**PROGRAM – LEAP YEAR**

**Q1:** Write a LeapYear program that takes a year as input and outputs the Year is a Leap Year or not a Leap Year.

**Hint =>** a. The LeapYear program only works for year >= 1582, corresponding to a year in the Gregorian calendar. So ensure to check for the same.

b. Further, the Leap Year is a Year divisible by 4 and not 100 unless it is divisible by 400. E.g. 1800 is not a Leap Year and 2000 is a Leap Year.

c. Write code having multiple ***if else*** statements based on conditions provided above and a second part having only one if statement and multiple logical

**CODE:**

import java.util.\*;

public class leapYear {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

System.out.println("ENTER A YEAR TO CHECK IT'S TYPE- LEAP YEAR OR NON-LEAP YEAR :");

int year = sc.nextInt();

if (year % 4 == 0) {

if (year % 100 != 0) {

System.out.println("ENTERED YEAR IS LEAP YEAR");

} else if (year % 400 == 0) {

System.out.println("ENTERED YEAR IS LEAP YEAR");

} else {

System.out.println("ENTERED YEAR IS NON-LEAP YEAR");

}

} else {

System.out.println("ENTERED YEAR IS NON-LEAP YEAR");

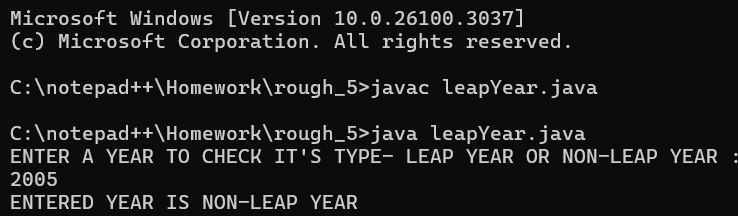
}

sc.close();

}

}

**OUTPUT:**

****

**PROGRAM – LEAP YEAR WITH LOGICAL OPERATORS**

**Q2:** Rewrite program 1 to determine Leap Year with single if condition using logical and ***&&*** and or ***||*** operators

**CODE:**

import java.util.Scanner;

public class leapYearLogicalOparetors {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

System.out.println("ENTER A YEAR TO CHECK IT'S TYPE- LEAP YEAR OR NON-LEAP YEAR :");

int year = sc.nextInt();

boolean x = (year % 4) == 0;

boolean y = (year % 100) != 0;

boolean z = ((year % 100) == 0 && (year % 400) == 0);

String r = (x && (y || z) ? "ENTERED YEAR IS LEAP YEAR" : "ENTERED YEAR IS NON-LEAP YEAR");

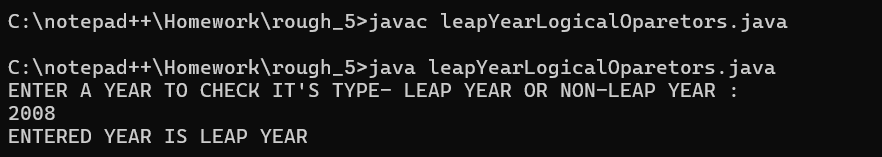
System.out.println(r);

sc.close();

}

}

**OUTPUT:**

****

**PROGRAM – AVERAGE MARK AND GRADE**

**Q3:**  Write a program to input marks and 3 subjects’ physics, chemistry and maths. Compute the percentage and then calculate the grade as per the following guidelines



**Hint =>** a. Ensure the Output clearly shows the Average Mark as well as the Grade and Remarks

**CODE:**

import java.util.\*;

public class AvgmarksGradeRemarks {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

System.out.println("ENTER YOUR MARKS IN PHYSICS : ");

int p = sc.nextInt();

System.out.println("ENTER YOUR MARKS IN CHEMISTRY : ");

int c = sc.nextInt();

System.out.println("ENTER YOUR MARKS IN MATHS : ");

int m = sc.nextInt();

int total = p + c + m;

double percentage = ((double) total / 300) \* 100;

char grade;

String remarks;

if (percentage >= 80) {

grade = 'A';

remarks = "Level 4, above agency-normalized standards";

} else if (percentage >= 70 && percentage <= 79) {

grade = 'B';

remarks = "Level 3, at agency-normalized standards";

