**OOP Part-2**

Yash Bhandari | @behumoury

**Q1.**

*/\**

*1)  Write a program to print the following pattern:*

*6 5 4 3 2 1*

*6 5 4 3 2*

*6 5 4 3*

*6 5 4*

*6 5*

*6*

*\*/*

public class Q1 {

    public static void main(String[] args) {

*//Outer loop*

*for* (int i = 6; i > 0; i--) {

*//inner loop*

*for* (int j = 6; j > (6-i); j--) {

                System.out.print(j+" ");

            }

            System.out.println();

        }

    }

}

# 

**Q2.**

*/\*2)    Write a program to print the following pattern:*

*\**

*\* \**

*\* \* \**

*\* \* \* \**

*\* \* \* \* \**

*\* \* \* \**

*\* \* \**

*\* \**

*\**

*\*/*

*public* class Q2 {

*public* *static* void main(String[] args) {

        int n = 5;

*//prints the upper portion*

*for* (int i = 1; i <= n; i++) {

*for* (int j = n-i; j > 0; j--) {

                System.out.print(" ");

            }

*for* (int j = 0; j < i; j++) {

                System.out.print("\* ");

            }

            System.out.println();

        }

*//prints the lower half*

*for* (int i = 1; i < n; i++) {

*for* (int j = 0; j < i; j++) {

                System.out.print(" ");

            }

*for* (int j = n; j > i; j--) {

                System.out.print("\* ");

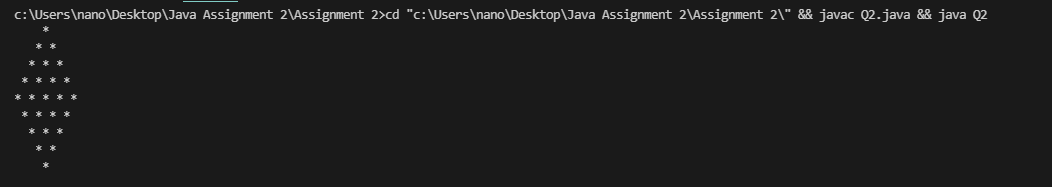
            }

            System.out.println();

        }

    }

}

****

**Q3.**

*// 3)   Write a program to display Fibonacci series up to a given term.*

import java.util.Scanner;

*public* class Q3 {

*// Calculate fabbonacii term with recursion*

*public* *static* int fabonacii(int n) {

*if* (n == 0 || n == 1) {

*return* n;

        }

*return* fabonacii(n - 1) + fabonacii(n - 2);

    }

*public* *static* void main(String[] args) {

        Scanner s = *new* Scanner(System.in);

        int n = s.nextInt();

*// Print fabonnaci term*

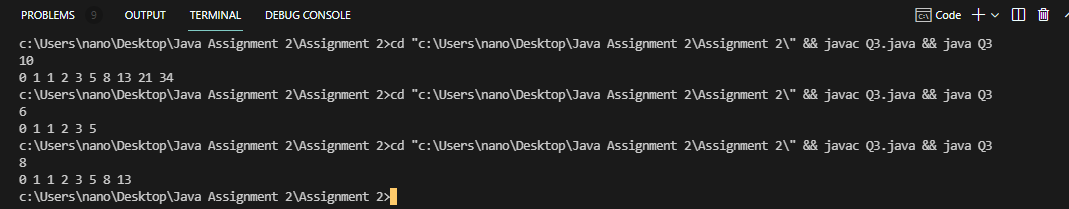
*for* (int i = 0; i < n; i++) {

            System.out.print(fabonacii(i) + " ");

        }

    }

}

****

**Q4.**

*/\**

*4)  Write a program to print the following pattern:*

*(a)*

*(a + b)*

*(a + b + c)*

*(a + b + c + d)*

*(a + b + c + d + e)*

*(a + b + c + d + e + f)*

*\*/*

*public* class Q4 {

*public* *static* void main(String[] args) {

        int ascii\_of\_a = (int)'a';

        int n = 6;

*//Print the pattern*

*for* (int i = 1; i <= n; i++) {

            System.out.print("(a");

*for* (int j = 1; j < i; j++) {

                System.out.print(" + " + (char)(ascii\_of\_a + j));

