

ACCENTURE CODING QUESTIONS

Question -1 : Find the weird number from the given array. *A weird number is a number which is not divisible by "K" given to us but whose sum of Digits of number is divisible by K.*

Question -2 : *Find the Longest Subsequence from the given array whose sum of consecutive elements in array is divisible by K given to us ?*

QUESTION -3: Given an array of integers nums and an integer target, return *indices of the two numbers such that they add up to target.*

You may assume that each input would have **exactly one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

Input: nums = [2,7,11,15], target = 9

Output: [0,1]

Explanation: Because $\text{nums}[0] + \text{nums}[1] == 9$, we return [0, 1].

Example 2:

Input: nums = [3,2,4], target = 6

Output: [1,2]

Example 3:

Input: nums = [3,3], target = 6

Output: [0,1]

Constraints:

- $2 \leq \text{nums.length} \leq 10^4$
- $-10^9 \leq \text{nums}[i] \leq 10^9$
- $-10^9 \leq \text{target} \leq 10^9$
- **Only one valid answer exists.**

QUESTION -4: 3. Longest Substring Without Repeating Characters

Given a string s, find the length of the **longest substring**

without repeating characters.

Example 1:**Input:** s = "abcabcbb"**Output:** 3**Explanation:** The answer is "abc", with the length of 3.**Example 2:****Input:** s = "bbbbbb"**Output:** 1**Explanation:** The answer is "b", with the length of 1.**Example 3:****Input:** s = "pwwkew"**Output:** 3**Explanation:** The answer is "wke", with the length of 3.

Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

Constraints:

- $0 \leq \text{s.length} \leq 5 * 10^4$
- s consists of English letters, digits, symbols and spaces.

QUESTION -5: Roman to Integer

Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

Symbol Value

I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, 2 is written as II in Roman numeral, just two ones added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is

before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

- I can be placed before V (5) and X (10) to make 4 and 9.
- X can be placed before L (50) and C (100) to make 40 and 90.
- C can be placed before D (500) and M (1000) to make 400 and 900.

Given a roman numeral, convert it to an integer.

Example 1:

Input: s = "III"

Output: 3

Explanation: III = 3.

Example 2:

Input: s = "LVIII"

Output: 58

Explanation: L = 50, V = 5, III = 3.

Example 3:

Input: s = "MCMXCIV"

Output: 1994

Explanation: M = 1000, CM = 900, XC = 90 and IV = 4.

Constraints:

- $1 \leq s.length \leq 15$
- s contains only the characters ('I', 'V', 'X', 'L', 'C', 'D', 'M').
- It is **guaranteed** that s is a valid roman numeral in the range [1, 3999].

Question -6 : Longest Common Prefix

Write a function to find the longest common prefix string amongst an array of strings.

If there is no common prefix, return an empty string "".

Example 1:

Input: strs = ["flower", "flow", "flight"]

Output: "fl"

Example 2:**Input:** strs = ["dog","racecar","car"]**Output:** ""**Explanation:** There is no common prefix among the input strings.**Constraints:**

- $1 \leq \text{strs.length} \leq 200$
- $0 \leq \text{strs}[i].\text{length} \leq 200$
- `strs[i]` consists of only lowercase English letters.

Question -7 : Longest Common Prefix

Write a function to find the longest common prefix string amongst an array of strings.

If there is no common prefix, return an empty string "".

Example 1:**Input:** strs = ["flower","flow","flight"]**Output:** "fl"**Example 2:****Input:** strs = ["dog","racecar","car"]**Output:** ""**Explanation:** There is no common prefix among the input strings.**Constraints:**

- $1 \leq \text{strs.length} \leq 200$
- $0 \leq \text{strs}[i].\text{length} \leq 200$
- `strs[i]` consists of only lowercase English letters.

Question -8: [319. Bulb Switcher](#)

Medium

Topics

Companies

There are n bulbs that are initially off. You first turn on all the bulbs, then you turn off every second bulb.

On the third round, you toggle every third bulb (turning on if it's off or turning off if it's on). For the i^{th} round, you toggle every i bulb. For the n^{th} round, you only toggle the last bulb.

Return *the number of bulbs that are on after n rounds*.

Example 1:



Input: $n = 3$

Output: 1

Explanation: At first, the three bulbs are [off, off, off].

After the first round, the three bulbs are [on, on, on].

After the second round, the three bulbs are [on, off, on].

After the third round, the three bulbs are [on, off, off].

So you should return 1 because there is only one bulb is on.

Example 2:

Input: $n = 0$

Output: 0

Example 3:

Input: $n = 1$

Output: 1

Question -9: Find the Original Number from Multiple Flipped Bits

Given a number n, k places of the number are flipped resulting in number x., the number n is flipped two more times in k places resulting in number y and z. remember that a place where bit is once flipped is never to be flipped again. simply put, bit flip takes place at $k*3$ unique places. Find the original number n.