} else if (percentage >= 60 && percentage <= 69) {

grade = 'C';

remarks = "Level 2, below but approaching agency-normalized standards";

} else if (percentage >= 50 && percentage <= 59) {

grade = 'D';

remarks = "Level 1, well below agency-normalized standards";

} else if (percentage >= 40 && percentage <= 49) {

grade = 'E';

remarks = "Level 1, too below agency-normalized standards";

} else {

grade = 'R';

remarks = "REMEDIAL STANDARDS";

}

int avg = total / 3;

System.out.println("YOUR AVERAGE MARK IS : " + avg + "\n,GRADE IS : " + grade + "\nand REMARKS IS : " + remarks);

} catch (InputMismatchException e) {

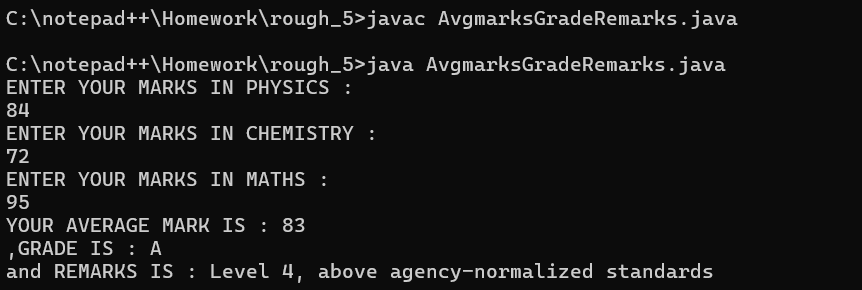
System.out.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – PRIME NUMBER**

**Q4:** Write a Program to check if the given number is a prime number or not

**Hint =>** a. A number that can be divided exactly only by itself and 1 are Prime Numbers, b. Prime Numbers checks are done for number greater than 1 c. Loop through all the numbers from 2 to the user input number and check if the reminder is zero. If the reminder is zero break out from the loop as the number is divisible by some other number and is not a prime number. d. Use isPrime boolean variable to store the result

**CODE:**

import java.util.\*;

public class primeNo {

public static void prime(int n) {

if (n <= 1) {

System.out.println(n + " IS NEITHER A PRIME NOR A COMPOSITE NUMBER.\nBECAUSE " + n + " IS NOT DIVISIBLE BY TWO NUMBERS.\n");

} else {

if (n == 2 || n == 3) {

System.out.print("THE ENTERED NUMBER IS PRIME.");

} else {

String isprime = " ";

for (int i = 2; i <= Math.sqrt(n); i++) {

isprime = ((n % i) == 0) ? "ENTERED NUMBER IS NON-PRIME." : "ENTERED NUMBER IS PRIME.";

}

System.out.print(isprime);

}

}

}

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

System.out.println(": LETS CHECK A NUMBER IS EITHER PRIME OR NOT :");

System.out.print("ENTER THE NUMBER : ");

int n = sc.nextInt();

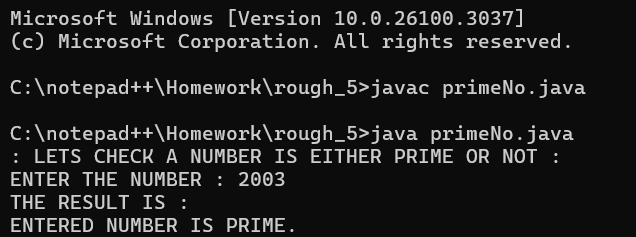
System.out.print("THE RESULT IS : \n");

prime(n);

}

}

**OUTPUT:**

****

**PROGRAM – ARMSTRONG NUMBER**

**Q5:** Create a program to check if a number is armstrong or not. Use the hints to show the steps clearly in the code.

**Hint =>** a. Armstrong Number is a number whose Sum of cubes of each digit results in the original number as in for e.g. 153 = 1^3 + 5^3 + 3^3 b. Get an integer input and store it in the number variable and define sum variable, initialize it to zero and originalNumber variable and assign it to input number variable c. Use the ***while*** loop till the originalNumber is not equal to zero d. In the ***while*** loop find the reminder number by using the modulus operator as in ***number % 10*** . Find the cube of the number and add it to the ***sum*** variable e. Again in while loop find the quotient of the number and assign it to the original number using number / 10 expression. This romoves the last digit of the original number. f. Finally check if the number and the sum are the same, if same its an Armstrong number else not. So display accordingly

