

# Algorithm Lab (Course Code: MC504)

## Assignment - 8

**Submission Deadline:** by 12:00 PM, (25/03/2023)

**Total Marks:** 30

---

### Instructions:

- Proper indentation is mandatory.
  - Program files **must** be compiled using **linux gcc compiler**.
  - **VERY IMPORTANT:** You must add comments whenever necessary, to make the code understandable.
  - Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
  - Take inputs from users. Make necessary assumptions if required.
  - **ANSWER FILE:** Source code: (file name) e.g. A8\_Q1.c, A8\_PP.c
- 

### Q1.

Chandu and Chandni talk on phone for a long time daily. Being afraid that someone will hear their private conversation, Chandu suggested Chandni an idea. He suggested that he will talk only with encrypted strings with her and only she would know how to decrypt the string. So that even if someone hears, He/She would not be able to anticipate their conversation.

Rules of encryption are as follows:

1. String on length N is assumed to be cyclic consisting of lower case English alphabets. (**cyclic string:** We call a string cyclic if it can be obtained by concatenating another string to itself many times (*for example*,  $s_2 = \text{"abcabcabcabc..."}$  is cyclic since it can be obtained from  $s_1 = \text{"abc"}$  in such a way)).
2. In each iteration, we pick the last character and put it in the starting of the string. For example: april performing iterations and collecting each string formed in a set until we get the original string. *Ex:* {april, lapri, ilapr, rilap, prila}
3. Sort the set of strings in lexicographically reverse order. *Ex:* {rilap, prila, lapri, ilapr, april}
4. Taking the last character of each string in the set is the encrypted string. *Ex:* pairl

Chandu also sends the position(K) of first letter in encrypted string from original string i.e 2 (p is on position 2 in original string and is the first character of encrypted string) Now, Chandni is

ofcourse not that brilliant to decrypt the strings in real time and understand what Chandu is saying. So, Chandu decided to write a program for the same.

Help Chandu write this program.

**(USE HEAP SORT)**

***Input:***

First line contains an integer t, which is the number of test cases. Next t lines contain an encrypted string and K as described above.

***Output:***

Print the decrypted string for each test case.

-----

**Sample Input**

```
2
d 1
pairl 2
```

**Sample Output**

```
d
april
```

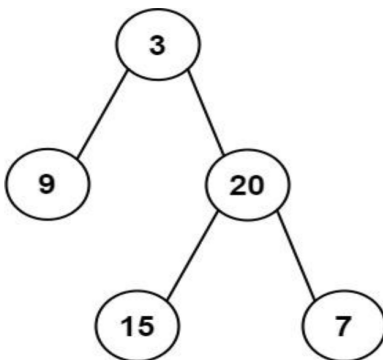
-----

**OR**

A Binary Tree is defined as a tree data structure where each node has at most 2 children. Since each element in a binary tree can have only 2 children, we typically name them the left and right child. The level order traversal of the tree is stored in an array representing the nodes of the tree. Starting from the root of a binary tree, return the sum of all left leaves.

A leaf is a node with no children. A left leaf is a leaf that is the left child of any node.

***Example 1:***



Input: root = [3,9,20,null,null,15,7]

Output: 24

Explanation: There are two left leaves in the binary tree, with values 9 and 15 respectively.

### **Example 2:**

Input: root = [1]

Output: 0

### **Constraints:**

- The number of nodes in the tree is in the range [1, 1000].
- $-1000 \leq \text{Node.val} \leq 1000$

## **PRACTICE PROBLEM (PP)**

You are given an integer array `score` of size `n`, where `score[i]` is the score of the  $i^{\text{th}}$  athlete in a competition. All the scores are guaranteed to be **unique**.

The athletes are placed based on their scores, where the 1st place athlete has the highest score, the 2nd place athlete has the 2nd highest score, and so on. The placement of each athlete determines their rank:

- The 1st place athlete's rank is "Gold Medal".
- The 2nd place athlete's rank is "Silver Medal".
- The 3rd place athlete's rank is "Bronze Medal".
- For the 4th place to the `n`th place athlete, their rank is their placement number (i.e., the `x`th place athlete's rank is "`x`").

Return an array `answer` of size `n` where `answer[i]` is the rank of the  $i^{\text{th}}$  athlete.

### **Example 1:**

**Input:** `score = [5,4,3,2,1]`

**Output:** `["Gold Medal","Silver Medal","Bronze Medal","4","5"]`

**Explanation:** The placements are [1st, 2nd, 3rd, 4th, 5th].

### **Example 2:**

**Input:** `score = [10,3,8,9,4]`

**Output:** `["Gold Medal","5","Bronze Medal","Silver Medal","4"]`

**Explanation:** The placements are `[1st, 5th, 3rd, 2nd, 4th]`.

**Note:** Use the **Q1** sorting algorithm in the **practice problem** question. You can also use the concept of Priority Queue for solving the practice problem.