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`
$1 <= n <= 10^3$
`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.

##

SET-E

1 **Migratory-birds**

Given an array of bird sightings where every element represents a bird type id, determine the id of the most frequently sighted type. If more than 1 type has been spotted that maximum amount, return the smallest of their ids.

Example

```
arr = [1,1,2,2,3]
```

There are two each of types 1 and 2, and one sighting of type 3. Pick the lower of the two types seen twice: type 1.

Input Format

The first line contains an integer, **n**, the size of arr.
The second line describes **arr** as **n** space-separated integers, each a type number of the bird sighted.

Sample Input 0

```
6
1 4 4 4 5 3
```

Sample Output 0

```
4
```

Explanation 0

The different types of birds occur in the following frequencies:
Type 1:1 bird
Type 2:0 birds
Type 3:1 bird
Type 4:3 birds
Type 5:1 bird

The type number that occurs at the highest frequency is type 4, so we print 4 as our answer.

Sample Input 1

```
11
1 2 3 4 5 4 3 2 1 3 4
```

Sample Output 1

```
3
```

Explanation 1

The different types of birds occur in the following frequencies:
Type 1:2 birds
Type 2:2 birds
Type 3:3 birds
Type 4:3 birds
Type 5:1 birds
Two types have a frequency of 3, and the lower of those is type 3.

2 **Check if the Sentence Is Pangram**

A pangram is a sentence where every letter of the English alphabet appears at least once. Given a string sentence containing only lowercase English letters, return true *if* sentence *is a pangram, or* false *otherwise.*

Example 1:

```
Input: sentence = "thequickbrownfoxjumpsoverthelazydog"
Output: true
```

Explanation: sentence contains at least one of every letter of the English alphabet.

Example 2:

```
Input: sentence = "leetcode"
Output: false
```

Constraints:

1 <= sentence.length <= 1000

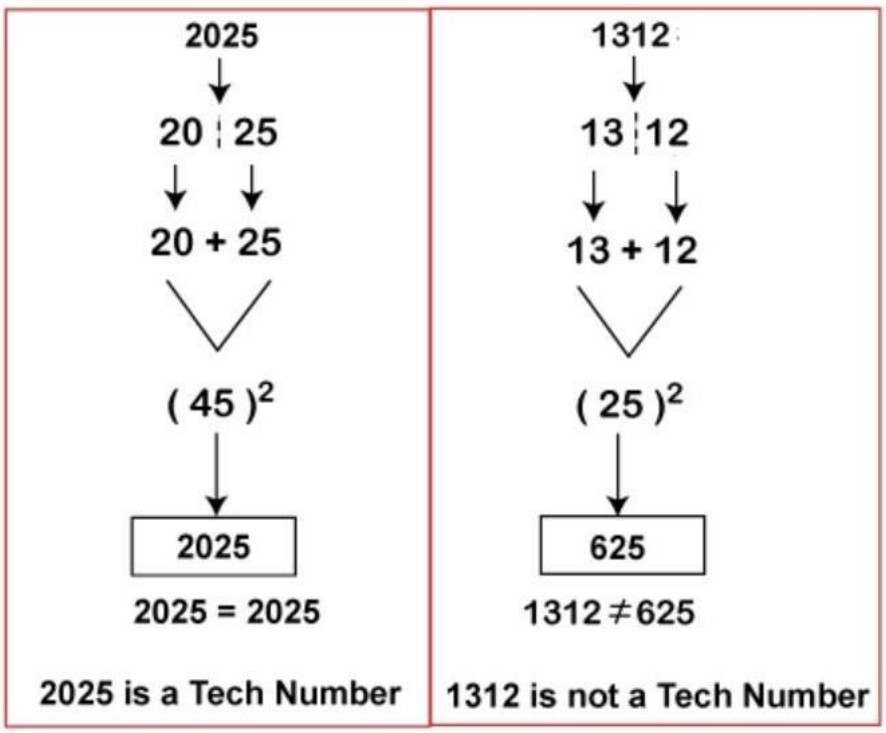
sentence consists of lowercase English letters.

3 **Tech Number**

A number is called a **tech number** if the given number has an even number of digits and the number can be divided exactly into two parts from the middle. After equally dividing the number, sum up the numbers and find the square of the sum. If we get the number itself as square, the given number is a tech number, else, not a tech number. For example, 3025 is a tech number.

Tech Number Example

Let's take an example and check 2025 and 1312 are tech numbers or not.