            }

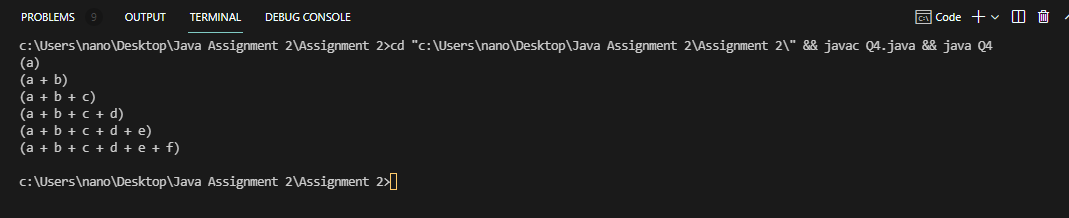
            System.out.print(")");

            System.out.println();

        }

    }

}

****

**Q5.**

import java.util.Scanner;

*//5)    Write a program to print sum of series (i.e. 0, 1, 4, 9, 16…).*

*public* class Q5 {

*//function to calculate sum of series*

*public* *static* int sum\_series(int n){

        int sum  = 0;

*for* (int i = 0; i < n; i++) {

            sum += (int)Math.pow(i, 2);

        }

*return* sum;

    }

*public* *static* void main(String[] args) {

        int n;

        Scanner sc = *new* Scanner(System.in);

*//gets no. of terms to calculate sum till*

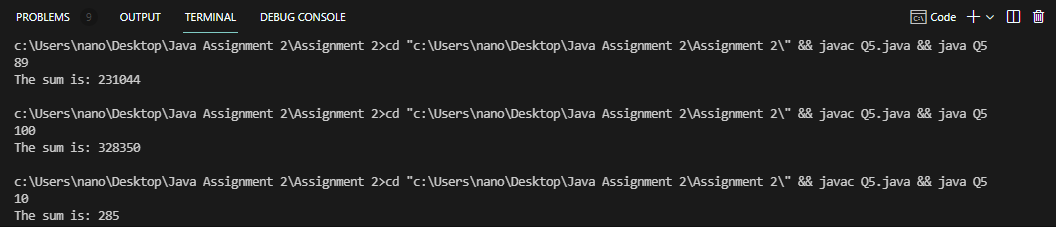
        n = sc.nextInt();

*//Sum and print*

        System.out.println("The sum is: " + sum\_series(n));

    }

}

****

**Q6.**

import java.util.Scanner;

*//6)    Write a program to print sum of series (i.e. 1 + 1/1! + 2/2! + 3/3! + 4/4! +…..).*

*public* class Q6 {

*//Calculates factorial of number by recursion*

*public* *static* int fact(int n){

*if* (n == 0 || n == 1) {

*return* 1;

        }

*return* n\*fact(n-1);

    }

*//Calculates the sum of the series*

*public* *static* double sum\_series(int n){

        double sum = 1;

*for* (int i = 1; i < n; i++) {

            sum += (double)i/fact(i);

        }

*return* sum;

    }

*public* *static* void main(String[] args) {

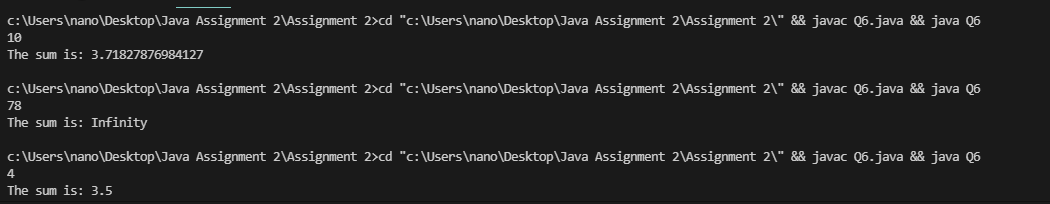
        Scanner sc = *new* Scanner(System.in);

        int n = sc.nextInt();

        System.out.println("The sum is: " + sum\_series(n));

