# Placement Questions

## SET-A

1 **Plus-one**

You are given a large integer represented as an integer array digits, where each digits[i] is the $i^{th}$git of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's.

Increment the large integer by one and return *the resulting array of digits*

Example 1:

```
Input: digits = [1,2,3]
Output: [1,2,4]
```

Explanation: The array represents the integer 123.
Incrementing by one gives 123 + 1 = 124.
Thus, the result should be [1,2,4].

Example 2:

```
Input: digits = [4,3,2,1]
Output: [4,3,2,2]
```

Explanation: The array represents the integer 4321.
Incrementing by one gives 4321 + 1 = 4322.
Thus, the result should be [4,3,2,2].

Example 3:

```
Input: digits = [9]
Output: [1,0]
```

Explanation: The array represents the integer 9.
Incrementing by one gives 9 + 1 = 10.
Thus, the result should be [1,0].

2 **Add Binary**

Given two binary strings a and b, return *their sum as a binary string*.

Example 1:

```
Input: a = "11", b = "1"
Output: "100"
```

Example 2:

```
Input: a = "1010", b = "1011"
Output: "10101"
```

Constraints:
1 <= a.length, b.length <= $10^4$
a and b consist only of '0' or '1' characters.
Each string does not contain leading zeros except for the zero itself.

3 Check if a List of integers contains only odd numbers?

####################################################################

## SET – B

### 1 Single Number

Given a non-empty array of integers nums, every element appears *twice* except for one. Find that single one.

Example 1:

```
Input: nums = [2,2,1]
Output: 1
```

Example 2:

```
Input: nums = [4,1,2,1,2]
Output: 4
```

Example 3:

```
Input: nums = [1]
Output: 1
```

### 2 Sort the People

You are given an array of strings names, and an array heights that consists of distinct positive integers. Both arrays are of length n. For each index i, names[i] and heights[i] denote the name and height of the i$^{th}$ person. Return names *sorted in descending order by the people's heights*.

Example 1:

```
Input: names = ["Mary","John","Emma"], heights = [180,165,170]
Output: ["Mary","Emma","John"]
```

Explanation: Mary is the tallest, followed by Emma and John.

Example 2:

```
Input: names = ["Alice","Bob","Bob"], heights = [155,185,150]
Output: ["Bob","Alice","Bob"]
```

Explanation: The first Bob is the tallest, followed by Alice and the second Bob.

Constraints:

n == names.length == heights.length<br>
1 <= n <= $10^3$
1 <= names[i].length <= 20
1 <= heights[i] <= $10^5$
names[i] consists of lower and upper case English letters.
All the values of heights are distinct.

### 3 Find factorial of an integer without using loop.

###################################################################

## SET-C

1 **House Robber**

You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed, the only constraint stopping you from robbing each of them is that adjacent houses have security systems connected and it will automatically contact the police if two adjacent houses were broken into on the same night.

Given an integer array nums representing the amount of money of each house, return *the maximum amount of money you can rob tonight without alerting the police*.

Example 1:

```
Input: nums = [1,2,3,1]
Output: 4
```

Explanation: Rob house 1 (money = 1) and then rob house 3 (money = 3).
Total amount you can rob = 1 + 3 = 4.

Example 2:

```
Input: nums = [2,7,9,3,1]
Output: 12
```

Explanation: Rob house 1 (money = 2), rob house 3 (money = 9) and rob house 5 (money = 1).
Total amount you can rob = 2 + 9 + 1 = 12.

2 **Count the Number of Consistent Strings**

You are given a string allowed consisting of distinct characters and an array of strings words. A string is consistent if all characters in the string appear in the string allowed.

Return *the number of consistent strings in the array* words.

Example 1:

```
Input: allowed = "ab", words = ["ad","bd","aaab","baa","badab"]
Output: 2
```

Explanation: Strings "aaab" and "baa" are consistent since they only contain characters 'a' and 'b'.

Example 2:

```
Input: allowed = "abc", words = ["a","b","c","ab","ac","bc","abc"]
Output: 7
```

Explanation: All strings are consistent.

Example 3:

```
Input: allowed = "cad", words = ["cc","acd","b","ba","bac","bad","ac","d"]
Output: 4
```

Explanation: Strings "cc", "acd", "ac", and "d" are consistent.

Constraints:

1 <= words.length <= $10^4$
1 <= allowed.length <= 26
1 <= words[i].length <= 10
The characters in allowed are distinct.
words[i] and allowed contain only lowercase English letters.

3 Fibonacci Series without using recursion.

##########################################################################

## SET-D

1 **Rotate Array**

Given an array, rotate the array to the right by k steps, where k is non-negative.

Example 1:
Input: nums = [1,2,3,4,5,6,7], k = 3
Output: [5,6,7,1,2,3,4]

Explanation:
rotate 1 steps to the right: [7,1,2,3,4,5,6]
rotate 2 steps to the right: [6,7,1,2,3,4,5]
rotate 3 steps to the right: [5,6,7,1,2,3,4]

Example 2:
Input: nums = [-1,-100,3,99], k = 2
Output: [3,99,-1,-100]

Explanation:
rotate 1 steps to the right: [99,-1,-100,3]
rotate 2 steps to the right: [3,99,-1,-100]

2 **Check If Two String Arrays are Equivalent**

Given two string arrays word1 and word2, return true *if the two arrays represent the same string,and* false *otherwise.* A string is represented by an array if the array elements concatenated in order forms the string.

Example 1:
Input: word1 = ["ab", "c"], word2 = ["a", "bc"]
Output: true

Explanation:
word1 represents string "ab" + "c" -> "abc"
word2 represents string "a" + "bc" -> "abc"
The strings are the same, so return true.

Example 2:
Input: word1 = ["a", "cb"], word2 = ["ab", "c"]
Output: false

Example 3:
Input: word1 = ["abc", "d", "defg"], word2 = ["abcddefg"]
Output: true

Constraints:
1 <= word1.length, word2.length <= $10^3$
1 <= word1[i].length, word2[i].length <= $10^3$
1 <= sum(word1[i].length), sum(word2[i].length) <= $10^3$
word1[i] and word2[i] consist of lowercase letters.

3 Print all prime numbers less than or equal to N using recursion.

######################################################################