##

SET-E

1 **Min-max-sum**

Given five positive integers, find the minimum and maximum values that can be calculated by summing exactly four of the five integers. Then print the respective minimum and maximum values as a single line of two space-separated long integers.

Example
arr = [1,3,5,7,9]

The minimum sum is 1 + 3 + 5 + 7 = 16 and the maximum sum is 3 + 5 + 7 + 9 = 24. The function prints

16 24

Input Format
A single line of five space-separated integers.

Output Format
Print two space-separated long integers denoting the respective minimum and maximum values that can be calculated by summing exactly *four* of the five integers. (The output can be greater than a 32 bit integer.)

Sample Input
1 2 3 4 5

Sample Output
10 14

Explanation
The numbers are 1, 2, 3, 4, and 5. Calculate the following sums using four of the five integers:
Sum everything except 1, the sum is 2 + 3 + 4 + 5 = 14.
Sum everything except 2, the sum is 1 + 3 + 4 + 5 = 13.
Sum everything except 3, the sum is 1 + 2 + 4 + 5 = 12.
Sum everything except 4, the sum is 1 + 2 + 3 + 5 = 11.
Sum everything except 5, the sum is 1 + 2 + 3 + 4 = 10.

2 **Truncate Sentence**

A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each of the words consists of only uppercase and lowercase English letters (no punctuation).
For example, "Hello World", "HELLO", and "hello world hello world" are all sentences.
You are given a sentence s and an integer k. You want to truncate s such that it contains only the first k words. Return s *after truncating it.*

Example 1:
Input: s = "Hello how are you Contestant", k = 4
Output: "Hello how are you"

Explanation:
The words in s are ["Hello", "how" "are", "you", "Contestant"].The
first 4 words are ["Hello", "how", "are", "you"].
Hence, you should return "Hello how are you".

Example 2:

Input: s = "What is the solution to this problem", k = 4

Explanation:

The words in s are ["What", "is" "the", "solution", "to", "this", "problem"].
The first 4 words are ["What", "is", "the", "solution"].

Example 3:

Input: s = "chopper is not a tanuki", k = 5

Constraints:

1 <= s.length <= 500

k is in the range [1, the number of words in s].

s consist of only lowercase and uppercase English letters and spaces.The words in s are separated by a single space.

##########

SET-G

1 Array-left-rotation

A *left rotation* operation on an array of size n shifts each of the array's elements 1 unit to the left. Givenan integer, d, rotate the array that many steps left and return the result.

Exampled = 2
arr =

After 2 rotations, arr=[3,4,5,1,2].

Input Format
The first line contains two space-separated integers that denote n, the number of integers, and d, thenumber of left rotations to perform.
The second line contains n space-separated integers that describe arr[].

Sample Input5
4

Sample Output
5 1 2 3 4

Explanation
To perform d=4 left rotations, the array undergoes the following sequence of changes:[1,2,3,4,5] -> [2,3,4,5,1] -> [3,4,5,1,2] -> [4,5,1,2,3].

2 Day of the Year

Given a string `date` representing a Gregorian calendar date formatted as `YYYY-MM-DD`, return *the day number of the year*.

Example 1:
```
Input: date = "2019-01-09"
Output: 9
```

Explanation: Given date is the 9th day of the year in 2019.

Example 2:
```
Input: date = "2019-02-10"
Output: 41
```

Constraints:
`date.length == 10`
`date[4] == date[7] == '-'`, and all other `date[i]`'s are digits
`date` represents a calendar date between Jan 1st, 1900 and Dec 31th, 2019.

3 **ISBN Number**

ISBN is another special number. **ISBN** stands for the **International Standard Book Number** that is carried by almost each every book. The ISBN is a ten-digit unique number. With the help of the ISBN, we can easily find any book. The ISBN number is a legal number when 1*Digit1 + 2*Digit2 + 3*Digit3 + 4*Digit4 + 5*Digit5 + 6*Digit6 + 7*Digit7 + 8*Digit8 + 9*Digit9 + 10*Digit10 is divisible by 11. The digits are taken from right to left. So, if the ten-digit number is 7426985414, Digit1 and Digit10 will be 4 and 7, respectively.
Let's take two numbers and check whether the numbers are legal or not.

