

Exercise -1

Date: 19/09/2020

Day: Saturday

Name : Gorantla Venkatesh

Reg .no : 9919004093

Subject : java programming language

Code :CSE18R272



1. Find mean and standard deviation.

Program:

```
Import java.lang.*;
Import java.util.Scanner;
Class Main_9919004093 {
Public static void main (String [] args)
{
    System.out.println("Enter 5 numbers to find standard deviation");
    Scanner in = new Scanner (System.in);
    Double arr[]= new double [5];
    Double sum=0, mean=0,dist;
    For (int i=0; i<5; i++)
    {

        Arr[i]=in.nextDouble();
        Sum+=arr[i];
    }
    Mean=sum/5;
    Sum=0;
    System.out.println("Mean : "+mean);

    For (int i=0; i<5; i++)
    {
```



```

        Dist=Math.pow((arr[i]-mean),2);
        Sum+=dist;
    }
    Mean=sum/5;
    Double deviation=Math.sqrt(mean);
    System.out.println("Deviation : "+ deviation);
}
}

```

Output ::

Enter 5 numbers to find standard deviation

1 2 3 4 5

Mean : 3.0

Deviation : 1.4142135623730951

2.Find the nCr and nPr.

Program:

Import java.util.Scanner;

Class Main_9919004093

```

{
    Public static void main(String args[])
    {

```



```

    Int n, r;
    Scanner s = new Scanner(System.in);

    System.out.print("Enter Value of n : ");
    N = s.nextInt();
    System.out.print("Enter Value of r : ");
    R = s.nextInt();

    System.out.print("NCR = " + (fact(n)/(fact(n-r)*fact(R))));
    System.out.print("\nNPR = " + (fact(n)/(fact(n-r))));

}

Public static int fact(int num)
{
    Int fact=1, i;
    For(i=1; i<=num; i++)
        Fact = fact*i;
    Return fact;
}

}

```

Output ::

Enter Value of n : 6



Enter Value of r : 4

NCR = 15

NPR = 360

3.Print all prime numbers in the given range.

Program ::

Import java.util.Scanner;

Class Main_9919004093

{

Public static void main(String args[])

{

Int start, end;

Scanner s = new Scanner(System.in);

System.out.print("set a starting number: ");

Start =s.nextInt();

System.out.print("set a ending number: ");

End =s.nextInt();

For (int i=start;i<=end;i++)

If(prime(i))

System.out.println(+i);

}



```
Public static boolean prime(int n)
{
    For (int i=2;i<=n/2;i++)
        If(n%i==0)
            Return false;
    Return true;
}
```

Output ::

Set a starting number: 2

Set a ending number: 11

2

3

5

7

11

4. Find sum of the digits.

Program::

Import java.util.Scanner;



Class Main_9919004093

```
{  
    Public static void main(String args[])  
    {  
        Int num,sum=0,r;  
        Scanner s = new Scanner(System.in);  
        System.out.println("Enter a number:");  
        Num =s.nextInt();  
        While (num >0)  
        {  
            R=num%10;  
            Sum+=r;  
            Num=num/10;  
        }  
        System.out.println("sum of the digits :"+sum);  
    }  
}
```

Output ::

Enter a number:

333

Sum of the digits :9

5. Check whether a given number is palindrome or not.

Program :



```

Import java.util.Scanner;
Class Main_9919004093
{
    Public static void main(String args[])
    {
        Int num,n;
        Scanner s = new Scanner(System.in);
        System.out.print("Input a number: ");
        Num =s.nextInt();
        Int sum =0,r;
        N=num;
        While (num >0)
        {
            R=num%10;
            Sum=(sum*10)+r;
            Num=num/10;
        }
        If (n== sum)
            System.out.println( n + " is a palindrome");
        Else
            System.out.println( n + " is not palindrome");

    }
}

```

Output::