Example 1:

Given $x=9$, $y=17$, $z=29$, $k=1$

$x=01001$

$y=10001$

$z=11101$

since $k=1$ only one bit is flipped from the original number

To get x from n 5th position from left is flipped

To get y from n 4th position from left is flipped

To get z from n 3rd position from left is flipped

so, original number would be 25 : 11001

Example 2:

Given $x=11$, $y=16$, $z=61$, $k=2$

$x=001011$

$y=010000$

$z=111101$

Since $k=2$, two bits are flipped from the original number.

To get x from n 5th and 2nd position from left is flipped

To get y from n 4th and 1st position from left is flipped

To get z from n 3rd and 6th position from left is flipped

So, original number would be 25 : 11001

he idea is:

To find out the places where bits were flipped for one number so that the the bits in those places could be flipped again to get the original number

question-10: Problem Description:

The Binary number system only uses two digits, 0 and 1 and the number system can be called binary string. You are required to implement the following function:

```
int OperationsBinaryString(char* str);
```

The function accepts a string str as its argument. The string str consists of binary digits separated with an alphabet as follows:

– A denotes AND operation

- B denotes OR operation
- C denotes XOR Operation

You are required to calculate the result of the string str, scanning the string to right taking one operation at a time, and return the same.

Note: No order of priorities of operations is required.

Length of str is odd.

If str is NULL or None (in case of Python), return -1.

Input:

1C0C1C1A0B1

Output:

1

Question 11: The function accepts two positive integers 'r' and 'unit' and a positive integer array 'arr' of size 'n' as its argument 'r' represents the number of rats present in an area, 'unit' is the amount of food each rat consumes and each ith element of array 'arr' represents the amount of food present in 'i+1' house number, where $0 \leq i$.

Note:

Return -1 if the array is null.

Return 0 if the total amount of food from all houses is not sufficient for all the rats.

Computed values lie within the integer range.

Example:

Input:

r: 7

unit: 2

n: 8

arr: 2 8 3 5 7 4 1 2

Output:

4

Explanation:

Total amount of food required for all rats = $r * \text{unit}$

= $7 * 2 = 14$.

The amount of food in 1st houses = $2+8+3+5 = 18$. Since, the amount of food in 1st 4 houses is sufficient for all the rats. Thus, output is 4.

Question 12: You are given a function,

int findCount(int arr[], int length, int num, int diff);

The function accepts an integer array 'arr', its length and two integer variables 'num' and 'diff'.

Implement this function to find and return the number of elements of 'arr' having an absolute difference of less than or equal to 'diff' with 'num'.

Note: In case there is no element in 'arr' whose absolute difference with 'num' is less than or equal to 'diff', return -1.

Example:

Input:

arr: 12 3 14 56 77 13

num: 13

diff: 2

Output:

3

Explanation:

Elements of 'arr' having absolute difference of less than or equal to 'diff' i.e. 2 with 'num' i.e. 13 are 12, 13 and 14.

Question 13:

Implement the following Function

def ProductSmallestPair(sum, arr)

The function accepts an integer sum and an integer array arr of size n.

Implement the function to find the pair, (arr[j], arr[k]) where $j \neq k$, such that arr[j] and arr[k] are the least two elements of array ($\text{arr}[j] + \text{arr}[k] \leq \text{sum}$) and return the product of element of this pair.

Note:

Return -1 if array is empty or if $n < 2$

Return 0, if no such pairs found.

All computed values lie within integer range.

Example:

Input

sum:9

Arr:5 2 4 3 9 7 1

Output

2

Question 14: N-base notation is a system for writing numbers that uses only n different symbols.

These symbols are the first n symbols from the given notation list(Including the symbol for o)

Decimal to n base notation are (0:0, 1:1, 2:2, 3:3, 4:4, 5:5, 6:6, 7:7, 8:8, 9:9, 10:A, 11:B and so on upto 35:Z).

Implement the following function

Char* DectoNBase(int n, int num):

The function accept positive integer n and num Implement the function to calculate the n-base equivalent of num and return the same as a string

Steps:

- Divide the decimal number by n,Treat the division as the integer division
- Write the the remainder (in n-base notation)
- Divide the quotient again by n, Treat the division as integer division
- Repeat step 2 and 3 until the quotient is 0
- The n-base value is the sequence of the remainders from last to first

Assumption:

$$1 < n \leq 36$$

Example

Input

n: 12

num: 718

Output

4BA

Explanation:

- num = 718, divisor = 12, quotient=59, remainder=10(A).
- num = 59, divisor = 12, quotient=4, remainder=11(B).
- num = 4, divisor = 12, quotient=0, remainder=4(A).

question -15: A carry is a digit that is transferred to left if sum of digits exceeds 9 while adding two numbers from right-to-left one digit at a time.

