**FOR BOTH EAGLE AND FLAMINGO - 100 Marks - 10 Hours**

Full stack application

Restaurant Management System

Problem Statement

Create an application to manage billing system in a Restaurant

To generate a bill

1. First the table is selected

2. Second the waiter serving that table is selected

3. Customer name and mobile is entered, the items ordered are selected from the menu

4. Bill is generated and saved in the backend

Flow

Frontend

1. Select table on screen one

2. Select waiter on screen two

3. Set user name, mobile, select items from menu on screen three, two sections on the third page, left for menu items, right for cart, with generate bill button

Backend - Database Tables for each one of these

1. Tables api - CRUD

2. Waiters api - CRUD

3. Menu api - CRUD

4. Orders(Bill) api - CRUD

"Tables" table fields

1. Seating strength

2. Table name

3. Floor number(table kept on which floor)

"Waiters" table fields

1. Waiter name

2. Waiter age

3. Waiter mobile

4. Waiter ratings

5. Waiter experience

"Menu" table fields

1. Item name

2. Cuisine name

3. Veg/Egg/Non-Veg

4. Item price

"Order(Bill)" table fields

1. User name

2. User mobile

3. Items ordered

4. Total price

5. Payment mode

6. Waiter id

7. Table id

<https://forms.gle/sMJeAS8ZgufRby7J6>

**FOR EAGLE ONLY - 50 Marks - 3 Hours**

1)

Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand.

(i.e., [0,1,2,4,5,6,7] might become [4,5,6,7,0,1,2]).

You are given a target value to search. If found in the array return its index, otherwise return -1.

You may assume no duplicate exists in the array.

Your algorithm's runtime complexity must be in the order of *O*(log *n*).

Example 1:

Input: nums = [4,5,6,7,0,1,2], target = 0

Output: 4

Example 2:

Input: nums = [4,5,6,7,0,1,2], target = 3

Output: -1

2)

Write an algorithm to determine if a number is "happy".

A happy number is a number defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers.

Example:

Input: 19

Output: true

Explanation:

12 + 92 = 82

82 + 22 = 68

62 + 82 = 100

12 + 02 + 02 = 1

3)

You are working at the cash counter at a fun-fair, and you have different types of coins available to you in infinite quantities. The value of each coin is already given. Can you determine the number of ways of making change for a particular number of units using the given types of coins?

For example, if you have types of coins, and the value of each type is given as respectively, you can make change for units in three ways: , , and .

Function Description

Complete the getWays function in the editor below. It must return an integer denoting the number of ways to make change.

getWays has the following parameter(s):

* n: an integer, the amount to make change for
* c: an array of integers representing available denominations

input Format

The first line contains two space-separated integers describing the respective values of n and m , where:

n is the number of units

m is the number of coin types

The second line contains m space-separated integers describing the respective values of each coin type :

c = [c[0], c[1], …. c[m-1]] (the list of distinct coins available in infinite amounts).

4)

Given a binary tree, find its maximum depth.

The maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.

Note: A leaf is a node with no children.

Solve using both recursive and iterative method.

5)

Reverse a singly linked list iteratively and recursively.

[**https://docs.google.com/forms/d/e/1FAIpQLSeyo\_jAd\_mI8R7CzLhjFECccP814bdU3fpmkMgfIzCgK0IyKw/viewform?vc=0&c=0&w=1**](https://docs.google.com/forms/d/e/1FAIpQLSeyo_jAd_mI8R7CzLhjFECccP814bdU3fpmkMgfIzCgK0IyKw/viewform?vc=0&c=0&w=1)