1. Write a program to count all the prime and composite numbers entered by the user.

   program:

```
             import java.util.Scanner;
public class prime_and_composite{
    public static void main(String args[])
    {
        try {
            Scanner input = new Scanner(System.in);
            System.out.print("Enter the number of numbers: ");
            int n = input.nextInt();
            int[] arr = new int[n];
            System.out.print("Enter the numbers\n");
            for (int i = 0; i < n; i++) {
                arr[i] = input.nextInt();
            }
            int prime, composite;
            prime = 0;
            composite = 0;
            for (int i = 0; i < n; i++) {
                int pri = 0;
                for (int j = 1; j < 100; j++) {
                    if (arr[i] % j == 0)
                        pri = pri + 1;
                }
                if (pri == 2)
                    prime = prime + 1;
                else
                    composite = composite + 1;
            }
            System.out.print("Composite number: " + composite);
```

```
      System.out.print("\nPrime number: " + prime);
    }
    catch(Exception e)
    {
      System.out.print("Input format is not INTEGER"+e);
    }
  }
}
```

Sample Input:

```
Enter the numbers
4
54
29
71
7
59
98
23
```

Sample Output:

```
Composite number:3
Prime number:5
```

Test cases:

1. 33, 41, 52, 61,73,90
2. TEN, FIFTY, SIXTY-ONE, SEVENTY-SEVEN, NINE
3. 45, 87, 09, 5.0 ,2.3, 0.4
4. -54, -76, -97, -23, -33, -98
5. 45, 73, 00, 50, 67, 44

2. Find the Mth maximum number and Nth minimum number in an array and then find the sum of it and difference of it.

program:

```
import java.util.*;
class max
 {
 public static void main(String[] args)
{
 try
{
 Scanner input = new Scanner(System.in);
 System.out.print("enter the size of the array:- ");
 int size = input.nextInt();
 int[] arr = new int[size];
 System.out.println("enter the values in the array:- ");
 for(int i=0;i<size;i++)
{
 arr[i] = input.nextInt();
 }
 Arrays.sort(arr);
 System.out.print("enter the Mth max number:- ");
 int m = input.nextInt();
 System.out.print("enter the Nth min number:- ");
 int n = input.nextInt();
 int max=0,min=0;
 if(m==0)
 System.out.println("please enter the valid input");
 else
{
 max = arr[arr.length-m];
 min = arr[n-1];
 System.out.println("the max is "+max);
 System.out.println("the min is "+min);
 System.out.println("the sum is: "+(max+min));
 System.out.println("the min is: "+(max-min));
 }
}
 catch(Exception e)
 {
   System.out.println("Enter only numbers");
 }
}
}
```

Sample Input:

Array of elements = {14, 16, 87, 36, 25, 89, 34}

M = 1

N = 3

Sample Output:

1stMaximum Number = 89

3rdMinimum Number = 25

Sum = 114

Difference = 64

Test cases:

1. {16, 16, 16 16, 16}, M = 0, N = 1
2. {0, 0, 0, 0}, M = 1, N = 2
3. {-12, -78, -35, -42, -85}, M = 3 , N = 3
4. {15, 19, 34, 56, 12}, M = 6 , N = 3
5. {85, 45, 65, 75, 95}, M = 5 , N = 7

3. Write a program to print the total amount available in the ATM machine with the conditions applied.

Total denominations are 2000, 500, 200, 100, get the denomination priority from the user and the total number of notes from the user to display the total available balance to the user

program:

```
import java.util.*;
public class atm
{
public static void main(String[] args)
{
try {

Scanner sc= new Scanner(System.in);
int arr[]={100,200,500,2000};
System.out.print("Enter the 1st Denomination : ");
int a= sc.nextInt();
System.out.print("Enter the 1st Denomination number of notes: ");
int a1= sc.nextInt();
System.out.print("Enter the 2nd Denomination: ");
int b= sc.nextInt();
System.out.print("Enter the 2nd Denomination number of notes: ");
int b1= sc.nextInt();
System.out.print("Enter the 3rd Denomination : ");
int c= sc.nextInt();
System.out.print("Enter the 3rd Denomination number of notes: ");
int c1= sc.nextInt();
System.out.print("Enter the 4th Denomination: ");
int d= sc.nextInt();
System.out.print("Enter the 4th Denomination number of notes: ");
int d1= sc.nextInt();
if(a==100||a==200||a==500||a==2000&&b==100||b==200||b==500||b==2000&&c==100||c==200||c==500||c==2000&&d==100||d==200||d==500||d==2000)
{
    int e=a*a1+b*b1+c*c1+d*d1;
    System.out.println("Total Available Balance in ATM = " +e);
}
else
{
      System.out.println("Enter the correct Denomination");
}
}
catch(Exception e)
{
    System.out.println("Enter the correct Denomination");
    }




}
}
```