You are required to implement the following function.

Int NumberOfCarries(int num1 , int num2);

The function accepts two numbers 'num1' and 'num2' as its arguments. You are required to calculate and return the total number of carries generated while adding digits of two numbers 'num1' and ' num2'.

Assumption: num1, num2>=0

Example:

Input

Num 1: 451

Num 2: 349

Output

2

Explanation:

Adding 'num 1' and 'num 2' right-to-left results in 2 carries since (1+9) is 10. 1 is carried and (5+4=1) is 10, again 1 is carried. Hence 2 is returned.

Question 16: You are given a function,

```
void *ReplaceCharacter(Char str[], int n, char ch1, char ch2);
```

The function accepts a string 'str' of length n and two characters 'ch1' and 'ch2' as its argument.

Implement the function to modify and return the string 'str' in such a way that all occurrences of 'ch1' in the original string are replaced by 'ch2' and all occurrences of 'ch2' in the original string are replaced by 'ch1'.

Assumption: String Contains only lower-case alphabetical letters.

Note:

Return null if the string is null.

If both characters are not present in the string or both of them are the same , then return the string unchanged.

Example:

Input:

Str: apples

ch1:a

ch2:p

Output:

paales

Explanation:

'A' in the original string is replaced with 'p' and 'p' in the original string is replaced with 'a', thus output is paales.

Question 17: You are given a function:

```
Int MaxExponents (int a , int b);
```

You have to find and return the number between 'a' and 'b' (range inclusive on both ends) which has the maximum exponent of 2.

The algorithm to find the number with maximum exponent of 2 between the given range is

- Loop between 'a' and 'b'. Let the looping variable be 'i'.
- Find the exponent (power) of 2 for each 'i' and store the number with maximum exponent of 2 so far in a variable , let say 'max'. Set 'max' to 'i' only if 'i' has more exponent of 2 than 'max'.

- Return 'max'.

Assumption: $a < b$

Note: If two or more numbers in the range have the same exponents of 2, return the small number.

Example

Input:

7

12

Output:

8

Explanation:

Exponents of 2 in:

7-0

8-3

9-0

10-1

11-0

12-2

Hence the maximum exponent if two is of 8.

Question 18: You are required to input the size of the matrix then the elements of matrix, then you have to divide the main matrix in two sub matrices (even and odd)

in such a way that element at 0 index will be considered as even and element at 1st index will be considered as odd and so on.

Then you have sort the even and odd matrices in ascending order then print the sum of second largest number from both the matrices.

Example:

enter the size of array : 5

enter element at 0 index : 3

enter element at 1 index : 4

enter element at 2 index : 1

enter element at 3 index : 7

enter element at 4 index : 9

Sorted even array : 1 3 9

Sorted odd array : 4 7

7

Question 19: You are required to implement the following function:

Int Calculate(int m, int n);

The function accepts 2 positive integers 'm' and 'n' as its arguments. You are required to calculate the sum of numbers divisible both by 3 and 5, between 'm' and 'n' both inclusive and return the same.

Note:

$0 < m \leq n$

Example

Input:

m : 12

n : 50

Output

90

Explanation:

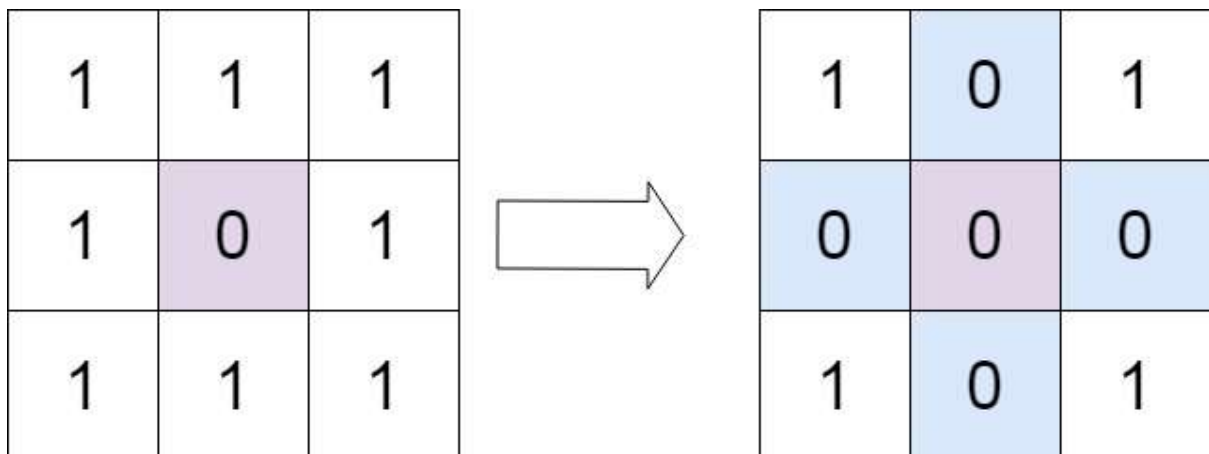
The numbers divisible by both 3 and 5, between 12 and 50 both inclusive are {15, 30, 45} and their sum is 90.