**CODE:**

import java.util.\*;

public class armstrongNumber {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

int temp, r, sum = 0;

System.out.println("ENTER A NUMBER : ");

int n = sc.nextInt();

temp = n;

while (n != 0) {

r = n % 10;

sum += (r \* r \* r);

n = n / 10;

}

String print = (temp == sum) ? "ENTERED NUMBER IS AN ARMSTRONG NUMBER" : "ENTERED NUMBER IS NOT AN ARMSTRONG NUMBER";

System.out.println(print);

} catch (InputMismatchException e) {

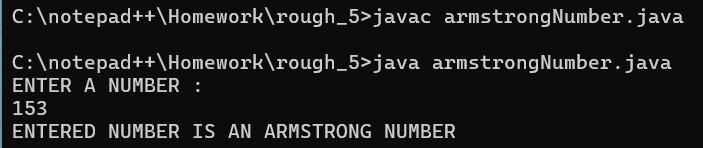
System.out.println("Error!");

}

}

}

**OUTPUT:**

****

**PROGRAM – COUNT NUMBER OF DIGITS**

**Q6:**  Create a program to count the number of digits in an integer.

**Hint =>** a. Get an integer input for the number variable. b. Create an integer variable count with value 0. c. Use a loop to iterate until number is not equal to 0. d. Remove the last digit from number in each iteration e. Increase count by 1 in each iteration. f. Finally display the count to show the number of digits

**CODE:**

import java.util.\*;

public class CountDigits {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

int count = 0, r;

System.out.println("ENTER A NUMBER : ");

int n = sc.nextInt();

while (n != 0) {

r = n % 10;

count++;

n = n / 10;

}

System.out.println("THE TOTAL DIGITS IN " + n + " IS : " + count);

} catch (InputMismatchException e) {

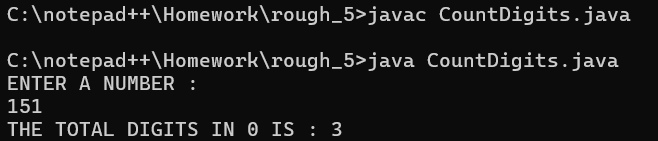
System.out.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – FIND BMI**

**Q7:** Create a program to find the BMI of a person

**Hint =>** a. Take user input in double for the weight (in kg) of the person and height (in cm) for the person and store it in the corresponding variable. b. Use the formula BMI = weight / (height \* height). Note unit is kg/m^2. For this convert cm to meter c. Use the table to determine the weight status of the person



**CODE:**

import java.util.\*;

public class bmi {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

System.out.println("ENTER YOUR WEIGHT IN KG : ");

double w = sc.nextDouble();

System.out.println("ENTER YOUR HEIGHT IN CM : ");

double h = sc.nextDouble();

double hInM = h / 100.0;

double BMI = w / (hInM \* hInM);

System.out.println("YOUR BMI STATUS IS : ");

String status = (BMI <= 18.4) ? "UNDERWEIGHT" : (BMI >= 18.5 && BMI <= 24.9) ? "NORMAL" : (BMI >= 25.0 && BMI <= 39.9) ? "OVERWEIGHT" : (BMI >= 40.0) ? "OBESE" : "INVALID INPUT";

System.out.println(status);

} catch (InputMismatchException e) {

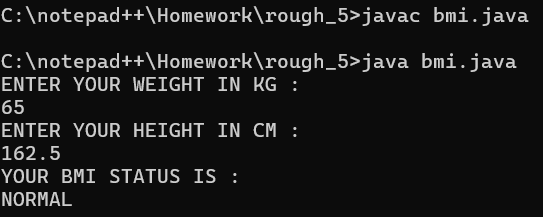
System.out.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – HARSHAD NUMBER**

**Q8:** Create a program to check if a number taken from the user is a Harshad Number.

**Hint =>** a. A Harshad number is an integer which is divisible by the sum of its digits. For example, 21 which is perfectly divided by 3 (sum of digits: 2 + 1). b. Get an integer input for the number variable. c. Create an integer variable sum with initial value 0. d. Create a while loop to access each digit of the number. e. Inside the loop, add each digit of the number to sum. f. Check if the number is perfectly divisible by the sum. g. If the number is divisible by the sum, print Harshad Number. Otherwise, print Not a Harshad Number.