Sample Input:

Enter the 1st Denomination: 500

Enter the 1st Denomination number of notes: 4

Enter the 2nd Denomination: 100

Enter the 2nd Denomination number of notes: 20

Enter the 3rd Denomination: 200

Enter the 3rd Denomination number of notes: 32

Enter the 4th Denomination: 2000

Enter the 4th Denomination number of notes: 1

Sample Output:

Total Available Balance in ATM: 1240**0**

Test Cases:

3 Hidden Test cases (Think Accordingly based on Denominations)

4. Write a program using choice to check

Case 1: Given string is palindrome or not

Case 2: Given number is palindrome or not

program:

```
import java.util.Scanner;
class casep
{
 public static void main(String[] args)
 {
   try
   {
    int a,rev=0,rem,choice;
    String a1,b="";
    char c;
    int d=0,i;
    Scanner sc=new Scanner(System.in);
    System.out.println("Case:");
    choice=sc.nextInt();
    switch(choice)
    {
      case 1:
      {
         System.out.println("Enter the string:");
         a1=sc.next();
         d=a1.length();
             for(i=d-1;i>=0;i--)
             {
               b=b+a1.charAt(i);
              }
   if(a1.equals(b))
   {
      System.out.println("PALINDROME");
    }
   else
   {
       System.out.println("NOT A PALINDROME");
    }
  break;

}
case 2:
{

       System.out.println("Enter a number:");
       a=sc.nextInt();
       int d1=a;
       while(a!=0)
       {
          rem=a%10;
          rev=rev*10+rem;
          a=a/10;
        }
       if(d1==rev)
       {

         System.out.println("PALINDROME");
       }
       else
      {
           System.out.println("NOT A PALINDROME");
         }
        break;
     }
     default:
     {

        System.out.println("Executed");
     }
   }
  }

   catch(Exception e)
   {
      System.out.print("Enter only numbers");
   }
 }
}
```

Sample Input:

Case = 1

String = MADAM

Sample Output:

Palindrome

Test cases:

- MONEY
- 5678765
- MALAY12321ALAM

- MALAYALAM
- 1234.4321

5.Write a program to convert Decimal number equivalent to Binary number and octal numbers**?**

program:

```
import java.util.*;
class bin
{
 public static void main(String[] args)
 {
  try
 {
 Scanner scan;
 int num;

 scan = new Scanner(System.in);
 System.out.println("Decimal Number: ");
 num = Integer.parseInt(scan.nextLine());



 String binary = Integer.toBinaryString(num);
 System.out.println("Binary Number = " + binary);

 String octal = Integer.toOctalString(num);
 System.out.println("Octal = " + octal);
 }
 catch(Exception e)
  {
    System.out.println("Enter only numbers");
 }
}
}
```

Sample Input:

Decimal Number: 15

Sample Output:

Binary Number = 1111

Octal = 17

Test cases:

1. 111
2. 15.2
3. 0
4. B12
5. 1A.2

6. In an organization they decide to give bonus to all the employees on New Year. A 5% bonus on salary is given to the grade A workers and 10% bonus on salary to the grade B workers. Write a program to enter the salary and grade of the employee. If the salary of the employee is less than $10,000 then the employee gets an extra 2% bonus on salary Calculate the bonus that has to be given to the employee and print the salary that the employee will get.

program:

```
import java.util.*;
class salary
{
    public static void main(String[] args)
    {
        try
        {
        double salary;
        double bonus;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the Grade of an employee: ");
        char g=sc.next().charAt(0);
```

```
  System.out.println("Enter the salary of an employee:");
  salary=sc.nextDouble();

  if(g=='A')
  {
     bonus=0.05;
     salary=salary+salary*bonus;
     System.out.println("The Total salary of a employee is:"+salary);

   }
  if(g=='B')
   {
     bonus=0.1;
     salary=salary+salary*bonus;
     System.out.println("The Total salary of a employee is:"+salary);
    }
  if(salary<=0)
   {

     System.out.println("Enter the valid salary of an employee");
   }
  if(g!='A'&&g!='B')
   {
      System.out.println("Enter the correct grade or salary of an employee");
    }
}
catch(Exception e)
 {
    System.out.println("Enter the valid salary of an employee");
  }
```

```
    }
}
```