Input a number: 98789

98789 is a palindrome

6. Check whether a given number is prime factor or not.

Program:

Import java.util.Scanner;

Class Main {

Public static void main(String[] args) {

Scanner s= new Scanner(System.in);

System.out.println("Enter a number");

Int n=s.nextInt();

Int mulprime=1,num=n;

While(n%2 ==0)

{

System.out.println(2+" ");

Mulprime=2;*

N=n/2;

}

For(int i=3;i<=n/2;i=i+2)

{

While(n%i==0)



```

    {
        System.out.println(l + " ");
        N=n/l;
        Mulprime*=l;
    }
}

If(n>2)
{
    System.out.println(n);
    Mulprime*=n;
}

If (num==mulprime)
{
    System.out.println("prime factor number");
}

Else
{
    System.out.println("this is not prime factor");
}

}
}

```

Output::



Enter a number

5

5 prime factor number

7. Check whether a given number is perfect number or not.

Program:

Import java.util.Scanner;

Class Main_9919004093 {

Public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.println("Enter a number:");

Int n=s.nextInt();

Int sum=0;

For (int i=1;i<6;i++)

If(n%i==0)

Sum+=i;

If(sum==n)

System.out.println(n+" is a perfect number");

Else

System.out.println(n+" is not a perfect number");

}

}



Output ::

Enter a number:

6

6 is a perfect number

8. Check whether a given number is deficient number or not.

Program :

Import java.util.Scanner;

Public class Deficient {

Public static void main(String[] args){

Int n,sum =0;

Scanner s = new Scanner(System.in);

System.out.print("Enter a number: ");

N = s.nextInt();

For(int l = 1; l < n; i++)



```
{  
  
    If( $n \% i == 0$ )  
  
    {  
  
        Sum = sum + i;  
  
    }  
  
}  
  
If(sum < n)  
  
{  
  
    System.out.println("The number is Deficient");  
  
}  
  
Else  
  
{  
  
    System.out.println("The number is not a Deficient");  
  
}
```



```
}  
}
```

Output::

Enter a number: 6

The number is not a Deficient

9. Apply any one of the sorting algorithm.

Program :

Import java.util.Scanner;

Class Main_9919004093 {

Public static void main(String []args) {

Int n, i, j, temp;

Scanner s = new Scanner(System.in);

System.out.println("Enter the number of integers to sort:");

N = s.nextInt();

Int arr[] = new int[n];

System.out.println("Enter " + n + " integers: ");



```
For (l = 0; l < n; i++)
```

```
    Arr[i] = s.nextInt();
```

```
For (l = 0; l < ( n - 1 ); i++) {
```

```
    For (j = 0; j < n - l - 1; j++) {
```

```
        If (arr[j] > arr[j+1])
```

```
        {
```

```
            Temp = arr[j];
```

```
            Arr[j] = arr[j+1];
```

```
            Arr[j+1] = temp;
```

```
        }
```

```
    }
```

```
}
```

```
System.out.println("Sorted list of integers:");
```

```
For (l = 0; l < n; i++)
```

```
    System.out.println(arr[i]);
```

```
}
```

```
}
```

Output ::

Enter the number of integers to sort:

5

Enter 5 integers:



1 5 6 4 3

Sorted list of integers:

1

3

4

5

6

10. Number conversion from decimal to binary.

Program:

Class Convert{

Public void convertBinary(int n){

Int b[] = new int[40];

Int i= 0;

While(n > 0){

B[i++] = n%2;

N = n/2;

}

For(int j= i-1;j>= 0;j--){

System.out.print(b[j]);

}

}




```
Public static void main(String a[]){  
    Convert obj = new Convert();  
  
    System.out.println("Binary for 12: ");  
  
    Obj.convertBinary(12);  
  
    System.out.println("\nBinary for 25: ");  
  
    Obj.convertBinary(25);  
  
    System.out.println("\n Binary for 19: ");  
    Obj.convertBinary(19);  
}  
}
```