**CODE:**

import java.util.\*;

public class harshadNumber {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

System.out.println("ENTER A NUMBER : ");

int n = sc.nextInt();

int temp = n, sum = 0, r;

while (n != 0) {

r = n % 10;

sum += r;

n = n / 10;

}

String result = ((temp % sum) == 0) ? "ENTERED NUMBER IS A HARSHAD NUMBER" : "ENTERED NUMBER IS NOT A HARSHAD NUMBER";

System.out.println(result);

} catch (InputMismatchException e) {

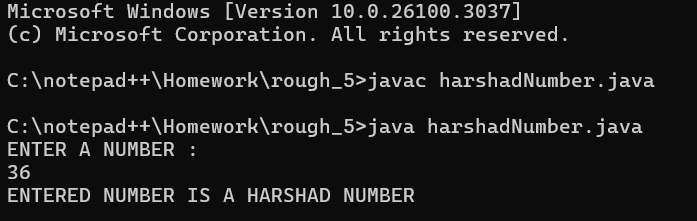
System.out.println("Invalid Input!");

}

}

}

**OUTPUT:**

****

**PROGRAM – ABUNDANT NUMBER**

**Q9:** Create a program to check if a number is an Abundant Number.

**Hint =>** a. An abundant number is an integer in which the sum of all the divisors of the number is greater than the number itself. For example, Divisor of 12: 1, 2, 3, 4, 6 Sum of divisor: 1 + 2 + 3 + 4 + 6 = 16 > 12 b. Get an integer input for the number variable. c. Create an integer variable sum with initial value 0. d. Run a for loop from i = 1 to i < number. e. Inside the loop, check if number is divisible by i. f. If true, add i to sum. g. Outside the loop Check if sum is greater than number. h. If the sum is greater than the number, print Abundant Number. Otherwise, print Not an Abundant Number.

**CODE:**

import java.util.\*;

public class abundantNumber {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

System.out.println("ENTER A NUMBER : ");

int n = sc.nextInt();

int sum = 0;

for (int i = 1; i < n; i++) {

if (n % i == 0) {

sum += i;

}

}

String result = (sum > n) ? "ENTERED NUMBER IS AN ABUNDANT NUMBER" : "ENTERED NUMBER IS NOT AN ABUNDANT NUMBER";

System.out.println(result);

} catch (InputMismatchException e) {

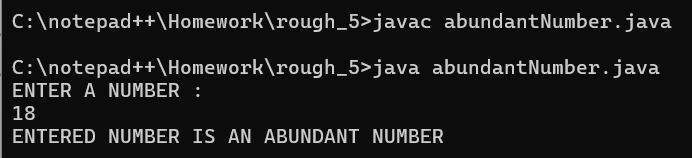
System.out.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – CALCULATOR USING SWITCH CASE**

**Q10:** Write a program to create a calculator using ***switch...case*** .

**Hint =>** a. Create two double variables named first and second and a String variable named op. b. Get input values for all variables. c. The input for the operator can only be one of the four values: "+", "-", "\*" or "/". d. Run a for loop from i = 1 to i < number. e. Based on the input value of the op, perform specific operations using the ***switch...case*** statement and print the result. f. If op is +, perform addition between first and second; if it is -, perform subtraction and so on. g. If op is neither of those 4 values, print Invalid Operator.

**CODE:**

import java.util.\*;

public class calculatorSwitchCase {

public static void main(String[] args) {

double result = 0;

char continueCalc = 'n'; // Initialize with a default value

try (Scanner sc = new Scanner(System.in)) {

do {

System.out.println("Enter the first number:");

double num1 = sc.nextDouble();

sc.nextLine(); // Consume the newline character

System.out.println("Enter the second number:");

double num2 = sc.nextDouble();

sc.nextLine(); // Consume the newline character

System.out.println("Choose an operation: +, -, \*, /");

char operator = sc.nextLine().charAt(0);

switch (operator) {

case '+':

result = num1 + num2;

break;

case '-':

result = num1 - num2;

break;

case '\*':

result = num1 \* num2;

break;

case '/':

if (num2 != 0) {

result = num1 / num2;