Sample Input & Output:

```
Enter the grade of the employee: B
Enter the employee salary: 50000
Salary=50000
Bonus=5000.0
Total to be paid:55000.0
```

Test cases:

1. Enter the grade of the employee: A

   Enter the employee salary: 8000

1. Enter the grade of the employee: C

   Enter the employee salary: 60000

1. Enter the grade of the employee: B

   Enter the employee salary: 0

1. Enter the grade of the employee: 38000

   Enter the employee salary: A

1. Enter the grade of the employee: B

   Enter the employee salary: -8000

7. Write a program to print the first n perfect numbers. (Hint Perfect number means **a positive integer that is equal to the sum of its proper divisors)**

program:

```
import java.util.Scanner;
class Perfect
{
        static boolean perfect(int num)
        {
            int sum = 0;
```

```
   for(int i=1; i<num; i++)
   {
      if(num%i==0)
      {
         sum = sum+i;
      }
   }
   if(sum==num)
      return true;
   else
      return false;
}
public static void main(String[] args)
{
  try
  {
   Scanner obj = new Scanner (System.in);
   int n=0;
   System.out.println("enter the value for N");
   int N = obj.nextInt();
   if(N<=0)
   System.out.println("enter the N value correctly");
   if(N==3)
   n=1000;
   if(N==5)
   n=100000000;
   for(int i=1; i<=n; i++)
   {
      if(perfect(i))
         System.out.println(i);
```

```
            }
           }
           catch(Exception e)
           {
                  System.out.println("enter the N value correctly");
            }



       }
 }
```

Sample Input:

N = 3

Sample Output:

First 3 perfect numbers are: 6 , 28 , 496

Test Cases:

1. N = 0
2. N = 5
3. N = -2
4. N = -5
5. N = 0.2

8.Write a program to print the first n perfect numbers. (Hint Perfect number means **a positive integer that is equal to the sum of its proper divisors)**

Sample Input:

N = 3

Sample Output:

First 3 perfect numbers are: 6 , 28 , 496

Test Cases:

1. N = 0
2. N = 5
3. N = -2

4. N = -5
5. N = 0.2

Program

```
import java.util.Scanner;
    import java.lang.Math;

public class DAY2NPERFECTNUMBERS
{
  static int isPerfect(int n)
  {
    int dsum = 0;
    int i;
    for (i = 1; i <= Math.pow(n,0.5); ++i)
    {
      if (n % i == 0)
      {
        if (i == n / i)
        {
          dsum += i;
        }
        else
        {
          dsum += i;
          dsum += n / i;
        }
      }
    }
    dsum = dsum - n;
    if (dsum == n)
```

```
         return 1;
      else
         return 0;
   }

   public static void main(String args[])
   {
      Scanner input=new Scanner(System.in);
      int n, i, temp;
      try
      {
         System.out.print("N = ");
         n=input.nextInt();
         i = 1;
         System.out.print("First "+n+" Perfect numbers are: ");
         while (n > 0) {
            if (isPerfect(i) == 1) {
               System.out.print(i + " ");
               n = n - 1;
            }
            i = i + 1;
         }
         System.out.print("\n");
      }
      catch(Exception e)
      {
         System.out.print("Perfect number not exist for NEGATIVE AND FLOATING
NUMBERS");
      }
   }
```

}

9.Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.

program:

```
import java.util.*;

public class Main

{

 public static void main(String args[])

 {

 Scanner sc = new Scanner(System.in);

 int n;

 System.out.println("Enter the total subjects ");

 n=sc.nextInt();

 int arr[] = new int[n];

 for(int i=0;i<n;i++)

 {

 System.out.println("Enter subject mark:");

 arr[i]=sc.nextInt();


 }


 int total=0;

 for(int i=0;i<n;i++)

 {

 total=total+arr[i];

 }

 System.out.println("The total marks obtained is "+total);

 float percentage;

 percentage = (total / (float)n);
```

```
          System.out.println( "Total Percentage : " + percentage + "%");
          }
         }
```