Output ::

Binary for 12:

1100

Binary for 25:

11001

Binary for 19:

10011



11. Write a program to complete and exp€

$$\text{Exp€} = 1 + x/1! + x^2/2! + x^3/3! + \dots$$

Program:

```
Import java.util.Scanner;
```

```
Public class Expon {
```

```
    Static float exponential(int n, float x)
```

```
{
```

```
    Float sum = 1;
```

```
    For (int l = n - 1; l > 0; --l )
```

```
        Sum = 1 + x * sum / l;
```

```
    Return sum;
```



```
}
```

```
Public static void main (String[] args)
```

```
{
```

```
System.out.print("Enter a Number : ");
```

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

```
Int n = sc.nextInt();
```

```
Float x = 1;
```

```
System.out.println("e^x = "+exponential(n,x));
```

```
}
```

```
}
```

Output ::

Enter a Number : 7

$E^x = 2.7180555$



12. Program to compute row sum, column sum and trace of a matrix.

Program:

```
Import java.io.*;
```

```
Class Matrix {
```

```
Static int m = 4;
```

```
Static int n = 4;
```

```
Static void row_sum(int arr[][])
```

```
{
```

```
Int i,j,sum = 0;
```

```
System.out.print( "\nSum of each row:\n\n");
```



```
For (i = 0; i < 4; ++i) {
```

```
For (j = 0; j < 4; ++j) {
```

```
Sum = sum + arr[i][j];
```

```
}
```

```
System.out.println("row + i + " = " + sum);
```

```
Sum = 0;
```

```
}
```

```
}
```

```
Static void column_sum(int arr[][])
```

```
{
```



```
Int l,j,sum = 0;
```

```
System.out.print( "\nSum of each col:\n\n");
```

```
For (l = 0; l < 4; ++l) {
```

```
    For (j = 0; j < 4; ++j) {
```

```
        Sum = sum + arr[j][l];
```

```
    }
```

```
    System.out.println( "col "+ l + " = " + sum);
```

```
    Sum = 0;
```

```
}
```

```
}
```

```
Public static void main (String[] args) {
```

```
    Int l,j;
```

```
    Int [][]arr = new int[m][n];
```

```
    Int x = 1;
```



For (l = 0; l < m; i++)

For (j = 0; j < n; j++)

Arr[i][j] = x++;

Row_sum(arr);

Column_sum(arr);

}

}

Output::

Sum of each row:

Row 0 = 10

Row 1 = 26

Row 2 = 42

Row 3 = 58

Sum of each col:

Col 0 = 28

Col 1 = 32



Col 2 = 36

Col 3 = 40

13. Write a program to encrypt a code in Caesar's code

Program:

```
Class Caesar {  
    String plain;  
    Int key;  
  
    Caesar(String text,int k)  
    {  
        Plain =text;  
        Key =k;  
    }  
  
    String encrypt()  
    {  
        String out="";char ch;  
  
        For(int i=0;i<plain.length();i++)  
        {  
            If(Character.isUpperCase(plain.charAt(i)))  
            Ch=(char)((((int)plain.charAt(i)-97+key)%26+97);  
            Else
```




```

        Ch=(char)((((int)plain.charAt(i)-97+key)%26+97);
        Out =out+ch;
    }
    Return out;
}

```

```

Public static void main(String[] args) {
    Caesar c=new Caesar("welcome",3);
    String out=c.encrypt();
    System.out.println(" The encrypted is : "+out);
}
}

```

Output :

The encrypted is : zhofrph

14. Write a program Java Program to implement the Mono alphabetic Cipher.

Program:



Class MONO

{

Static String encoder(char[] key)

{

String encoded = "";

Boolean[] arr = new boolean[26];

For (int i = 0; i < key.length; i++)

{

If (key[i] >= 'A' && key[i] <= 'Z')

{



If (arr[key[i] - 65] == false)

