

Accenture Advanced Coding Questions

- 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

- 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 `i`th 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

- 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

- 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

- 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

- 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

$$\text{num1, num2} \geq 0$$

Example

Input

Num 1: 451

Num 2: 349

Output

2

- 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

- 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

- 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

- **Longest Substring Without Repeating Characters**

- **Description:** Given a string, find the length of the longest substring without repeating characters.
- **Input:** "abcabcbb"
- **Output:** 3 (Explanation: The answer is "abc", with the length of 3.)

- **3Sum**

- **Description:** Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that $i \neq j$, $i \neq k$, and $j \neq k$, and $nums[i] + nums[j] + nums[k] == 0$.
- **Input:** [-1,0,1,2,-1,-4]
- **Output:** [[-1,-1,2],[-1,0,1]]

- **Letter Combinations of a Phone Number**

- **Description:** Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent.
- **Input:** "23"
- **Output:** ["ad","ae","af","bd","be","bf","cd","ce","cf"]

- **Merge Intervals**

- **Description:** Given an array of intervals where $intervals[i] = [start_i, end_i]$, merge all overlapping intervals, and return an array of the non-overlapping intervals that cover all the intervals in the input.
- **Input:** [[1,3],[2,6],[8,10],[15,18]]
- **Output:** [[1,6],[8,10],[15,18]]

- **Spiral Matrix**

- **Description:** Given an $m \times n$ matrix, return all elements of the matrix in spiral order.
- **Input:** `[[1,2,3],[4,5,6],[7,8,9]]`
- **Output:** `[1,2,3,6,9,8,7,4,5]`

- **Rotate Image**

- **Description:** You are given an $n \times n$ 2D matrix representing an image, rotate the image by 90 degrees (clockwise).
- **Input:** `[[1,2,3],[4,5,6],[7,8,9]]`
- **Output:** `[[7,4,1],[8,5,2],[9,6,3]]`

- **Group Anagrams**

- **Description:** Given an array of strings `strs`, group the anagrams together. You can return the answer in any order.
- **Input:** `["eat","tea","tan","ate","nat","bat"]`
- **Output:** `[["bat"],["nat","tan"],["ate","eat","tea"]]`

- **Minimum Path Sum**

- **Description:** Given a $m \times n$ grid filled with non-negative numbers, find a path from top left to bottom right, which minimizes the sum of all numbers along its path.
- **Input:** `[[1,3,1],[1,5,1],[4,2,1]]`
- **Output:** 7 (Explanation: Because the path $1 \rightarrow 3 \rightarrow 1 \rightarrow 1 \rightarrow 1$ minimizes the sum.)

- **Unique Paths**

- **Description:** A robot is located at the top-left corner of a $m \times n$ grid. The robot can only move either down or right at any point in time. Find

the number of possible unique paths that the robot can take to reach the bottom-right corner.

- **Input:** $m = 3, n = 7$
- **Output:** 28

- **Word Search**

- **Description:** Given an $m \times n$ grid of characters board and a string word, return true if word exists in the grid.
- **Input:** board = [["A","B","C","E"], ["S","F","C","S"], ["A","D","E","E"]], word = "ABCCED"
- **Output:** true

- **Find Peak Element**

- **Description:** A peak element is an element that is strictly greater than its neighbors. Given an integer array nums, find a peak element, and return its index.
- **Input:** nums = [1,2,3,1]
- **Output:** 2 (Explanation: 3 is a peak element and your function should return the index number 2.)

Special Numbers

Problem Statement

you are given a function:

```
int DesiredArray(int Arr, int N, int k):
```

The function accepts an array 'Arr' of size 'N' and an integer 'k'.

You have to find the 'K' smallest integers that are not divisible by any of the 'N' integers and return the sum of all 'K' integers.

Note:

- Array won't contain 1.

Example:**Input:**

```
K: 4  
N: 5  
Arr: [2, 3, 4, 5, 6]
```

Output:

```
32
```

Add Alternate Nodes in Linked List**Problem Statement**

There is a singly linked list represented by the following structure:

```
struct node  
{  
    int data;  
    struct Node* next;  
};
```

Implement the following function:

```
struct Node* AddAlternateNodes(struct Node* head);
```

The function accepts a pointer to the start of the linked list , 'head' as its argument . Implement the function to modify the given list in such a way that origin added to the value of next to the next node and return the modified list.