Sample Input & Output:

```
Enter the marks in python: 90
Enter the marks in c programming: 91
Enter the marks in Mathematics: 92
Enter the marks in Physics: 93
Total= 366
Aggregate = 91.5
DISTINCTION
```

Test cases:

- 18, 76,93,65
- 73,78,79,75
- 98,106,120,95
- 96,73, -85,95
- 78,59.8,76,79

10. Write a program to calculate tax given the following conditions:

- If income is less than or equal to 1,50,000 then no tax
- If taxable income is 1,50,001 – 3,00,000 the charge 10% tax
- If taxable income is 3,00,001 – 5,00,000 the charge 20% tax
- If taxable income is above 5,00,001 then charge 30% tax

program:

```
import java.io.*;
import java.util.*;
public class Main
{
```

```
public static void main(String[] args)
{
    try
    {
        int i;
        float j;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the income");
        i=sc.nextInt();
        if(i<=0)
        {
            System.out.println("Enter the valid income");
        }
        if(i<=150000)
        {
            j=0;
            System.out.println("Tax:"+j);
        }
        if(i>150000&&i<=300000)
        {
            j=i*10/100;
            System.out.println("Tax:"+j);
        }
        if(i>300000&&i<=500000)
        {
            j=i*20/100;
```

```
                System.out.println("Tax:"+j);
            }
            if(i>500000)
            {
                j=i*30/100;
                System.out.println("Tax:"+j);
            }

        }
        catch(Exception e)
        {
                System.out.println("Enter the valid income");
        }
    }
}
```

Sample Input:

Enter the income:200000

Sample Output:

Tax= 20000

Test cases:

1. 400700
2. 2789239
3. 150000
4. 00000
5. -125486

11.Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.

program

```
import java.util.*;

public class grade{

    public static void main(String[] args) {

        try {

            float m1, m2, m3, m4, m5, m6;

            Scanner sc = new Scanner(System.in);

            System.out.println("enter marks in python:");

            m1 = sc.nextInt();

            System.out.println("enter marks in c programming:");

            m2 = sc.nextInt();

            System.out.println("enter marks in mathematics:");

            m3 = sc.nextInt();

            System.out.println("enter marks in physics:");

            m5 = sc.nextInt();

            System.out.println("enter marks in chemistry:");

            m6 = sc.nextInt();

            System.out.println("enter marks in professional ethics:");

            m4 = sc.nextInt();

            if (m1 > 100  || m2 > 100 || m3 > 100 || m4 > 100 || m5 > 100 || m6 > 100)

            {

                throw new NullPointerException("invalid due to higher values.");

            }

            if (m1 <0  || m2 <0 || m3 <0 || m4 <0 || m5 < 0 || m6 < 0)

            {

                throw new ArithmeticException("invalid due to higher values.");

            }
            float total=m1+m2+m3+m4+m5+m6;
            float agg=total/6;
            System.out.println("TOTAL= " + total);
            System.out.println("Aggregate=" + agg);

            if(agg>75)
            {
                System.out.println("DISTINCTION");
            }
            else if(agg>=60 && agg<75)
            {
                System.out.println("FIRST DIVISION");
            }
            else if(agg>=50 && agg<60)
            {
                System.out.println("SECOND DIVISION");
            }
            else if(agg>=40 && agg<50)
            {
                System.out.println("THIRD DIVISION");
            }
            else
            {
                System.out.println("FAIL");
            }
        }
        catch(NullPointerException e)
        {
```

```
         System.out.println("invalid due to higher values.");
      }
      catch(ArithmeticException e)
      {
         System.out.println("invalid due to negative values.");
      }
      catch(Exception e)
      {
         System.out.println("invalid due to floating values.");
      }



   }
}
```

Sample Input & Output:

```
Enter the marks in python: 90
Enter the marks in c programming: 91
Enter the marks in Mathematics: 92
Enter the marks in Physics: 93
Total= 366
Aggregate = 91.5
DISTINCTION
```

Test cases:

1. 18, 76,93,65
2. 73,78,79,75
3. 98,106,120,95
4. 96,73, -85,95
5. 78,59.8,76,79

12. Write a program to print the multiplication table of number m up to n

program:

```
import java.util.Scanner;
class mul
```

```
{
 public static void main(String[] args)
 {
   try
   {
    int i,n,n1;
    Scanner s= new Scanner(System.in);
    System.out.print(" Enter the number : ");
    n= s.nextInt();
    System.out.print("Enter which table u want:");
    n1=s.nextInt();
    for(i=1;i<=n;i++)
    {
        System.out.print(i+"x"+n1+"="+i*n1+"\n");
    }


    }
    catch(Exception e)
    {
        System.out.print("Enter only numbers:");
    }
  }
}
```

