1. Write a program to check for balanced brackets in an expression.. Balanced brackets are the pairs and the orders of “{“, “}”, “(“, “)”, “[“, “]” are correct in the given expression.

**For example**: [()]{}{[()()]()} is a balanced bracket.

          [({}]) is not a balanced bracket

You will be given a string, you need to check if the brackets are balanced or not. If they are balanced, print “YES”, otherwise print “NO”.

**Expected time complexity**: O(N)

|  |  |
| --- | --- |
| **Sample Input** | **Sample output** |
| ({(())[{}]) | NO |
| ({(())[{}]}) | YES |

**Reference**: <https://www.geeksforgeeks.org/check-for-balanced-parentheses-in-an-expression/>

1. Given N integers, the task is to insert those elements in the queue. Also, given M integers,your task is to find the frequency of each number of M in the Queue.

**Expected time complexity**: O(N)

|  |  |
| --- | --- |
| **Sample Input** | **Sample output** |
| 8  1 2 3 4 5 2 3 1  5  1 3 2 9 10 | 2  2  2  -1  -1 |
|  |

**Explanation:** Frequency of 1 is 2. Frequency of 3 is 2. Frequency of 2 is 2. Frequency of 9 is -1 and Frequency of 10 is  -1 (since 9 and 10 are not there in the queue).

**Reference**: [GFG-Queue Operations](https://practice.geeksforgeeks.org/problems/queue-operations/1?page=2&category%5b%5d=Queue&sortBy=submissions)

1. Write a program to sort a stack of integers.

You will be given a size N, and N integer values. You need to insert those values in a stack, and sort that stack and print it.

**Expected time complexity**: O(N\*N)

|  |  |
| --- | --- |
| **Sample Input** | **Sample output** |
| 5  1 8 5 4 2 | 1 2 4 5 8 |
| 8  5 1 4 7 9 2 5 4 | 1 2 4 4 5 5 7 9 |

**Reference**: <https://www.geeksforgeeks.org/sort-stack-using-temporary-stack/>

1. Write a program to reverse a queue of integers.

You will be given a size N, and N integer values. You need to insert those values in a queue, and reverse that queue and print it.

**Expected time complexity**: O(N)

|  |  |
| --- | --- |
| **Sample Input** | **Sample output** |
| 5  5 4 1 3 7 | 7 3 1 4 5 |
| 7  1 4 5 1 2 7 4 | 4 7 2 1 5 4 1 |

**Reference**: <https://www.geeksforgeeks.org/reversing-a-queue/>

1. Given a number N. The task is to generate and print all binary representations of decimal values from 1 to N.

**Expected time complexity**: O(N\*log2(N))

|  |  |
| --- | --- |
| **Sample Input** | **Sample output** |
| 2 | 1 10 |
| 5 | 1 10 11 100 101 |

**Reference**: [https://practice.geeksforgeeks.org/problems/generate-binary-numbers-1587115620/](https://practice.geeksforgeeks.org/problems/generate-binary-numbers-1587115620/1?page=1&difficulty%5b%5d=-2&difficulty%5b%5d=-1&category%5b%5d=Queue&sortBy=submissions)

1. Write a program to reverse the first K elements of a queue.

You will be given a size N and K. In the next line you will be given N integer values. You need to insert those values in a queue, and reverse the first K elements of that queue and print it.

**Expected time complexity**: O(N+K)

|  |  |
| --- | --- |
| **Sample Input** | **Sample output** |
| 10 5  10 20 30 40 50 60 70 80 90 100 | 50 40 30 20 10 60 70 80 90 100 |
| 7 4  1 2 3 4 5 6 7 | 4 3 2 1 5 6 7 |

**Reference**: <https://www.geeksforgeeks.org/reversing-first-k-elements-queue/>

1. Given a String. Reverse each word in it where the words are separated by dots.

**Expected Time Complexity:** O(N).

|  |  |
| --- | --- |
| **Sample Input** | **Sample output** |
| i.like.this.program.very.much | i.ekil.siht.margorp.yrev.hcum |
| pqr.mno | rqp.onm |

**Reference**: [https://practice.geeksforgeeks.org/problems/reverse-each-word-in-a-given-string1001](https://practice.geeksforgeeks.org/problems/reverse-each-word-in-a-given-string1001/1?page=2&category%5b%5d=Stack&sortBy=submissions)

1. You are given an array A of size N. You need to first push the elements of the array into a stack and then print the current minimum present in the stack at each pop until the stack becomes empty.

**Expected Time Complexity:** O(N).

|  |  |
| --- | --- |
| **Sample Input** | **Sample output** |
| 5  1 2 3 4 5 | 1 1 1 1 1 |
| 7  1 6 43 1 2 0 5 | 0 0 1 1 1 1 1 |

**Explanation 1:**

After pushing elements to the stack,

the stack will be "top -> 5, 4, 3, 2, 1"

Now, start popping elements from the stack:

popping 5: current min in the stack is 1.

popping 4: current  min in the stack is 1.

popping 3: current  min in the stack is 1.

popping 2: current  min in the stack is 1.

popping 1: current  min in the stack is 1.

**Explanation 2:**

After pushing the elements to the stack,

the stack will be “top -> 5->0->2->1->43->6->1”

Now, popping the elements from the stack:

popping 5: current min in the stack is 0.

popping 0: current min in the stack is 0.

popping 2: current min in the stack is 1.

popping 1: current  min in the stack is 1.

popping 43: current  min in the stack is 1.

popping 6: current  min in the stack is 1.

popping 1: current  min in the stack is 1.

**Reference**: [https://practice.geeksforgeeks.org/problems/get-min-at-pop](https://practice.geeksforgeeks.org/problems/get-min-at-pop/1?page=2&category%5b%5d=Stack&sortBy=submissions)

**এক্সট্রা কিছু প্রবলেম (এগুলো অপশনাল যারা এক্সট্রা আরো প্রবলেম সল্ভ করতে চাও তাদের জন্য) -**

<https://www.hackerearth.com/practice/data-structures/stacks/basics-of-stacks/practice-problems/>

<https://www.cs.princeton.edu/courses/archive/spr01/cs126/exercises/adt.html>