

Question 1: Rat Count House

(Asked in Accenture OnCampus 10 Aug 2022, Slot 1)

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 $0 \leq i$

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, amount of food in 1st 4 houses is sufficient for all the rats. Thus, output is 4.

C++

C

Python

Java

Run

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int calculate (int r, int unit, int arr[], int n)
```

```
{
```

```

if (n == 0)
    return -1;

int totalFoodRequired = r * unit;
int foodTillNow = 0;
int house = 0;
for (house = 0; house < n; ++house)
{
    foodTillNow += arr[house];
    if (foodTillNow >= totalFoodRequired)
    {
        break;
    }
}
if (totalFoodRequired > foodTillNow)
    return 0;
return house + 1;
}

int main ()
{
    int r;
    cin >> r;
    int unit;
    cin >> unit;
    int n;
    cin >> n;
    int arr[n];
    for (int i = 0; i < n; ++i)
    {
        cin >> arr[i];
    }
    cout << calculate (r, unit, arr, n);

```

```
    return 0;
}
```

Question 2:

(Asked in Accenture OnCampus 10 Aug 2022, Slot 2)

Problem Description :

The Binary number system only uses two digits, 0 and 1 and 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 eparated 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 opearation 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:

str: 1C0C1C1A0B1

Output:

1

Explanation:

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

Sample Input:

0C1A1B1C1C1B0A0

Output:

0

C++

C

python

Java

Run

```
#include<bits/stdc++.h>

using namespace std;

int OperationsBinaryString (char *str)
{
    if (str == NULL)
        return -1;
    int i = 1;
    int a = *str - '0';
    str++;
    while (*str != '\0')
    {
        char p = *str;
        str++;
        if (p == 'A')
            a &= (*str - '0');
        else if (p == 'B')
            a |= (*str - '0');
        else
            a ^= (*str - '0');
        str++;
    }
    return a;
}

int main ()
{
    string s;
    getline (cin, s);
    int len = s.size ();
    char *str = &s[0];
    cout << OperationsBinaryString (str);
```

```
}
```

Question 3: Password Checker

(Asked in Accenture OnCampus 10 Aug 2022, Slot 3)

You are given a function.

```
int CheckPassword(char str[], int n);
```

The function accepts string str of size n as an argument. Implement the function which returns 1 if given string str is valid password else 0.

str is a valid password if it satisfies the below conditions.

- – At least 4 characters
- – At least one numeric digit
- – At Least one Capital Letter
- – Must not have space or slash (/)
- – Starting character must not be a number

Assumption:

Input string will not be empty.

Example:

Input 1:

aA1_67

Input 2:

a987 abC012

Output 1:

1

Output 2:

0

C++

C

python

Java

Run

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int CheckPassword (char str[], int n)
```

```
{
```

```
    //At least 4 characters
```

```
    if (n < 4)
```

```

    return 0;

//Starting character must not be a number
if (str[0] - '0' >= 0 && str[0] - '0' <= 9)

    return 0;

int a = 0, cap = 0, nu = 0;
while (a < n)
{
    //Must not have space or slash (/)
    if (str[a] == ' ' || str[a] == '/')

        return 0;

    //counting capital letters
    if (str[a] >= 65 && str[a] <= 90)
    {
        cap++;
    }

    //counting numeric digit
    else if (str[a] - '0' >= 0 && str[a] - '0' <= 9)

        nu++;

    //incrementing for while loop
    a++;
}

// returns 1 if there are > 0 numeric digits and capital letters
return cap > 0 && nu > 0;
}

int main ()
{
    string s;
    getline (cin, s);
    int len = s.size ();
    char *c = &s[0];
    cout << CheckPassword (c, len);

```

}

Question 4:

(Asked in Accenture OnCampus 11 Aug 2022, Slot 1)

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.

C++

C

python

Java

Run

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int findCount (int n, int arr[], int num, int diff)
```

```
{
```

```
    int count = 0;
```

```
    for (int i = 0; i < n; ++i)
```

```
    {
```

```
        if (abs (arr[i] - num) <= diff)
```

```
        {
```

```
            count++;
```

```
        }
```

```
    }
```

```
    return count > 0 ? count : -1;
```

```
}
```

```
int main ()
```

```
{
```

```
    int n;
```

```
    cin >> n;
```

```
    int arr[n];
```

```
    for (int i = 0; i < n; ++i)
```

```
    {
```

```
        cin >> arr[i];
```

```
    }
```

```
    int num;
```

```
    cin >> num;
```

```
    int diff;
```

```
    cin >> diff;
```



```
cout << findCount (n, arr, num, diff);  
}
```

Question 5 :

(Asked in Accenture OnCampus 11 Aug 2022, Slot 2)

Implement the following Function

```
def differenceofSum(n, m)
```

The function accepts two integers n, m as arguments Find the sum of all numbers in range from 1 to m(both inclusive) that are not divisible by n. Return difference between sum of integers not divisible by n with sum of numbers divisible by n.

Assumption:

- $n > 0$ and $m > 0$
- Sum lies between integral range

Example

Input

n:4

m:20

Output

90

Explanation

- Sum of numbers divisible by 4 are $4 + 8 + 12 + 16 + 20 = 60$
- Sum of numbers not divisible by 4 are $1 + 2 + 3 + 5 + 6 + 7 + 9 + 10 + 11 + 13 + 14 + 15 + 17 + 18 + 19 = 150$
- Difference $150 - 60 = 90$

Sample Input

n:3

m:10

Sample Output

19

C++

C

Python

Java

Run

```
#include <iostream>
```

```
using namespace std;
```

```
int differenceofSum (int n, int m)
```

```
{
```

```
    int i, sum1 = 0, sum2 = 0;
```

```
    for (i = 1; i <= m; i++) { if (i % n == 0) { sum1 = sum1 + i; } else { sum2 = sum2 + i; } } if (sum2 > sum1)
```

```
        return sum2 - sum1;
```

```
    else
```

```
        return sum1 - sum2;
```

```
}
```

```
int main ()
```

```
{
```

```
    int n, m;
```

```
    int result;
```

```
    cin >> n;
```

```
    cin >> m;
```

```
    result = differenceofSum (n, m);
```

```
    cout << result;
```

```
    return 0;
```

```
}
```