Accenture Coding Questions

1. **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

   Explanation:
   The alphabets in str when expanded becomes “1 XOR 0 XOR 1 XOR 1 AND 0 OR 1”, the result of the expression becomes 1, hence 1 is returned.

Python Code:

```
def OperationsBinaryString(str):

    a=int(str[0])

    i=1

    while i<len(str):

        if str[i]=='A':

            a&=int(str[i+1])
        elif str[i]=='B':
            a|=int(str[i+1])
        else:
            a^=int(str[i+1])
        i+=2
    return a
str=input()
print(OperationsBinaryString(str))
```

2.

Problem Description:

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 <= 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 * 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.

Python Code:

```
def calculate(r,unit,arr,n):
    if n==0:
        return -1
    totalFoodRequired=r*unit
    foodTillNow=0
    house=0
    for house in range(n):
        foodTillNow+=arr[house]
        if foodTillNow >= totalFoodRequired:
            break
    if totalFoodRequired > foodTillNow:
        return 0
    return house+1
r = int(input())
unit = int(input())
n = int(input())
arr = list(map(int,input().split()))
print(calculate(r,unit,arr,n))
```

3. **Problem Statement**:

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.

Python Code:

```
def findCount(n, arr, num, diff):
    count=0
    for i in range(n):
        if(abs(arr[i]-num)<=diff):
            count+=1
    if count:
        return count
    return 0
n=int(input())
arr=list(map(int,input().split()))
num=int(input())
diff=int(input())
print(findCount(n, arr, num, diff))
```

4.

Problem Statement

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 : 6
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
enter element at 5 index:  8
Sorted even array : 1 3 7 9
Sorted odd array : 4 8
Sum of second largest numbers from both sorted arrays = 11
```

Python Code:

```
earr=[]
oarr=[]
i=0
n=int(input("Enter array size"))
while(i<n):
    x=int(input("Enter the element"))
    if(x%2==0):
        earr.append(x)
    else:
        oarr.append(x)
    i+=1
earr.sort()
oarr.sort()
print(earr[-2]+oarr[-2])
```

5.

Problem Statement:

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 <= 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.

Python Code:

```
m=int(input())

n=int(input())

sum=0

while(m<=n):

    if(m%3==0 and m%5==0):

        sum=sum+m

    m=m+1

print(sum)
```