Sample Input:

M = 4

N = 5

Sample Output:

```
1x4=4

2x4=8

3x4=12

4x4=16

5x4=20
```

Test cases:

1. M = 6, N = -3
2. M = -3, N = 5
3. M = 4, N = 0
4. M = 0, N = 0
5. M = -5, N = -5

13. Write a program to read the numbers until -1 is encountered. Find the average of positive numbers and negative numbers entered by user.

program:

```
        import java.util.*;
class positive
{
    public static void main(String[] args)
    {
        try
        {
            Scanner input = new Scanner(System.in);
            float pos = 0,neg = 0,num=0,p=0,n=0;
            while(num!=-1)
            {
                System.out.print("enter the number:- ");
                num = input.nextInt();
                if(num>0)
                {
                    pos++;
                p=p+num;
            }

            else if(num<0)
            {
                neg++;
                n=n+num;
            }
        }
        System.out.println("the no.of.negative values are "+neg);
        System.out.println("the no.of.positive values are "+pos);
        System.out.println("the sum of positive values "+p);
        System.out.println("the sum of negative values"+n);
        float p1=p/pos;
        float p2=n/neg;
        System.out.println("avg of positive numbers"+p1);
        System.out.println(" avg of negative numbers"+p2);
    }
    catch(Exception e)
    {
        System.out.println("Enter a valid number");
    }
  }
}
```

Sample Input:

```
Enter -1 to exit…
Enter the number: 7
Enter the number: -2
Enter the number: 9
Enter the number: -8
Enter the number: -6

Enter the number: -4

Enter the number: 10

Enter the number:  -1
```

Sample Output:

```
The average of negative numbers is: -5.0

The average of positive numbers is : 8.66666667
```

Test cases:

1. -1,43, -87, -29, 1, -9
2. 73, 7-6,2,10,28,-1
3. -5, -9, -46,2,5,0
4. 9, 11, -5, 6, 0,-1
5. -1,-1,-1,-1,-1

14. Write a program to read a character until a ***** is encountered. Also count the number of uppercase, lowercase, and numbers entered by the **users.**

program:

```
import java.util.Scanner;
class alpha
{
 public static void main(String[] args)
 {
 Scanner s= new Scanner(System.in);
 System.out.print("Enter any Character: ");
 char ch=s.next().charAt(0);
 int l_count=0,u_count=0,num_count=0;
 while (ch!='*')
 {
 if(ch>='A'&&ch<='Z')
 u_count++;
if(ch>='a'&&ch<='z')
l_count++;
if(Character.isDigit(ch))
num_count++;
System.out.print("Enter any Character: ");
ch=s.next().charAt(0);
}
System.out.println("Total count of lower case: "+l_count);
System.out.println("Total count of upper case: "+u_count);
System.out.println("Total count of numbers: "+num_count);

}
}
```

Sample Input:

```
Enter * to exit…
Enter any character: W
Enter any character: d
Enter any character: A
Enter any character: G
Enter any character: g
Enter any character: H
Enter any character: *
```

Sample Output:

```
Total count of lower case:2
Total count of upper case:4
Total count of numbers =0
```

Test cases:

1. 1,7,6,9,5
2. S, Q, l, K,7, j, M
3. M, j, L, &, @, G

4. D, K, I, 6, L, *
5. *, K, A, e, 1, 8, %, *

15. Write a program to calculate the factorial of number using recursive function.

program:

```
import java.util.Scanner;
class fact1
{
 public static void main(String args[])
{
 try
 {
 Scanner scanner = new Scanner(System.in);
 System.out.println("Enter the number:");
 int num = scanner.nextInt();
 int factorial = fact(num);
 System.out.println("Factorial of entered number is: "+factorial);
 }
 catch(Exception e)
 {
   System.out.println("Enter only numbers");
  }
}



 static int fact(int n)
 {
 int output;
 if(n==1)
{
 return 1;
 }
 output = fact(n-1)* n;
 return output;
 }
}
```

Sample Input & Output:

Enter the value of n: 6

Sample Input & Output:

The factorial of 6 is: 720

Test cases:

