Result & Analysis

Student: AMBER JAIN Test: 2023_Hour of Code_Day ... Course: 2023 Batch_Hour of C...

Attempt 1

IP Address: 103.101.103.37 **Tab switches:** 4 **OS used:** Windows **Browser used:** Chrome

Test Duration: 00:12:03 **Test Start Time:** May 2, 2022 | 09:54 AM

Test Submit Time: May 2, 2022 | 10:07 AM

Overall score

0

/ 20

Rank: NA

Topper score: 20.00 / 20

Average score: 9.70 / 20

Least score: 0.00 / 20

Coding

0

/ 20

Rank: NA

Topper score: 20.00 / 20

Average score: 13.34 / 20

Least score: 0.00 / 20

Overall Question Status

0

/ 2

Total Questions: 2

Questions Attempted: 0

Questions Correct: 0

Question Wrong: 0

Partially Correct: 0

Question Not Viewed: 0

Coding - Question Status

0

/ 2

Total Questions: 2

Questions Attempted: 0

Questions Correct: 0

Question Wrong: 0

Partially Correct: 0

Question Not Viewed: 0

Topic wise Analysis Coding

Question No: 1 **Single File Programming Question** **Report Error**

Count occurrences of a digit

Problem statement

Implement the following function:

int CountDigiOoccurrences(int l, int u, int x);

The function accepts 3 positive integers ‘l’, 'u' and 'x' as its arguments. You are required to calculate the number of occurrences of a digit 'x' in the digits of numbers lying in the range between ‘l’ and 'u' both inclusive, and return the same.

Note

l < u

0 < x < 9

Example

Input

l: 2

u: 13

x: 3

Output

2

Explanation

The number of occurrences of digit 3 in the digits of numbers lying in the range [2, 13] both inclusive is 2, ie (3, 13), hence 2 is returned.

Input format

The input accepts the three integers separated by a new line as given in the question.

Output format

The output is a single integer type.

Sample testcases

Input 1	Output 1
2 13 3	2

Input 2	Output 2
1 100 9	20

C (17)

```
// You are using GCC

```

Status: Skipped **Mark obtained:** 0/10 **Hints used:** 0 **Times compiled:** 5

Times submitted: 0 **Level:** Medium **Question type:** Single File Programming

Subject: Programming **Subject:** C++ Programming **Subject:** Branching and Looping

□ Show testcase scores ☑ Show solution

Solution 1

C (17)

```
#include<stdio.h>

int countDigitOccurrences(int l, int u, int x)
{
    int rem, count = 0;
    for(int i = l; i <= u; i++)
    {
        int temp = i;
        while(temp != 0 )
        {
            if(temp % 10 == x)
                count++;
            temp = temp / 10;
        }
    }
    return count;
}
int main()
{
   int l, u, x;
   scanf("%d %d %d", &l, &u, &x);
   printf("%d", countDigitOccurrences(l, u, x));
   return 0;
}
```

Solution 2

C++ (17)

```
#include<iostream>
using namespace std;

int countDigitOccurrences(int l, int u, int x)
{
    int rem, count = 0;
    for(int i = l; i <= u; i++)
    {
        int temp = i;
        while(temp != 0 )
        {
            if(temp % 10 == x)
                count++;
            temp = temp / 10;
        }
    }
    return count;
}
int main()
{
   int l, u, x;
   cin >> l >> u >> x;
   cout << countDigitOccurrences(l, u, x);
   return 0;
```

Solution 3

Java (11)

```
import java.util.*;
class Main
{
    static int countDigitOccurrences(int l, int u, int x)
    {
        int rem, count = 0;
        for(int i = l; i <= u; i++)
        {
            int temp = i;
            while(temp != 0 )
            {
                if(temp % 10 == x)
                    count++;
                temp = temp / 10;
            }
        }
        return count;
    }

    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        int l = sc.nextInt();
        int u = sc.nextInt();
```

Solution 4

Python (3.8)

```
def countDigitOccurrences(l, u, x):
    count = 0
    for i in range(l, u+1):
        temp = i
        while temp != 0:
            if temp % 10 == x:
                count = count + 1
            temp = temp // 10
    return count

l = int(input())
u = int(input())
x = int(input())

print(countDigitOccurrences(l, u, x))


```

Question No: 2 Single File Programming Question

Report Error

Sum of numbers with set bit

You are given a function,

int SumofSetBitNumbers(int n, int set);

The function accepts two integers ‘n’ and ‘set’ as its argument where ‘n’ is the number of bits and ‘set’ is the number of set bits. Implement the function to find the sum of all numbers possible from ‘n’ bits having the count of ‘set’ bits equal to set.

Note

n > 0

set >= 0

Example

Input

n: 3

set: 2

Output

14

Explanation

All possible 3 bit numbers are 0, 1, 2, 3, 4, 5, 6, 7 with binary representation 000, 001, 010, 011, 100, 101, 110, 111. Numbers with 2 bits set are 3, 5 and 6 summation of which is equal to 14.

Input format

The first line of integer input represents the number of bits
The second line of integer input represents the number of set bits in the binary representation.

Output format

Sum of all possible decimal values of n bits with the exact number of set bits.

Sample testcases

Input 1	Output 1
3 2	14

Input 2	Output 2
2 1	3

C (17)

```
// You are using GCC

```

Status: Skipped **Mark obtained:** 0/10 **Hints used:** 0 **Times compiled:** 0

Times submitted: 0 **Level:** Medium **Question type:** Single File Programming

Subject: Programming **Subject:** C++ Programming **Subject:** Branching and Looping

☐ Show testcase scores ☑ Show solution

Solution 1

C (17)

```
#include<stdio.h>
#include<math.h>

int SummationOfSetBit(int n, int set)
{
    int i, sum = 0, m, count;
    int range = pow(2, n);
    for(i = 0; i < range; i++)
    {
        count = 0;
        m = i;
            while(m)
            {
                m &= (m - 1);
                count++;
            }
            if(count == set)
                sum = sum + i;
    }
    return sum;
}
int main( )
{
    int n, set;
```

Solution 2

C++ (17)

```
#include<iostream>
#include<cmath>
using namespace std;
int SummationOfSetBit(int n, int set)
{
    int i, sum = 0, m, count;
    int range = pow(2, n);
    for(i = 0; i < range; i++)
    {
        count = 0;
        m = i;
            while(m)
            {
                m &= (m - 1);
                count++;
            }
            if(count == set)
                sum = sum + i;
    }
    return sum;
}
int main( )
{
    int n, set;
```

Solution 3

Java (11)

```
import java.util.*;
import java.lang.*;
class Main
{
    static int sumofSetBits(int n, int set)
    {
        int i, sum = 0, m, count;
        int range =(int) Math.pow(2, n);
        for(i = 0; i < range; i++)
        {
            count = 0;
            m = i;
            while(m != 0)
            {
                m &= (m - 1);
                count++;
            }
            if(count == set)
                sum = sum + i;
        }
        return sum;
    }
    public static void main(String args[])
    {
```

Solution 4

Python (3.8)

```
def sumofSetBits(n, setbit):
    sum = 0
    r = 2**n
    for i in range(0, r):
        count = 0
        m = i
        while(m != 0):
            m = m & (m - 1)
            count = count + 1
        if(count == setbit):
            sum = sum + i;
    return sum

n = int(input())
setbit = int(input())
print(sumofSetBits(n, setbit))

```