{

Encoded += (char) key[i];

Arr[key[i] - 65] = true;

}

}

Else if (key[i] >= 'a' && key[i] <= 'z')

{

If (arr[key[i] - 97] == false)

{

Encoded += (char) (key[i] - 32);

Arr[key[i] - 97] = true;

}

}



```
}
```

```
For (int l = 0; l < 26; l++)
```

```
{
```

```
If (arr[l] == false)
```

```
{
```

```
Arr[l] = true;
```

```
Encoded += (char) (l + 65);
```

```
}
```

```
}
```

```
Return encoded;
```

```
}
```



```
Static String cipheredIt(String msg, String encoded)
```

```
{
```

```
String cipher = "";
```

```
For (int l = 0; l < msg.length(); l++)
```

```
{
```

```
If (msg.charAt(i) >= 'a' && msg.charAt(i) <= 'z')
```

```
{
```

```
Int pos = msg.charAt(i) - 97;
```

```
Cipher += encoded.charAt(pos);
```

```
}
```

```
Else if (msg.charAt(i) >= 'A' && msg.charAt(i) <= 'Z')
```

```
{
```

```
Int pos = msg.charAt(i) - 65;
```



```
Cipher += encoded.charAt(pos);
```

```
}
```

```
Else
```

```
{
```

```
Cipher += msg.charAt(i);
```

```
}
```

```
}
```

```
Return cipher;
```

```
}
```

```
Public static void main(String[] args)
```

```
{
```

```
String key;
```



```
Key = "skills";
```

```
System.out.println("Keyword : " + key);
```

```
String encoded = encoder(key.toCharArray());
```

```
String message = "java programming";
```

```
System.out.println("Message before Ciphering : " + message);
```

```
System.out.println("Ciphered Text : " + cipheredIt(message,
```

```
Encoded));
```

```
}
```

```
}
```



Output::

Keyword : skills

Message before Ciphering : java programming

Ciphered Text : FSVS OQNCQSJJEMC

15. Write a program Java Program to implement simple Encryption Decryption with Modulo 26.

Program:

Class Substitution{

String plain;

Substitution (String text)

{

Plain =text;

}

String encrypt()

{

String alpha="abcdefghijklmnopqrstuvwxyz";




```
String sub="mnbvcxzlkhgfsapoiuytrewq";
```

```
String out="";
```

```
For(int i=0;i<plain.length();i++)
```

```
{
```

```
    Char ch=plain.charAt(i);
```

```
    Int p=alpha.indexOf(ch);
```

```
    Char chr=sub.charAt(p);
```

```
    Out =out+chr;
```

```
}
```

```
Return out;
```

```
}
```

```
Public static void main(String[] args) {
```

```
    Subsitution c=new Subsitution("welcome");
```

```
    String out=c.encrypt();
```

```
    System.out.println("the encrypted : "+out);
```

```
}
```

```
}
```

Output::

The encrypted : rcgbsfc



16. Write a program Java Program for XOR Cipher.

Program:

```
Class Xor{
    String plain;
    Char key;

    Xor (String text,char k)
    {
        Plain =text;
        Key =k;
    }

    String encrypt()
    {
        String out="";char ch;

        For(int i=0;i<plain.length();i++)
        {
            Ch =(char)((int)plain.charAt(i)^(int)key);
            Out =out+ch;
        }
        Return out;
    }
}
```



```

    Public static void main(String[] args) {
        Xor c=new Xor("ramusrss",'S');
        String out=c.encrypt();
        System.out.println(" encryption: "+out);
    }
}

```

Output ::

Encryption: !2>& !