1. N = 0
2. N = -5
3. N = 1
4. N = M
5. N = %

16. Write a Program to Find the Nth Largest Number in a array.

program:

```
import java.util.*;
 class large
{
 public static void main(String[] args)
{
try{
 Scanner input = new Scanner(System.in);
 System.out.print("enter the size of the array:- ");
 int size = input.nextInt();
 int[] arr = new int[size];
 System.out.println("enter the values in the array:- ");
 for(int i=0;i<size;i++){
 arr[i] = input.nextInt();
 }
 Arrays.sort(arr);
 System.out.print("enter the Mth largest number:- ");
 int m = input.nextInt();
 int max=0;
 if(m<=0)
 System.out.println("please enter the valid input");
 else{
 max = arr[arr.length-m];
 System.out.println("the largest number is "+max);
}
 }
catch(Exception e)
{
    System.out.println("Invalid due to string or float size or value");
}
}
 }
```

Sample Input:

List : {14, 67, 48, 23, 5, 62}

N = 4

Sample Output:

4th Largest number: 23

Test cases:

1. N = 0

2. N = -5
3. N = 1
4. N = M
5. N = %

17. Write a program to convert the Binary to Decimal, Octal

program:

```
import java.util.Scanner;

public class binarytodecimaloctalhexadecimal {

    public static void main(String[] args) {
        Scanner myScanner=new Scanner(System.in);

        System.out.print("Enter a Binary number : ");
        String binary=myScanner.nextLine();

        int num=Integer.parseInt(binary,2);
        System.out.println("Decimal value is : "+num);
        String octal=Integer.toOctalString(num);
        System.out.println("Octal value is : "+octal);
        String hexa=Integer.toHexString(num);
        System.out.println("Hexadecimal value is : "+hexa);
    }

}
```

Sample Input:

Given Number: 1101

Sample Output:

Decimal Number: 13

Octal:15

Test cases:

1. 211
2. 11011
3. 22122
4. 111011.011
5. 1010.0101

18. Write a program to find the number of special characters in the given statement

program:

```
import java.util.Scanner;

public class DAY3SPECIALCHARACTERS{
    private static Scanner sc;
    public static void main(String[] args) {
        String aldisp_str;
        int i, alph, digi, spl;
        alph = digi = spl = 0;
        char ch;

        sc= new Scanner(System.in);
        System.out.print("\nPlease Enter Alpha Numeric Special String =  ");
        aldisp_str = sc.nextLine();

        for(i = 0; i < aldisp_str.length(); i++)
        {
            ch = aldisp_str.charAt(i);
            if(ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z' ) {
                alph++;
            }
            else if(ch >= '0' && ch <= '9') {
                digi++;
            }
            else {
                System.out.println(ch);
                spl++;
            }
        }
        System.out.println("Number of Special Characters   =  " + spl);
    }
}
```

Sample Input:

Given statement: Modi Birthday @ September 17, #&$% is the wishes code for him.

Sample Output:

Number of special Characters: 5

19. Write a Program to Remove the Duplicate Items from a array.

program:

```
import java.util.*;
public class duplicate
{
    static Scanner input = new Scanner(System.in);
    public static void main(String[] args) {
        System.out.print("enter the size of the array:- ");
        int size = input.nextInt();
        int[] arr = new int[size];
        SortArr.Dummyarr(arr,input);
    }
}
```

```
class SortArr {
    public static void Dummyarr(int[] arr, Scanner input) {
        int i, j;
        System.out.println("enter the elements in the array:- ");
        for (i = 0; i < arr.length; i++) {
            arr[i] = input.nextInt();
        }
        // sorting an array using bubbel sort
        for (i = 0; i < arr.length - 1; i++) {
            for (j = i + 1; j < arr.length; j++) {
                if (arr[i] > arr[j]) {
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }

        List<Integer> list = new ArrayList<>();

        for(i=0;i<arr.length-1;i++){
            if(arr[i]!=arr[i+1])
                list.add(arr[i]);
        }
        list.add(arr[arr.length-1]);
        System.out.println(list);
    }
}
```

Sample Input:

```
Enter the number of elements in array:7
Enter element1:10
Enter element2:20
Enter element3:20
Enter element4:30
Enter element5:40
Enter element6:40
Enter element7:50
```

Sample Output:

```
Non-duplicate items:
[10, 20, 30, 40, 50]
```

20. Bank is a class that provides method to get the rate of interest. But, rate of interest may differ according to banks. For example, SBI, ICICI and AXIS banks are providing 8.4%, 7.3% and 9.7% rate of interest. Write a Java program for above scenario.