Example:
Number1: 8147852369
Sum = (1*9) + (2*6) + (3*3) + (4*2) + (5*5) + (6*8) + (7*7) + (8*4) + (9*1) + (10*8)Sum
= 9 + 12 + 9 + 8 + 25 + 48 + 49 + 32 + 9 + 80
Sum = 281
Now, we divide the sum with 11 and check whether the remainder is 0 or not. If the remainder is 0, the number is a legal ISBN.

rem = 281 % 11
rem = 6 != 0

Number **8147852369** is not a legal ISBN because the remainder is not equal to the 0.

Number2: 1259060977
Sum = (1*10) + (2*9) + (5*8) + (9*7) + (0*6) + (6*5) + (0*4) + (9*3) + (7*2) + (7*1)Sum
= 10 + 18 + 40 + 63 + 0 + 30 + 0 + 27 + 14 + 7
Sum = 209
Now, we divide the sum with 11 and check whether the remainder is 0 or not.
rem = 209 % 11
rem = 0
Number **1259060977** is a legal ISBN because the remainder is equal to 0.
##

SET-H

1 **Birthday-cake-candles**

You are in charge of the cake for a child's birthday. You have decided the cake will have one candle for each year of their total age. They will only be able to blow out the tallest of the candles. Count howmany candles are tallest.

Example

```
Candles = [4,4,1,3]
```

Example

The maximum height candles are 4 units high. There are 2 of them, so return 2.

Input Format

The first line contains a single integer, n, the size of candles[].

The second line contains n space-separated integers, where each integer i describes the heightof candles[i].

Sample Input 0

```
4
3 2 1 3
```

Sample Output 0

```
2
```

Explanation 0

Candle heights are [3,2,1,3] . The tallest candles are 3 units, and there are 2 of them.

2 **Jewels and Stones**

You're given strings jewels representing the types of stones that are jewels, and stones representing the stones you have. Each character in stones is a type of stone you have. You want to know how many of the stones you have are also jewels.

Letters are case sensitive, so "a" is considered a different type of stone from "A".

Example 1:

Input: jewels = "aA", stones = "aAAbbbb"

Output: 3

Example 2:

Input: jewels = "z", stones = "ZZ"

Output: 0

Constraints:

1 <= jewels.length, stones.length <= 50

jewels and stones consist of only English letters.All the characters of jewels are unique.

3 Add two numbers without using + operator.

##########

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`
$1 <= n <= 10^3$
`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.

##

SET-E

1 **Migratory-birds**

Given an array of bird sightings where every element represents a bird type id, determine the id of the most frequently sighted type. If more than 1 type has been spotted that maximum amount, return the smallest of their ids.

Example

```
arr = [1,1,2,2,3]
```

There are two each of types 1 and 2, and one sighting of type 3. Pick the lower of the two types seen twice: type 1.

Input Format

The first line contains an integer, **n**, the size of arr.
The second line describes **arr** as **n** space-separated integers, each a type number of the bird sighted.

Sample Input 0

```
6
1 4 4 4 5 3
```

Sample Output 0

```
4
```

Explanation 0

The different types of birds occur in the following frequencies:
Type 1:1 bird
Type 2:0 birds
Type 3:1 bird
Type 4:3 birds
Type 5:1 bird

The type number that occurs at the highest frequency is type 4, so we print 4 as our answer.

Sample Input 1

```
11
1 2 3 4 5 4 3 2 1 3 4
```

Sample Output 1

```
3
```

Explanation 1

The different types of birds occur in the following frequencies:
Type 1:2 birds
Type 2:2 birds
Type 3:3 birds
Type 4:3 birds
Type 5:1 birds
Two types have a frequency of 3, and the lower of those is type 3.

2 **Check if the Sentence Is Pangram**

A pangram is a sentence where every letter of the English alphabet appears at least once. Given a string sentence containing only lowercase English letters, return true *if* sentence *is a pangram, or* false *otherwise.*

Example 1:

```
Input: sentence = "thequickbrownfoxjumpsoverthelazydog"
Output: true
```

Explanation: sentence contains at least one of every letter of the English alphabet.