17. Write a program Java Program for Latin alphabet cipher.

Program:

```

Class Latincipher{
    String plain;

    Latincipher (String text)
    {
        Plain =text;
    }
}

```



```

String encrypt()
{

    String alpha="abcdefghijklmnopqrstuvwxyz";

    String out="";

    For(int i=0;i<plain.length();i++)
    {
        Char ch=plain.charAt(i);
        Int p=alpha.indexOf(ch)+1;
        Out =out+p+" ";
    }
    Return out;
}

```

```

Public static void main(String[] args) {
    Latincipher c=new Latincipher("ramusrss");
    String out=c.encrypt();
    System.out.println(" encryption : "+out);
}
}

```

Output::



Encryption : 18 1 13 21 19 18 19 19

18. Write a program called Harmonic Sum to compute the sum of a harmonic series, as

Shown below, where $n=50000$. The program shall compute the sum from left-to-right as

Well as from the right-to-left. Are the two sums the same? Obtain the absolute difference

Between these two sums and explain the difference. Which sum is more accurate?

Program:

Public class Main {

Public static void main(String args[]){

Int harmonic = 50000;

Double lr=0, rl=0;

For(int i=1; i<=harmonic; i++){

Lr += (double)(1)/i;



```
Rl += (double)(1)/(harmonic-i+1);
```

```
}
```

```
Double difference = rl-lr;
```

```
System.out.println("left to right = " + lr);
```

```
System.out.println("right to left = " + rl);
```

```
System.out.println("difference is " + difference);
```

```
}
```

```
}
```

Output::

Left to right = 11.397003949278504

Right to left = 11.397003949278519

Difference is 1.4210854715202004E-14



9. Write a program which prompts user for the number of students in a class (a non-

Negative integer), and saves it in an int variable called num Students. It then prompts user

For the grade of each of the students (integer between 0 to 100) and saves them in an int

Array called grades. The program shall then compute and print the average (in double

Rounded to 2 decimal places) and minimum/maximum (in int).

Program:

```
Import java.util.Scanner;
```

```
Public class GradesAverage {
```

```
Private final int LOWEST_GRADE = 0;
```



```
Private final int HIGHEST_GRADE = 100;
```

```
Private int[] grades;
```

```
Private Scanner in;
```

```
Public static void main(String[] args)
```

```
{
```

```
GradesAverage aGradesAverage = new GradesAverage();
```

```
aGradesAverage.in = new Scanner(System.in);
```

```
System.out.print("no.of students: ");
```

```
Int numStudents = aGradesAverage.in.nextInt();
```




```
aGradesAverage.run(numStudents);  
  
}
```

```
Private void run(int numStudents)  
  
{  
  
If (numStudents <= 0) {  
  
    System.out.println("Invalid");  
  
    Return;  
  
}
```

```
Grades = new int[numStudents];
```

```
Double sum = 0;
```

```
Int i = 0;
```



```
While (I < numStudents)

{

    System.out.printf(" grade for student %1$d: ", (i+1));

    Int grade = in.nextInt();

    If ((grade >= LOWEST_GRADE) && (grade <= HIGHEST_GRADE)) {

        Grades[i] = grade;

        Sum    += grade;

        I++;

        Continue;

    }

    System.out.println("Invalid grade, try again...");

}
```



```
        System.out.printf("The average is %1$.2f\n", (sum / numStudents));

    }

}
```

Output::

No.of students: 2

Grade for student 1: 89

Grade for student 2: 98

The average is 93.50

20. Write a Java program to separate 0s on left side and 1s on right side of an array of 0s and

1s in random order.