Sample Input SBI, 8.4

program:

```
import java.io.*;
import java.util.*;
class Bank
{
   float getRateOfInterest()
   {
      return 0;
   }
}
class SBI extends Bank
{
   float getRateOfInterest()
   {
      return 8.4f;
```

```
    }
}
class ICICI extends Bank
{
    float getRateOfInterest()
    {
        return 7.3f;
    }
}
class AXIS extends Bank
{
    float getRateOfInterest()
    {
        return 9.7f;
    }
}
class DAY4BANKINTEREST
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        String B;
        Bank b;
        float c;
        float d;
        float e;
        float f;

        b=new SBI();
        System.out.println("SBI Rate of Interest: "+b.getRateOfInterest());
```

```
c= b.getRateOfInterest();
b=new ICICI();
System.out.println("ICICI Rate of Interest: "+b.getRateOfInterest());
d=b.getRateOfInterest();
b=new AXIS();
System.out.println("AXIS Rate of Interest: "+b.getRateOfInterest());
e= b.getRateOfInterest();
System.out.println("Enter the bank name:");
B=sc.next();
System.out.println("Enter the ROI:");
f=sc.nextFloat();
if(f!=c&&f!=d&&f!=e)
{

   System.out.println("RATE OF INTEREST IS INVALID");
}
if(B.equals("SBI") && f==c)
{
   System.out.println("VALID");
}
if(B.equals("ICICI") && f==d)
{
   System.out.println("VALID");
}
if(B.equals("AXIS") && f==e)
{
   System.out.println("VALID");
}
if(!B.equals("SBI")&&!B.equals("ICICI")&&!B.equals("AXIS"))
{
```

```
            System.out.println(" BANK NAME INVALID");
        }



    }
}
```

Sample Output

Test case

- SBI, 8.3
- ICICI, 7.3
- AXIS, 9.7
- SBI, 8.6
- AXIX, 7.6

21. Bring out the situation in which member names of a subclass hide members by the same name in the super class. How it can be resolved? Write Suitable code in Java and

Implement above scenario with the Parametrized Constructor (accept int type parameter) of the Super Class can be called from Sub Class Using super () and display the input values provided.

program:

```
import java.io.*;
        import java.util.*;
class base
{
    int i;
    base(int a)
    {
        i=a;
    }
}
class derived extends base
{
    int i;
    int j;
    derived(int a,int b)
    {
        super(a);
        i=a;
        j=b;
    }
    void print()
    {
        System.out.println("Super class instance variable");
        System.out.println(super.i);
        System.out.println("Sub Class instance variables");
        System.out.println(i);
        System.out.println(j);
    }
}

class DAY5SUPERCLASSSUBCLASS
{
    public static void main(String []arg)
    {
        try
        {
            Scanner sc=new Scanner(System.in);
            int a,b,c;
            System.out.println();
            a=sc.nextInt();
            System.out.println();
            b=sc.nextInt();
            derived d = new derived(a,b);
            d.print();
        }
        catch(Exception e)
        {
            System.out.println("Invalid due to string exception or number format
exception");
        }
    }
}
```

Sample Input : 100, 200

Sample Output : 100, 200

Test Cases

- 10, 20
- -20, -30
- 0, 0
- EIGHT FIVE
- 10.57, 12.58

22. Display Multiplication table for 5 and 10 using various stages of life cycle of the thread by generating a suitable code in Java.

Sample Input 5, 10

5 X 1 = 5

5 X 2 =10

….

10 X 1 =10

10 X 2 = 20

….

program:

```
import java.util.*;
public class multiplicationtable5and10
{
    public static void main(String[] args)
    {
        int a,b,i,cal;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the a and b");
        if(!s.hasNextInt())
        {
            System.out.println("Invalid");
            return;
        }
        a=s.nextInt();
        if(!s.hasNextInt())
        {
            System.out.println("Invalid");
            return;
        }
        b=s.nextInt();
        if(a<=0 && b<=0)
        {
            System.out.println("INVALID");
        }
        else
        {
            for(i=1;i<=10;i++)
            {
                cal=a*i;
                System.out.println(a +"*" +i+"=" +cal);
            }
            for(i=1;i<=10;i++)
            {
                cal=b*i;
                System.out.println(b +"*" +i+"="+cal );
            }

        }

    }
}
```