} else {

System.out.println("Error! Division by zero.");

continue;

}

break;

default:

System.out.println("Error! Invalid operator.");

continue;

}

System.out.println("The result is: " + result);

System.out.println("Do you want to perform another calculation? (y/n)");

continueCalc = sc.nextLine().charAt(0);

} while (continueCalc == 'y');

System.out.println("Calculator program has ended.");

} catch (InputMismatchException e) {

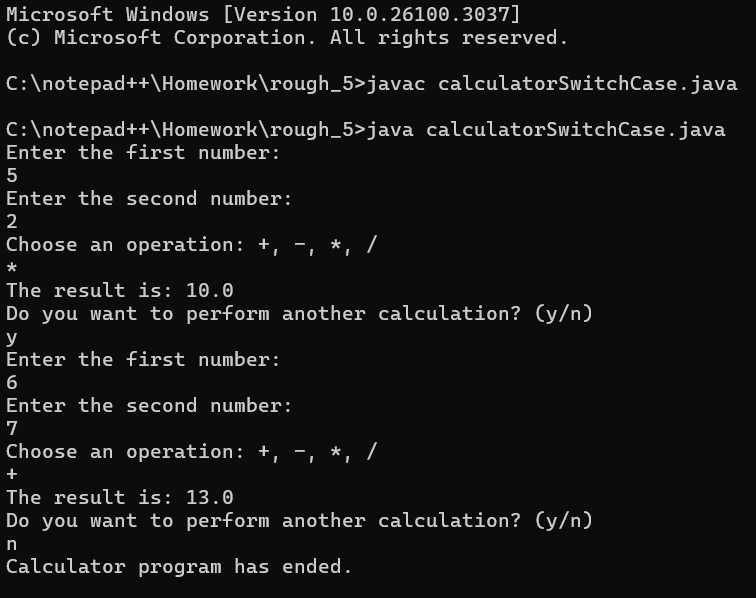
System.out.println("INVALID INPUT !");

}

}

}

**OUTPUT:**

****

**PROGRAM – DAY OF WEEK**

**Q11:** Write a program ***DayOfWeek*** that takes a date as input and prints the day of the week that the date falls on.

Your program should take three command-line arguments: m (month), d (day), and y (year). For m use 1 for January, 2 for February, and so forth. For output print 0 for Sunday, 1 for Monday, 2 for Tuesday, and so forth. Use the following formulas, for the Gregorian calendar (where / denotes integer division): *y* 0 = *y* − (14 − *m* ) / 12 *x* = *y* 0 + *y* 0 /4 − *y* 0 /100 + *y* 0 /400 *m* 0 = *m* + 12 × ((14 − *m* ) / 12) − 2 *d* 0 = ( *d* + *x* + 31 *m* 0 / 12) mod 7

**CODE:**

import java.util.\*;

public class dayOfWeek {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

System.out.println("ENTER THE MONTH (1-12) : ");

int m = sc.nextInt();

System.out.println("ENTER THE DAY (1-31) : ");

int d = sc.nextInt();

System.out.println("ENTER THE YEAR (e.g., 2006) : ");

int y = sc.nextInt();

// int m = Integer.parseInt(args[0]);

// int d = Integer.parseInt(args[1]);

// int y = Integer.parseInt(args[2]);

int y0 = y - (14 - m) / 12;

int x = y0 + y0 / 4 - y0 / 100 + y0 / 400;

int m0 = m + 12 \* ((14 - m) / 12) - 2;

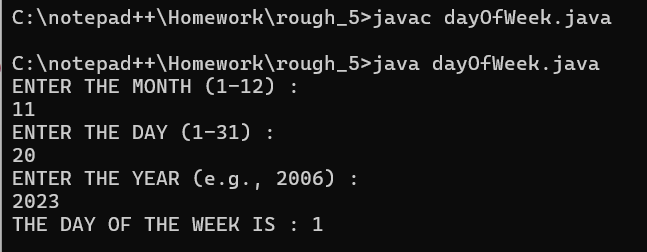
int d0 = (d + x + (31 \* m0) / 12) % 7;

System.out.println("THE DAY OF THE WEEK IS : " + d0);

}

}

**OUTPUT:**

****