Import java.util.Arrays;

Import java.util.Scanner;

Public class Main {

Public static void main(String[] args)

{



```
Int arr[] = new int[8];
```

```
Int result[];
```

```
System.out.println("Original Array ");
```

```
System.out.println(Arrays.toString(arr));
```

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

```
System.out.println("Enter the elements of the array: ");
```

```
For(int i=0; i<8; i++)
```

```
{
```

```
    Arr[i]=sc.nextInt();
```

```
}
```

```
Int n = arr.length;
```

```
Result = separate_arr(arr, n);
```

```
System.out.println("New Array ");
```

```
System.out.println(Arrays.toString(result));
```

```
}
```

```
Static int [] separate_arr(int arr[], int n)
```

```
{
```



```
Int count = 0;
```

```
For (int l = 0; l < n; i++) {
```

```
    If (arr[i] == 0)
```

```
        Count++;
```

```
}
```

```
For (int l = 0; l < count; i++)
```

```
    Arr[i] = 0;
```

```
For (int l = count; l < n; i++)
```

```
    Arr[i] = 1;
```

```
Return arr;
```

```
}
```

```
}
```



Output ::

Original Array

[0, 0, 0, 0, 0, 0, 0, 0]

Enter the elements of the array:

1 3 2 4 5 6 9 7

New Array

[1, 1, 1, 1, 1, 1, 1, 1]

21. Write a Java program to add and remove a specific element from an array..

Program:

Import java.util.Scanner;

Public class Delete

{

Public static void main(String[] args)

{

Int n, x, flag = 1, loc = 0;

Scanner s = new Scanner(System.in);



```
System.out.print("no. of elements in array:");
```

```
N = s.nextInt();
```

```
int a[] = new int[n];
```

```
System.out.println("Enter all elements:");
```

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

```
{
```

```
    a[i] = s.nextInt();
```

```
}
```

```
System.out.print("Enter the element want to delete:");
```

```
X = s.nextInt();
```

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

```
{
```

```
    if(a[i] == x)
```

```
{
```



Flag = 1;

Loc = l;

Break;

}

Else

{

Flag = 0;

}

}

If(flag == 1)

{

For(int l = loc+1; l < n; i++)

{



A[i-1] = a[i];

}

System.out.print("After Deleting:");

For (int l = 0; l < n-2; i++)

{

System.out.print(a[i]+",");

}

System.out.print(a[n-2]);

}

Else

{

System.out.println("Element not found");

}

}



}

Output::

No. of elements in array:4

Enter all elements:

2 3 4 5

Enter the element want to delete:4

After Deleting:2,3,5

22. Write a program called CozaLozaWoza which prints the numbers 1 to 110, 11 numbers per line. The program shall print "Coza" in place of the numbers which are multiples of 3,"Loza" for multiples of 5, "Woza" for multiples of 7, "CozaLoza" for multiples of 3 and 5,and so on.

Program:

Public class Main {

Public static void main(String args[]) {

Int I = 1;

While (I <= 110) {



```
Boolean test = false;
```

```
If (l % 3 == 0) {
```

```
    System.out.print("coza");
```

```
    Test = true;
```

```
}
```

```
If (l % 5 == 0) {
```

```
    System.out.print("loza");
```

```
    Test = true;
```

```
}
```

```
If (l % 7 == 0) {
```

```
    System.out.print("woza");
```

```
    Test = true;
```

```
}
```



```
If (!test) {
```

```
    System.out.print(i);
```

```
}
```

```
System.out.print(" ");
```

```
If (l % 11 == 0) {
```

```
    System.out.println();
```

```
}
```

```
l++;
```

```
}
```

```
}
```

```
}
```

Output ::



1 2 coza 4 loza coza woza 8 coza loza 11
Coza 13 woza cozalozza 16 17 coza 19 loza cozawoza 22
23 coza loza 26 coza woza 29 cozalozza 31 32 coza
34 lozawoza coza 37 38 coza loza 41 cozawoza 43 44
Cozalozza 46 47 coza woza loza coza 52 53 coza loza
Woza coza 58 59 cozalozza 61 62 cozawoza 64 loza coza
67 68 coza lozawoza 71 coza 73 74 cozalozza 76 woza
Coza 79 loza coza 82 83 cozawoza loza 86 coza 88
89 cozalozza woza 92 coza 94 loza coza 97 woza coza
Loza 101 coza 103 104 cozalozawoza 106 107 coza 109 loza

