **A** represents a histogram i.e **A[i]** denotes height of the $\mathbf{i^{th}}$ histogram’s bar. Width of each bar is **1**.



Above is a histogram where width of each bar is 1, given height = `[2,1,5,6,2,3]`.



The largest rectangle is shown in the shaded area, which has area = `10` unit.
Find the area of largest rectangle in the histogram.

Input Format

```
The only argument given is the integer array A.
```

Output Format

```
Return the area of largest rectangle in the histogram.
```

For Example

```
Input 1:
    A = [2, 1, 5, 6, 2, 3]
Output 1:
    10
    Explanation 1:
        The largest rectangle is shown in the shaded area, which has area = 10 unit.
```

4. Sliding Window Maximum

Given an array of integers **A**. There is a sliding window of size **B** which is moving from the very left of the array to the very right. You can only see the w numbers in the window. Each time the sliding window moves rightwards by one position. You have to find the maximum for each window. The following example will give you more clarity.
The array **A** is `[1 3 -1 -3 5 3 6 7]`, and **B** is 3.

Window position	Max
[1 3 -1] -3 5 3 6 7	3
1 [3 -1 -3] 5 3 6 7	3
1 3 [-1 -3 5] 3 6 7	5
1 3 -1 [-3 5 3] 6 7	5
1 3 -1 -3 [5 3 6] 7	6
1 3 -1 -3 5 [3 6 7]	7

Return an array **C**, where **C[i]** is the maximum value of from **A[i]** to **A[i+B-1]**.
Note: If **B** > length of the array, return 1 element with the max of the array.

Input Format

```
The first argument given is the integer array A.
The second argument given is the integer B.
```

Output Format

```
Return an array C, where C[i] is the maximum value of from A[i] to A[i+B-1]
```

For Example

```
Input 1:
    A = [1, 3, -1, -3, 5, 3, 6, 7]
    B = 3
Output 1:
    C = [3, 3, 5, 5, 6, 7]
```

5. Reverse a stack using recursion.