    }

}

****

**Q7.**

*//7)    Write a program to demonstrate widening and narrowing in java.*

*public* class Q7 {

*public* *static* void main(String[] args) {

*//Widening*

        int a = 45;

        long b = a;

        double c = b;

        System.out.println("a = " + a);

        System.out.println("b = " + b);

        System.out.println("c = " + c);

*//Narrowing - Require type casting*

        long d = 150000;

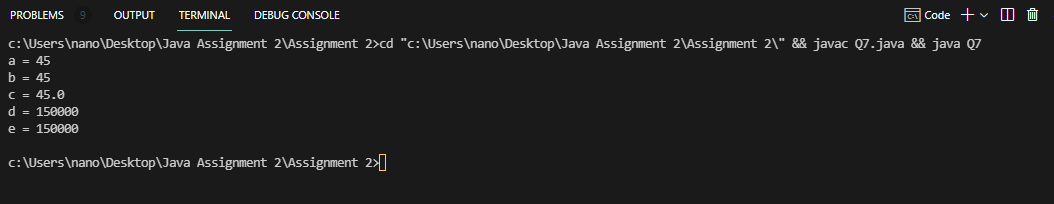
        int e = (int)d;

        System.out.println("d = " + d);

        System.out.println("e = " + e);

    }

}

****

**Q8.**

*//8)    Write a program to find sum of natural numbers from 1 to 100.*

*public* class Q8 {

*public* *static* void main(String[] args) {

        int sum = 0;

*for* (int i = 1; i <= 100; i++) {

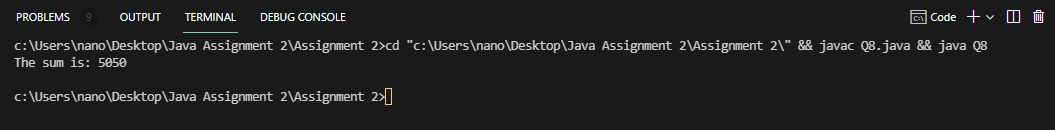
            sum += i;

        }

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

    }

}

****

**Q9.**

import java.util.Scanner;

*//9)    Write a program to calculate the sum of numbers entered by the user until user enters 0.*

*public* class Q9 {

*//calcultes the sum till n numbers*

*public* *static* int sum(int n){

        int sum = 0;

*for* (int i = 1; i <= n; i++) {

            sum += i;

        }

*return* sum;

    }

*public* *static* void main(String[] args) {

        Scanner sc = *new* Scanner(System.in);

        int n = 0;

*do*{

            n = sc.nextInt();

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

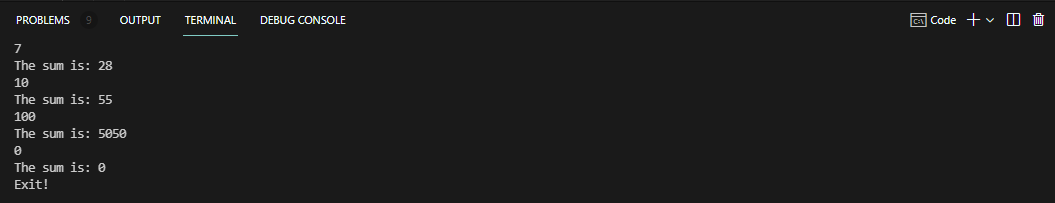
        }

*while*(n != 0);

        System.out.println("Exit!");

    }

}

****

**Q10.**

import java.nio.file.\*;

import java.io.IOException;

*//10)   Write a program to find the size of a specified file.*

*public* class Q10 {

*public* *static* void main(String[] args) throws IOException {

*//Creates the path for file*

        Path p = Paths.get("C:\\Users\\nano\\Desktop\\Java Assignment 2\\Assignment 2\\File.txt");

*try* {

            long size = Files.size(p);

            System.out.println("File Size = " + size + " bytes");

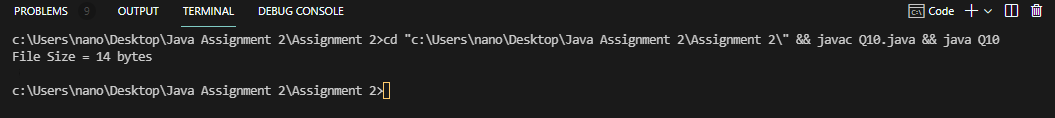
        } *catch* (Exception e) {

            e.getStackTrace();