Note:

- Return null if list is null, In case of python if list is None return None.
- Do not create new linked list , just modify the input linked list
- 1st and 2nd node values remain unchanged

Example:

Input:

head: 1 → 2 → 3 → 4 → 5 → 6 → 7

Output:

1 → 2 → 4 → 6 → 8 → 10 → 12

Distinct and Duplicate Integers**Problem Statement**

Implement the following function:

```
def AddDistinctDuplicate(a,b,c,d):
```

The function accepts four integers 'a','b','c' and 'd' as its argument . Implement the function to find the sum of distinct numbers and subtract the duplicate number and return the difference (sum of distinct number - duplicate number).

- For sum of distinct number add all number that appears only once
 - For duplicate number subtract the duplicate number only once
- Notes:** All computed values lie within integer range

Examples:**Input:**

```
a:5  
b:4  
c:4  
d:9
```

Output

10

Lettered Number**Problem Statement:**

you are required to implement the following function:

```
int LetteredNumberSum(char[] str, int len);
```

The function accepts string 'str' ('str1' in case of Python) as its argument. Implement the function which returns sum of number equivalents of each letter in the given string 'str'.

The number equivalents are as follows:

```
A = 1
B = 10
C = 100
D = 1000
E = 10000
F = 100000
G = 1000000
```

Assumption: 'str' contains upper case letters only

Note:

Number equivalent for any letter other than (A,B,C,D,E,F,G) is 0

Computed value lies within integer range

Return 0 if 'str' is null (None, in case of Python)

Example:

Input:

DCCBAA

Output:

1212

Evaluate the given expression

Problem Statement

You are given a function,

```
int EvaluateExpression(char* expr);
```

The function accepts a mathematical expression 'expr' as parameter. Implement the function to evaluate the given expression 'expr' and return the evaluated value.

Assumption:

- You can assume there is no space in between any of the characters of expression 'expr'.
- Expression 'expr' contains only digits and operators (+, -, * and /).

Note:

- Every operation should be integer based e.g.: 5/2 should give 2 not 2.5
- Consider the precedence of operators while evaluating the expression, precedence of ('/' or '*') > precedence of ('+' or '-').
- if 'expr' has multiple operators of same precedence then evaluate them from left to right.

Example:**Input:**

```
expr : 2+3+5*4/2
```

Output:

```
15
```

Maximum Element And Its Index**Problem Statement:**

You are given a function,

```
Void MaxInArray(int arr[],int length);
```

The function accepts an integer array 'arr' of size 'length' as its argument. Implement the function to find the maximum element of the array and print the

element and its index to the standard output (STDOUT) . The maximum element and its index should be printed in separate lines.

Notes:

- Array index start with 0.
- Maximum element and its index should be separated by a line in the output.
- Assume there is only 1 maximum element in the array.
- Print exactly what is asked, do not print any additional greeting messages

Exampe 1:

```
23 45 82 27 66 12 78 13 71 86
```

Output:

```
86
9
```

Fine Number**Problem Statement**

You are given a function:

```
int FineNumber(int* a, int* b, int n, int m);
```

The function accepts two arrays 'a' and 'b' of size 'n' and 'm' respectively. Implement the function to compute a fine number and return the same.

A fine number is the greatest number that can be obtained by taking the difference of two numbers such that one of the two numbers is taken from array 'a' and the other is taken from array 'b'.

Example:**Input:**

```
n: 5
m: 6
```

```
a: 1 2 3 4 5
b: 10 12 34 2 4 89
```

Output:

```
88
```

Mearge and Rearrange**Problem Statement**

You are given a function,

```
char* MergeStrings(char* str1, char* str2);
```

The function accepts strings 'str1' and 'str2' as its arguments, Implement the function to generate a string by iterating through each character of given string.

- For $i = 0$, on comparing charecters at index 0 of input strings, smaller character is placed at index 0 and larger character is placed at index $n-1$.
- For $i = 1$, on comparing charecters at index 1 of input strings, smaller character is placed at index 1 and larger character is placed at index $n-2$.
- For $i = k$, on comparing charecters at index k of input strings, smaller character is placed at index k and larger character is placed at index $n-k-1$. where $k < n$ and n is the length of output string (Length of string1 + Length of string2).

Assumption: String contain lower case characters only.

