

# Rajalakshmi Engineering College

Name: SANTHOSH S  
Email: 241801251@rajalakshmi.edu.in  
Roll no: 241801251  
Phone: 9790911586  
Branch: REC  
Department: AI & DS - Section 3  
Batch: 2028  
Degree: B.E - AI & DS

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:

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;
import java.lang.Math.*;
class Main
{
    public static void main (String [] args)
    {
        Scanner sc = new Scanner (System.in);
        int num = sc.nextInt();
        boolean res = false;
        if (num>0)
        {
            int sq =(int) Math.sqrt(num);
            res = (sq *sq == num);
        }
        System.out.println("Is the integer a perfect square? "+res);
    }
}
```

**Status :** Correct

**Marks :** 10/10

## **2. 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.Scanner;
public class Main
{
    public static void main (String [] args)
    {
        Scanner sc = new Scanner (System.in);
        int num = sc.nextInt();
        int res = num>>2;
        System.out.println(res);
    }
}
```

**Status :** Correct

**Marks :** 10/10

## **3. 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 =  $(960 + 96 - 1) / 96 = 10.98$ . Since 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**

```
// You are using Java
import java.util.Scanner;
public class Main
{
    public static void main (String [] args)
    {
        Scanner sc = new Scanner (System.in);
        int N = sc.nextInt();
        int C = sc.nextInt();
        int tt = N*C;
        int prior = 2*N;
        int pc = tt+prior;
        int packs =(int) (pc+N-1)/N;
        System.out.println(packs);
    }
}
```

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement

Mandy is working on a cybersecurity project that involves basic encryption techniques. She wants to write a program that takes an integer number and performs a bitwise XOR operation to flip all the bits.

Help Mandy in this encryption using bitwise operations.

**Input Format**

The input consists of an integer N, representing the number to be flipped.

**Output Format**

The output displays "Result: " followed by an integer representing the result of the bitwise XOR operation to flip all the bits.

Refer to the sample output for formatting specifications.

**Sample Test Case**

Input: 0

Output: Result: 255

**Answer**

```
import java.util.Scanner;
public class Main
{
    public static void main(String [] args)
    {
        Scanner read = new Scanner (System.in);
        int a = read.nextInt();
        int result = a^255;
        System.out.println("Result:"+result);
    }
}
```

**Status :** Correct

**Marks : 10/10**