Test Cases:

- 10, 20
- -10, -30
- 0, 0
- SIX, SIX
- 9.8, 9.6

23. Using the concepts of thread with implementing Runnable interface in Java to generate Fibonacci series.

program:

```
import java.io.*;
public class b_fibanocci
{
    public static void main(String[] args)
    {
        try
        {
            int a=0, b=1, c=0;
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

            System.out.print("Enter the Limit for fabonacci: ");

            int n = Integer.parseInt(br.readLine());
            System.out.println("\n  ENTER THE POSITIVE NUMBER");
```

```
         System.out.println("Fibonacci series:");
         while (n>0)
         {
            System.out.print(c+" ");
            a=b;
            b=c;
            c=a+b;
            n=n-1;
         }
      }
      catch (Exception e)
      {
         System.out.println("ENTER THE POSITIVE NUMBER");
      }
   }
}
```

Sample Input : 5

Sample Output : 0 1 1 2 3 …..

Test Cases

- 7
- -10
- 0
- EIGHT FIVE
- 12.65

24. Generate a Java code to find the sum of N numbers using array and throw ArrayIndexOutOfBoundsException when the loop variable beyond the size N.

program:

```
import java.util.Scanner;


public class c_sum_of_numbers {

   public static void main(String [] args) {

      try {

         Scanner sc = new Scanner(System.in);

         int a, sum = 0;

         System.out.println("enter the total number of elements to be: ");

         a = sc.nextInt();

         if(a<0){
             throw new Exception("invalid input");
         }
         int arr[] = new int[10];
         System.out.println("enter the elements: ");
         for (int i = 0; i < a; i++) {
             arr[i] = sc.nextInt();
         }
         for (int i = 0; i < a; i++) {
             sum = sum + arr[i];
         }
         System.out.println(sum);
      }
      catch (Exception e){
         System.out.println("invalid input");
      }
   }


}
```

Sample Input : 5
1 2 3 4 5

Sample Output : 15

Test Cases

- 4, 10
- -10
- 0
- EIGHT SEVEN
- 12.68

25. Using the concepts of thread with implementing Runnable interface in Java to find whether a given number is prime or not.

program:

```
import java.util.*;
public class threprim implements Runnable
{
    public void run()
    {
        int i,m=0,flag=0;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter number: ");
        if(!sc.hasNextInt())
        {
            System.out.println("Invalid");
            return;
        }
        int n=sc.nextInt();
        if(n<0)
        {
            System.out.println("Invalid");
            return;
        }
        m=n/2;
        if(n==0||n==1)
        {
            System.out.println(n+" is not prime number");
        }
        else
        {
            for(i=2;i<=m;i++)
            {
                if(n%i==0)
                {
                    System.out.println(n+" is not prime number");
                    flag=1;
                    break;
                }
            }
            if(flag==0)
            {
                System.out.println(n+" is prime number");
            }
        }
    }
    public static void main(String[] args)
    {
        threprim ex = new threprim();
        Thread t1= new Thread(ex);
        t1.start();


    }
}
```

Sample Input : 5
Sample Output : 5 is Prime

Sample Output : 15
Test Cases

- 4
- -10
- 0
- EIGHT SEVEN
- 11.48

- **26.** **Given a string s consisting of words and spaces, return the length of the** last **word** ***in the string*. A** word **is a maximal substring consisting of non-space characters only. There will be at least one word, consists of only English letters and spaces ' '.**

program:

```
import java.util.*;

public class d_last_word {

    public static void main(String[] args){

        try {

            String s1;

            int length1=0;

            Scanner sc = new Scanner(System.in);

            System.out.println("enter the string:");

            s1 = sc.nextLine();

            String[] words=s1.split(" ");

            if(words.length>0)

            {

                length1=words[words.length-1].length();

            }

            else {

                length1=0;

            }

            System.out.println("length= "+length1);

        }

        catch(Exception e)

        {

            System.out.println("enter valid input.");

        }

    }
```

}

Example 1:
Input: s = "Hello World"
Output: 5
Explanation: The last word is "World" with length 5.

Test Case

Test Case	**Inputs-1**
•	Maximal Substring Consisting
•	**lea@st one wor2d**
•	1254 98076
•	& * ( ) % # $
•	letters and spaces