Rajalakshmi Engineering College

Name: Priyan Kumar S
Email: 240701401@rajalakshmi.edu.in
Roll no: 240701401
Phone: 7305916381
Branch: REC
Department: CSE - Section 8
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Mandy is a software engineer working on a program to analyze two integers based on specific conditions using a logical operator. She needs to determine if both integers are odd or if at least one of them is divisible by 7.

Depending on the result, she wants to print different messages.

If the condition is met, the program should identify and print the first number that is divisible by 7 or indicate that both numbers are odd.If the condition is not met, the program should print a message indicating the condition was not met, along with the input numbers.

Input Format

The first line of input consists of an integer representing the first input number.

The second line consists of an integer representing the second input number.

Output Format

The output displays "Condition met: " followed by an integer representing the first number divisible by 7, or prints "Both numbers are odd" if the two inputs are odd.

If the condition is not met, it displays "Conditions not met: " followed by the two input integers, separated by a space.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 7
14
Output: Condition met: 7

Answer

```
import java.util.*;
import java.io.*;
class find{
   public static void main(String[] args){
      Scanner S=new Scanner(System.in);
      int n1=S.nextInt();
      int n2=S.nextInt();
      if(n1%7==0){
            System.out.println("Condition met: "+n1);
       }
      else if(n2%7==0){
          System.out.println("Condition met: "+n2);
      }
      else if((n1%2!=0 && n2%2!=0)){
         System.out.println("Condition met: Both numbers are odd");
       }

      else if((n1%2==0 && n2%2==0) && (n1%7!=0 && n1%7!=0))
      {
         System.out.println("Conditions not met: "+n1+" "+n2);
```

```
        }
        else if(n1%2!=0 || n2%2!=0)
        {
            if(n1%7==0){
                System.out.println("Condition met: "+n1);
            }else if(n2%7==0){
                System.out.println("Condition met: "+n2);
            }else
                System.out.println("Conditions not met: "+n1+" "+n2);
        }
    }
}
```

Status : Correct **Marks : 10/10**

2. Problem Statement

In a logistics company, each delivery pack contains a specific number of items, and the priority customer receives double the amount. Write a program to determine the total number of delivery packs required for the operation, considering the number of items per pack and the number of customers given as input by the user.

Example

Input:

Number of items per pack = 96

Number of customers = 8

Output:

10

Explanation:

Given the number of items per pack = 96 and the number of customers = 8, the calculations are as follows:

Total number of items needed = number of items per pack * number of customers = 96 * 8 = 768.Priority customer's share = double the amount of items per pack = 2 * 96 = 192.Total items with the priority customer = total

items needed + priority share = 768 + 192 = 960.Number of packs needed = (960 + 96 − 1)/ 96 = 10.98Since we cannot have a fraction of a pack, the output is 10.

Input Format

The input consists of two space-separated integers N and C, representing the number of items per pack and the number of customers.

Output Format

The output displays an integer, representing the total number of delivery packs required for the operation.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1 1

Output: 3

Answer

```
import java.util.*;
import java.io.*;
class find{
   public static void main(String[] args){
      Scanner S=new Scanner(System.in);
      int n1=S.nextInt();
      int n2=S.nextInt();
      int tot,res;
      tot=(n1*n2)+(2*n1);
      res=(tot+n1-1)/n1;
      System.out.println(res);

   }
}
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

In the faraway land of Arithmetica, there exists an ancient calculator that can only perform bitwise operations. The calculator is locked with a secret code that only works when the number is modified using a special operation called right shifting.

The ruler of Arithmetica, King Thales, needs your help to unlock the calculator. The lock on the calculator is encoded with a number, and the calculator will only open if you apply a right shift by 2 on the number. Your task is to help King Thales determine the magic number that will unlock the ancient calculator.

Input Format

The first line of input represents an integer.

Output Format

The output should display the right-shifted value by 2 bits.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 16

Output: 4

Answer

```
import java.util.*;
import java.io.*;
class find{
   public static void main(String[] args){
      Scanner S=new Scanner(System.in);
      int n=S.nextInt();
      int mask;
      mask=(n>>2);
      System.out.println(mask);
   }
}
```

Status : Correct ***Marks : 10/10***

4. Problem Statement:

Gilbert is tasked with writing a program that checks whether a given integer is an odd number. An odd number is one that cannot be exactly divided by 2. The program should take an integer as input and determine if it is an odd number or not. The task is to implement the logic to check if the provided integer is odd and return the result.

Input Format

The first line of the input contains an integer, "input".

Output Format

The output should display a boolean value, "result," which should be set to true if the input integer is an odd number and false if it is even.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 0

Output: Is the integer odd? false

Answer

```
import java.util.*;
import java.io.*;
class find{
   public static void main(String[] args){
      Scanner S= new Scanner(System.in);
      int n=S.nextInt();
      if(n%2==0 || n==0){
         System.out.println("Is the integer odd? false");
      }else
         System.out.println("Is the integer odd? true");
   }
}
```

Status : Correct ***Marks : 10/10***