Question 20: Set Matrix Zeroes

Given an m x n integer matrix matrix, if an element is 0, set its entire row and column to 0's.

You must do it [in place](#).

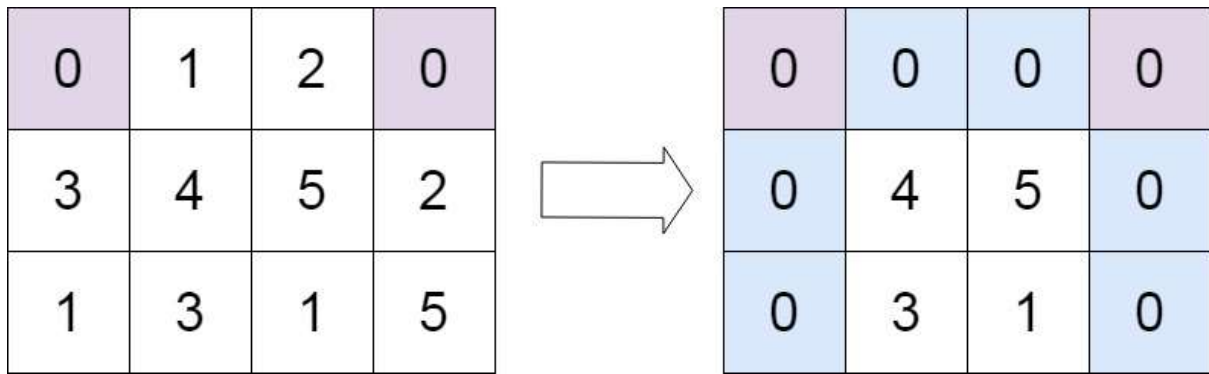
Example 1:



Input: matrix = [[1,1,1],[1,0,1],[1,1,1]]

Output: [[1,0,1],[0,0,0],[1,0,1]]

Example 2:



Input: matrix = [[0,1,2,0],[3,4,5,2],[1,3,1,5]]

Output: [[0,0,0,0],[0,4,5,0],[0,3,1,0]]

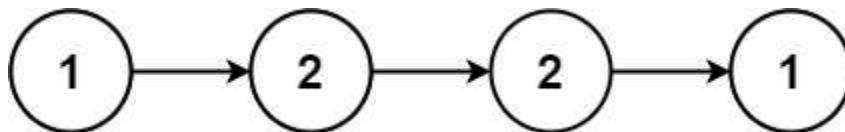
Constraints:

- $m == \text{matrix.length}$
- $n == \text{matrix}[0].\text{length}$
- $1 \leq m, n \leq 200$
- $-2^{31} \leq \text{matrix}[i][j] \leq 2^{31} - 1$

Question 21: [Palindrome Linked List](#)

Given the head of a singly linked list, return true *if it is a palindrome* or false *otherwise*.

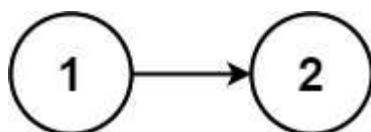
Example 1:



Input: head = [1,2,2,1]

Output: true

Example 2:



Input: head = [1,2]

Output: false

Constraints:

- The number of nodes in the list is in the range $[1, 10^5]$.
- $0 \leq \text{Node.val} \leq 9$

Question 22: Given a string *s* consisting of words and spaces, return *the length of the **last** word in the string*.

A **word** is a maximal substring consisting of non-space characters only.

Example 1:

Input: *s* = "Hello World"

Output: 5

Explanation: The last word is "World" with length 5.

Example 2:

Input: *s* = " fly me to the moon "

Output: 4

Explanation: The last word is "moon" with length 4.

Example 3:

Input: *s* = "luffy is still joyboy"

Output: 6

Explanation: The last word is "joyboy" with length 6.

Constraints:

- $1 \leq \text{s.length} \leq 10^4$
- *s* consists of only English letters and spaces ' '.
- There will be at least one word in *s*.

Solutions

Solution 1: Reverse Traversal + Two Pointers

We start traversing from the end of the string *s*, find the first character that is not a space, which is the last character of the last word, and mark the index as *i*. Then continue to

traverse forward, find the first character that is a space, which is the character before the first character of the last word, and mark it as j . Then the length of the last word is $i-j$.

The time complexity is $O(n)$, where n is the length of the string s . The space complexity is $O(1)$.