        }

    }

}

****

**Q11.**

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

*//  Write a program to display the current date time in specific format.*

*public* class Q11 extends Thread{

*public* *static* void main(String[] args) {

        LocalDateTime dt = LocalDateTime.now();

*//Date time formater*

        DateTimeFormatter df = DateTimeFormatter.ofPattern("dd.MM.yyyy");

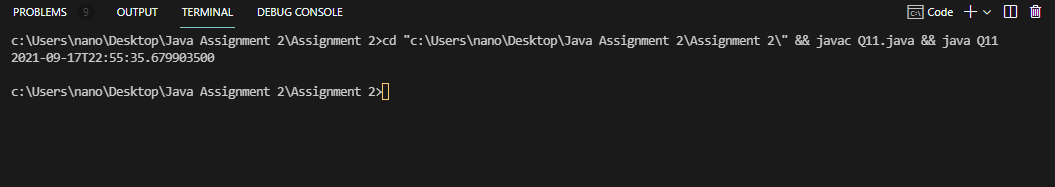
        String date = dt.format(df);

*//printing formated date*

        System.out.println(dt);

    }

}

****

**Q12.**

import java.util.Scanner;

*//12)   Write a program to calculate the sum of two integers and return true if the sum is equal to a third integer.*

*public* class Q12 {

*public* *static* String check(int a,int b,int c){

*if*(a+b == c){

*return* "True";

        }

*else*

*return* "False";

    }

*public* *static* void main(String[] args) {

        Scanner sc = *new* Scanner(System.in);

        int a = sc.nextInt();

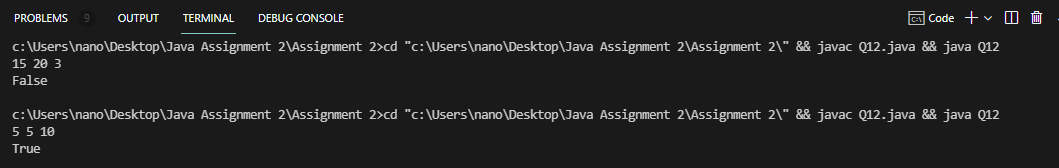
        int b = sc.nextInt();

        int c = sc.nextInt();

        System.out.println(check(a,b,c));

    }

}

****

**Q13.**

import java.util.Scanner;

*//13)   Write a program to convert decimal to binary equivalent.*

*public* class Q13 {

*public* *static* void convert\_To\_Binary(int n){

        String ans = "";

        Integer rem;

*while* (n!=0) {

            rem = n%2;

            ans += rem.toString(rem);

            n = n/2;

        }

        char[] arr = ans.toCharArray();

*for* (int i = arr.length - 1; i >= 0; i--)

            System.out.print(arr[i]);

    }

*public* *static* void main(String[] args) {

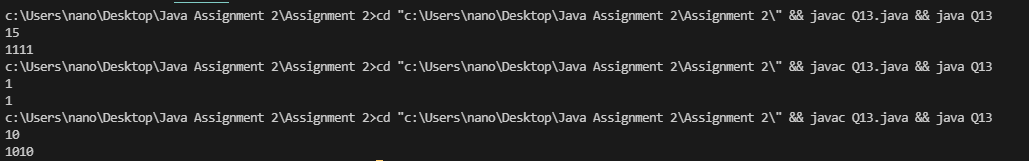
        Scanner sc = *new* Scanner(System.in);

        int n = sc.nextInt();

        convert\_To\_Binary(n);

    }

}

****

**Q14.**

*//  Write a program to convert binary to octal equivalent.*

import java.util.\*;

*public* class Q14 {

*//retruns to octal number after conversion*

*public* *static* int binary\_to\_octal(int binary) {

        int octal = 0, decimal = 0, i = 0;

*///converts binary to decimal*

*while* (binary != 0) {

            decimal += (binary % 10) \* Math.pow(2, i);

            ++i;

            binary /= 10;

        }

        i = 1;

