Rajalakshmi Engineering College

Name: Shreenidhi T
Email: 240701503@rajalakshmi.edu.in
Roll no: 240701503
Phone: 9150942326
Branch: REC
Department: CSE - Section 10
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

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.Scanner;

class DeliveryPacksCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int N = scanner.nextInt();
        int C = scanner.nextInt();
        int totalItems = N * C;
        int priorityShare = 2 * N;
        int totalWithPriority = totalItems + priorityShare;
    int numberOfPacks = (totalWithPriority + N - 1) / N;
    System.out.println(numberOfPacks);
  }
}
```

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

2. 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.Scanner;

class OddNumberChecker {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int input = scanner.nextInt();
    boolean result = (input % 2 != 0);
    System.out.println("Is the integer odd? " + result);
  }
}
```

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

3. Problem Statement:

Tom is tasked with writing a program that determines whether a given integer is the square of another integer. A perfect square is a number that can be expressed as the square of an integer. The program should take an integer as input and determine if it is a perfect square or not.

The task is to implement the logic to check if the provided integer is the square of an integer and return the result.

Input Format

The first line of the input contains an integer, "input", where |input| represents the absolute value of the integer.

Output Format

The output should display a boolean value, "result," which should be set to true if the input is a perfect square (the square of an integer), and false if it is not.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 16
Output: Is the integer a perfect square? true

Answer

```
import java.util.Scanner;

class PerfectSquareChecker {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int input = scanner.nextInt();
```

```
        boolean result = false;

        if (input >= 0) {
            int sqrt = (int) Math.sqrt(input);
            result = (sqrt * sqrt == input);
        }
        System.out.println("Is the integer a perfect square? " + result);
    }
}
```

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

4. 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.Scanner;
class NumberAnalyzer {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      int num1 = scanner.nextInt();
      int num2 = scanner.nextInt();
      boolean bothOdd = (num1 % 2 != 0) && (num2 % 2 != 0);
      boolean divisibleBy7 = (num1 % 7 == 0) || (num2 % 7 == 0);

      if (bothOdd || divisibleBy7) {
         System.out.print("Condition met: ");
         if (divisibleBy7) {
            if (num1 % 7 == 0) {
               System.out.println(num1);
            } else {
               System.out.println(num2);
            }
         } else {
            System.out.println("Both numbers are odd");
         }
      } else {
         System.out.println("Conditions not met: " + num1 + " " + num2);
      }
   }
}
```

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