Note:

- Character 'x' is smaller than 'y' since it occurs prior in alphabetical series.
- Return null if both the strings are null.
- Return other string if one of the string is null.
- Null refers to None in case of Python.
- If length of strings is not same, then rest of the characters are added on their original positions.

Example:

Input:

```
str1: are
str2: denim
```

Output:

```
aeeimnrd
```

Sum of digits**Problem Statement**

You are required to implement the following function:

```
int DifferenceSumOfDigits(int* arr, int n);
```

The function accepts an array 'arr' of 'n' positive integers as its argument. Let's suppose:

$$f(x) = \text{Sum of digits of an integer}$$

You are required to calculate the value of the following:

```
F1= [f(arr[0]) + f(arr[1]) + f(arr[2]) + .....+ f(arr
[n-1])] %10
F2 = [(arr[0] + arr[1] + arr[2] + .....+ arr[n-1])] % 1
0
F = F1 - F2
```

and return the value of F.

Note: $n > 0$

Example:**Input:**

```
arr: 11 14 16 10 9 8 24 5 4 3
```



```
n: 10
```

Output:

```
-4
```

Small Number Problem

Problem Statement:

Implement the following function:

```
int * NextSmallerNumber(int a[], int m);
```

The function accepts an integer array 'a' of size m. Replace each number of array with nearest smaller number on its right in the array.

Assumption: All integers are > 0.

Note:

- Return null if array is null.
- Null refers to None in case of Python.
- Replace the number with '-1', if no smaller number is present on its right.

Example:

Input:

```
a: 3 2 11 7 6 5 6 1
```

Output:

```
2 1 7 6 5 1 1 -1
```

Marching People

Problem Statement:

Infinite number of people are crossing a 2-D plane. They march in such a way that each integral x coordinate will have exactly one person who moves along it in positive y direction, starting from $(x, 0)$.

You have to implement the following functions:

```
int MaximumBarrier(int n, int** barrier);
```

The function takes an integer matrix 'barrier' having 'n' rows and '3' columns where n denotes the number of barriers. The ith barrier is defined by (x_i, y_i, d_i) , which means that the barrier is blocking all the people who want to pass through points lying on line segment connecting (x_i, y_i) and $(x_i + d_i, y_i)$. Once a person encounters a barrier, he stops moving.

Given all the barriers, your task is to find the total number of people who will be blocked at some point in their march.

Assumption:

- $n > 0$
- Length of barrier (d) > 0

Notes:

- Overlapping of barriers is possible.
- Do not use extra memory.

Example:

Input:

```
n:2
      x    y    d
Barrier 1 :  2    3    3
Barrier 2 :  4    6    4
```

Output:

```
7
```

Ball and Box Problem

Problem Statement

Implement the following function:

```
int NumberOfBalls(int arr[], int n);
```

The function accepts a non-negative integer array 'arr' of size 'n' as its argument. Every Kth element in the array is the number of balls in the Kth row of a box. Every Kth row of the box needs $(K+1)^2$ balls, where $0 \leq K \leq (n-1)$. Implement the function to find the number of balls required to complete each row of the box and return the total number of balls required.

Assumption: $\text{arr}[k] \leq (k+1)^2$

Note:

1. Return -1 if the array is null (or None in the case of Python).
2. Array indexing starts from 0.

Example:

Input:

```
Arr: 1 2 7 13
```

Output:

```
7
```

Decrypt the string

Problem Statement

Implement the following function:

```
char* Decrypt(char str[], int n);
```

The function accepts a string 'str' of size 'n' as its argument. Implement the function to decrypt the given 'str' in such a way that each character of the string is replaced as follows:

('a' → 'z', 'b' → 'y', 'c' → 'x', 'd' → 'w', 'e' → 'v', 'f' → 'u', 'g' → 't', 'h' → 's', 'i' → 'r', 'j' → 'q', 'k' → 'p', 'l' → 'o', 'm' → 'n', 'n' → 'm', 'o' → 'l', 'p' → 'k', 'q' → 'j', 'r' → 'i', 's' → 'h', 't' → 'g', 'u' → 'f', 'v' → 'e', 'w' → 'd', 'x' → 'c', 'y' → 'b', 'z' → 'a').

Return the decrypted string.

Assumption:

- The string contains only lowercase alphabets.

Note:

- Return NULL if the string is null (or None in the case of Python).

Example:

Input:

Str: vmxibkgrlm

Output:

Encryption