*//converts decimal to octal*

*while* (decimal != 0) {

            octal += (decimal % 8) \* i;

            decimal /= 8;

            i \*= 10;

        }

*return* octal;

    }

*public* *static* void main(String[] args) {

        Scanner sc = *new* Scanner(System.in);

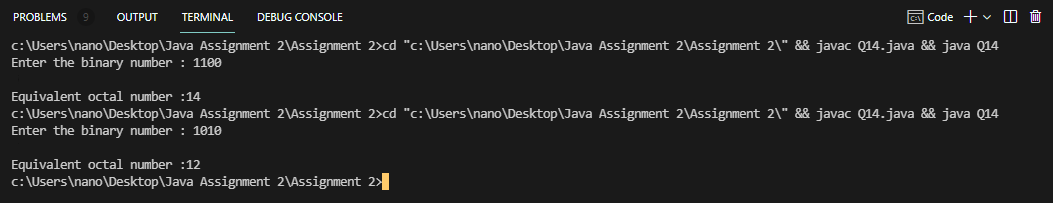
        System.out.print("Enter the binary number : ");

        int binary = sc.nextInt();

        System.out.print("\nEquivalent octal number :" + binary\_to\_octal(binary));

    }

}

****

**Q15.**

*// 15)  Write a program to find factorial of 60.*

*public* class Q15 {

*// Function to find factorial of 60*

*static* void factorial(int n) {

        int res[] = *new* int[500];

*// Initialize result*

        res[0] = 1;

        int res\_size = 1;

*// n! = 1 \* 2 \* 3 \* 4...\*n*

*for* (int x = 2; x <= n; x++)

            res\_size = multiply(x, res, res\_size);

        System.out.println("Factorial of given number is ");

*for* (int i = res\_size - 1; i >= 0; i--)

            System.out.print(res[i]);

    }

*// function to multipl x with the number represented by res[] and size res\_size*

*static* int multiply(int x, int res[], int res\_size) {

        int carry = 0; *// Initialize carry*

*// One by one multiply n with individual digits of res[]*

*for* (int i = 0; i < res\_size; i++) {

            int prod = res[i] \* x + carry;

            res[i] = prod % 10; *// Store last digit of 'prod' in res[]*

            carry = prod / 10; *// Put rest in carry*

        }

*// Put carry in res and increase result size*

*while* (carry != 0) {

            res[res\_size] = carry % 10;

            carry = carry / 10;

            res\_size++;

        }

*return* res\_size;

    }

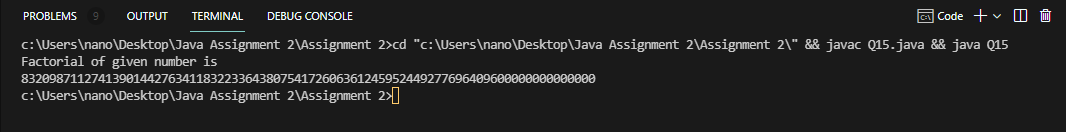
*//Main Class*

*public* *static* void main(String args[]) {

        factorial(60);

    }

}

****

**Q16.**

import java.math.BigInteger;

*//16)   Write a program to check whether the following number is prime or not: 21000254869723212850.*

*public* class Q16 {

*public* *static* void main(String[] args) {

*// long l = 21000254869723212850; cannot store this number*

*//Create big integer and stores value to it*

        BigInteger big\_Int = *new* BigInteger("21000254869723212850");

*//Certainity = 1*

*if* (big\_Int.isProbablePrime(1)) {

            System.out.println("Prime");

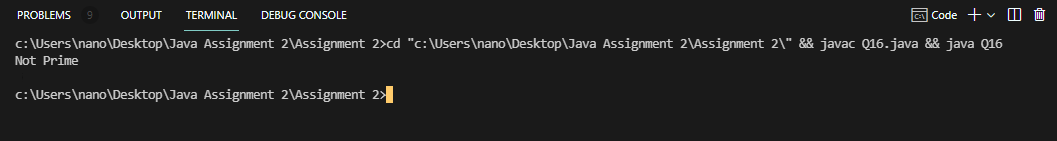
        }

*else*

            System.out.println("Not Prime");

    }

}

****