Example 2:

```
Input: sentence = "leetcode"
Output: false
```

Constraints:

1 <= sentence.length <= 1000

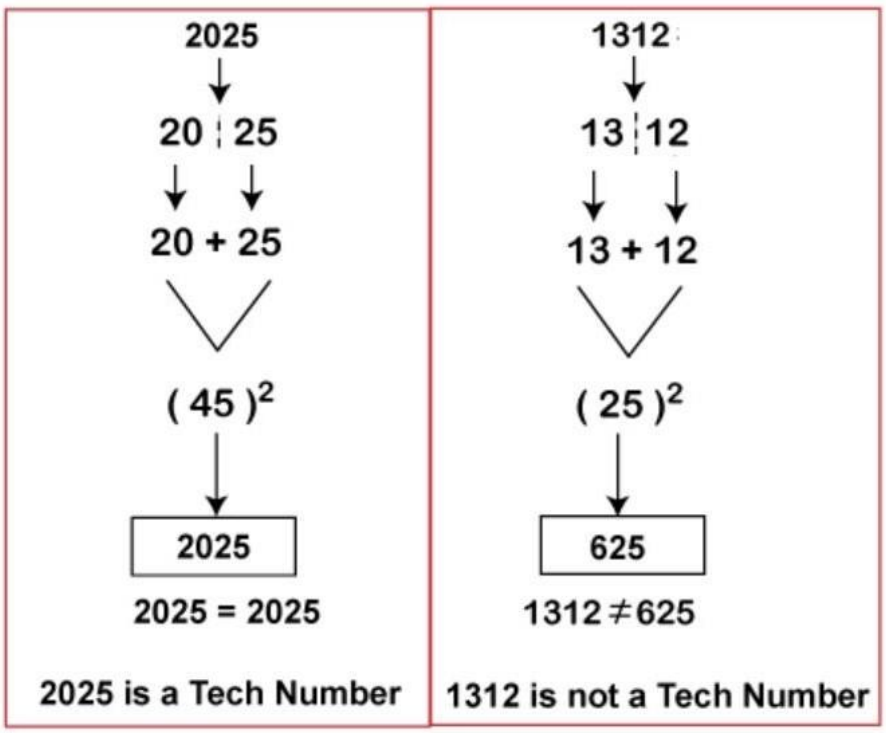
sentence consists of lowercase English letters.

3 **Tech Number**

A number is called a **tech number** if the given number has an even number of digits and the number can be divided exactly into two parts from the middle. After equally dividing the number, sum up the numbers and find the square of the sum. If we get the number itself as square, the given number is a tech number, else, not a tech number. For example, 3025 is a tech number.

Tech Number Example

Let's take an example and check 2025 and 1312 are tech numbers or not.



##

SET-E

1 **Min-max-sum**

Given five positive integers, find the minimum and maximum values that can be calculated by summing exactly four of the five integers. Then print the respective minimum and maximum values as a single line of two space-separated long integers.

Example
arr = [1,3,5,7,9]

The minimum sum is 1 + 3 + 5 + 7 = 16 and the maximum sum is 3 + 5 + 7 + 9 = 24. The function prints

16 24

Input Format
A single line of five space-separated integers.

Output Format
Print two space-separated long integers denoting the respective minimum and maximum values that can be calculated by summing exactly *four* of the five integers. (The output can be greater than a 32 bit integer.)

Sample Input
1 2 3 4 5

Sample Output
10 14

Explanation
The numbers are 1, 2, 3, 4, and 5. Calculate the following sums using four of the five integers:
Sum everything except 1, the sum is 2 + 3 + 4 + 5 = 14.
Sum everything except 2, the sum is 1 + 3 + 4 + 5 = 13.
Sum everything except 3, the sum is 1 + 2 + 4 + 5 = 12.
Sum everything except 4, the sum is 1 + 2 + 3 + 5 = 11.
Sum everything except 5, the sum is 1 + 2 + 3 + 4 = 10.

2 **Truncate Sentence**

A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each of the words consists of only uppercase and lowercase English letters (no punctuation).
For example, "Hello World", "HELLO", and "hello world hello world" are all sentences.
You are given a sentence s and an integer k. You want to truncate s such that it contains only the first k words. Return s *after truncating it.*

Example 1:
Input: s = "Hello how are you Contestant", k = 4
Output: "Hello how are you"

Explanation:
The words in s are ["Hello", "how" "are", "you", "Contestant"].The
first 4 words are ["Hello", "how", "are", "you"].
Hence, you should return "Hello how are you".

Example 2:

Input: s = "What is the solution to this problem", k = 4

Explanation:

The words in s are ["What", "is" "the", "solution", "to", "this", "problem"].
The first 4 words are ["What", "is", "the", "solution"].

Example 3:

Input: s = "chopper is not a tanuki", k = 5

Constraints:

1 <= s.length <= 500

k is in the range [1, the number of words in s].

s consist of only lowercase and uppercase English letters and spaces.The words in s are separated by a single space.

##########

SET-G

1 Array-left-rotation

A *left rotation* operation on an array of size n shifts each of the array's elements 1 unit to the left. Givenan integer, d, rotate the array that many steps left and return the result.

Exampled = 2
arr =

After 2 rotations, arr=[3,4,5,1,2].

Input Format
The first line contains two space-separated integers that denote n, the number of integers, and d, thenumber of left rotations to perform.
The second line contains n space-separated integers that describe arr[].

Sample Input5
4

Sample Output
5 1 2 3 4

Explanation
To perform d=4 left rotations, the array undergoes the following sequence of changes:[1,2,3,4,5] -> [2,3,4,5,1] -> [3,4,5,1,2] -> [4,5,1,2,3].

2 Day of the Year

Given a string `date` representing a Gregorian calendar date formatted as `YYYY-MM-DD`, return *the day number of the year*.

Example 1:
```
Input: date = "2019-01-09"
Output: 9
```

Explanation: Given date is the 9th day of the year in 2019.

Example 2:
```
Input: date = "2019-02-10"
Output: 41
```

Constraints:
`date.length == 10`
`date[4] == date[7] == '-'`, and all other `date[i]`'s are digits
`date` represents a calendar date between Jan 1st, 1900 and Dec 31th, 2019.

3 **ISBN Number**

ISBN is another special number. **ISBN** stands for the **International Standard Book Number** that is carried by almost each every book. The ISBN is a ten-digit unique number. With the help of the ISBN, we can easily find any book. The ISBN number is a legal number when 1*Digit1 + 2*Digit2 + 3*Digit3 + 4*Digit4 + 5*Digit5 + 6*Digit6 + 7*Digit7 + 8*Digit8 + 9*Digit9 + 10*Digit10 is divisible by 11. The digits are taken from right to left. So, if the ten-digit number is 7426985414, Digit1 and Digit10 will be 4 and 7, respectively.
Let's take two numbers and check whether the numbers are legal or not.

Example:
Number1: 8147852369
Sum = (1*9) + (2*6) + (3*3) + (4*2) + (5*5) + (6*8) + (7*7) + (8*4) + (9*1) + (10*8)Sum
= 9 + 12 + 9 + 8 + 25 + 48 + 49 + 32 + 9 + 80
Sum = 281
Now, we divide the sum with 11 and check whether the remainder is 0 or not. If the remainder is 0, the number is a legal ISBN.

rem = 281 % 11
rem = 6 != 0

Number **8147852369** is not a legal ISBN because the remainder is not equal to the 0.

Number2: 1259060977
Sum = (1*10) + (2*9) + (5*8) + (9*7) + (0*6) + (6*5) + (0*4) + (9*3) + (7*2) + (7*1)Sum
= 10 + 18 + 40 + 63 + 0 + 30 + 0 + 27 + 14 + 7
Sum = 209
Now, we divide the sum with 11 and check whether the remainder is 0 or not.
rem = 209 % 11
rem = 0
Number **1259060977** is a legal ISBN because the remainder is equal to 0.
##

SET-H

1 **Birthday-cake-candles**

You are in charge of the cake for a child's birthday. You have decided the cake will have one candle for each year of their total age. They will only be able to blow out the tallest of the candles. Count howmany candles are tallest.

Example

```
Candles = [4,4,1,3]
```

Example

The maximum height candles are 4 units high. There are 2 of them, so return 2.

Input Format

The first line contains a single integer, n, the size of candles[].

The second line contains n space-separated integers, where each integer i describes the heightof candles[i].

Sample Input 0

```
4
3 2 1 3
```

Sample Output 0

```
2
```

Explanation 0

Candle heights are [3,2,1,3] . The tallest candles are 3 units, and there are 2 of them.

2 **Jewels and Stones**

You're given strings jewels representing the types of stones that are jewels, and stones representing the stones you have. Each character in stones is a type of stone you have. You want to know how many of the stones you have are also jewels.

Letters are case sensitive, so "a" is considered a different type of stone from "A".

Example 1:

Input: jewels = "aA", stones = "aAAbbbb"

Output: 3

Example 2:

Input: jewels = "z", stones = "ZZ"

Output: 0

Constraints:

1 <= jewels.length, stones.length <= 50

jewels and stones consist of only English letters.All the characters of jewels are unique.

3 Add two numbers without using + operator.

##########