

ASSIGNMENT 1

CODE

//supplement class contains all the functions

```
import java.util.*;

public class supplement {

    static double divide(int x,int y)
    {
        return (double)x/y;
    }
    static int findGCD(int a,int b)
    {
        if(b==0)
            return a;
        return findGCD(b,a%b);
    }
    static long lcm(int a,int b)
    {
        int num=findGCD(a, b);
        long l=(a*b)/num;
        return l;
    }
    static long power(int a,int b){
        long l=1;
        for(int i=0;i<b;i++)
        {
            l=l*a;
        }
        return l;
    }
    static int max(int a,int b)
    {
        if(a>b)
            return a;
        return b;
    }
    static int min(int a,int b)
    {
```

```
        if(a<b)
            return a;
        return b;
    }
    static int abs(int a)
    {
        if(a<0)
        {
            return (-a);
        }
        return a;
    }
    static long factorial(int a)
    {
        long ans=1;
        for(int i=1;i<=a;i++)
        {
            ans=ans*i;
        }
        return ans;
    }
    static int sum(int arr[]){
        int l=arr.length;
        int sum=0;
        for(int i=0;i<l;i++)
        {
            sum+=arr[i];
        }
        return sum;
    }
    static int sum(int a){
        int temp=a;
        int sum=0;
        while(temp!=0)
        {
            sum+=(temp%10);
            temp=temp/10;
        }
        return sum;
    }
}
```

```
static int sqrt(int a){
    int i=0;
    for(i=0;i<a;i++)
    {
        if(i*i>=a)
            break;

    }
    return i;
}
static boolean isPrime(int a)
{
    for(int i=2;i<(a/2);i++)
    {
        if(a%i==0)
            return false;
    }
    return true;
}
static boolean isLeapYear(int a)
{
    if(a%100==0)
    {
        if(a%400==0)
            return true;
        else
            return false;
    }
    else{
        if(a%4==0)
            return true;
        else
            return false;
    }
}
static boolean isPalindrome(int a)
{
    String s="";
    s=s+a;
    StringBuffer sb = new StringBuffer(s);
```

```
        sb.reverse();
        String s1=sb.toString();
        if(s1.equals(s))
            return true;

        return false;
    }
    static boolean isArmstrong(int a)
    {
        int temp=a;
        long sum=0;
        while(temp!=0)
        {
            sum+=Math.pow((temp%10),3);
            temp=temp/10;
        }
        if(sum==a)
            return true;
        return false;
    }
    static int ArithmeticSequenceSum(int a, int d, int n)
    {
        double sum=(double)n/2*((2*a)+(n-1)*d);
        return (int)sum;
    }
    static int GeometricSequenceSum(int a, int r,int n)
    {
        int sum=a*((int)(Math.pow(r,n)-1)/(r-1));
        return sum;
    }
    static int linearSearch(int arr[],int num)
    {
        int size=arr.length;
        for(int i=0;i<size;i++)
        {
            if(arr[i]==num)
                return i;
        }
        return -1;
    }
}
```

```
static int[] reverseArray(int arr[])
{
    int size=arr.length;
    int arr1[]=new int[size];
    for(int i=size-1;i>=0;i--)
    {
        arr1[size-1-i]=arr[i];
    }
    return arr1;
}

static int maximumAbsoluteDifference(int arr[])
{
    int size=arr.length;
    int min=arr[0];
    int max=arr[0];
    for(int i=0;i<size;i++)
    {
        if(min>arr[i])
            min=arr[i];
        if(max<arr[i])
            max=arr[i];
    }
    return Math.abs(max-min);
}

static int[][] additionMatrices(int arr1[][],int arr2[][])
{
    int rows=arr1.length;
    int col=arr1[0].length;
    int sum[][]=new int[rows][col];
    for(int i=0;i<rows;i++)
    {
        for(int j=0;j<col;j++)
        {
            sum[i][j]=arr1[i][j]+arr2[i][j];
        }
    }
    return sum;
}

static int[][] multiplicationMatrices(int m1[][],int m2[][])
{

```

```
int r1=m1.length;
int c1=m1[0].length;
int r2=m2.length;
int c2=m2[0].length;
if(c1!=r2)
{
    System.out.println("Wrong input");
    System.exit(0);
}
int ans[][]=new int[r1][c2];
for(int i=0;i<r1;i++)
{
    for(int j=0;j<c2;j++)
    {
        int sum=0;
        for(int k=0;k<r2;k++)
        {
            sum=sum+m1[i][k]*m2[k][j];
        }
        ans[i][j]=sum;
    }
}
return ans;
}

//STRING PROBLEMS
static boolean isPalindrome(String s)
{
    StringBuffer sb=new StringBuffer(s);
    sb.reverse();
    String s1=sb.toString();
    if(s1.equals(s))
        return true;
    return false;
}

static boolean Numeric(String s)
{
    for(int i=0;i<s.length();i++)
    {
        if(Character.isDigit(s.charAt(i))) {
```

```
        continue;
    }
    else
        return false;
    }
    return true;
}

static boolean chkEqualString(String s1,String s2)
{
    if(s1.equals(s2))
        return true;
    return false;
}

static String sortString(String s)
{
    int l=s.length();
    int arr[]=new int[l];
    for(int i=0;i<l;i++)
    {
        arr[i]=s.charAt(i);
    }
    Arrays.sort(arr);
    String s1="";
    for(int i=0;i<l;i++)
    {
        s1+=(char)arr[i];
    }
    return s1;
}

static boolean checkAnagram(String s1,String s2)
{
    String s3=sortString(s1);
    String s4=sortString(s2);
    if(s3.equals(s4))
        return true;

    return false;
}

static int countCharacters(String s)
{

```

```
HashMap<Integer,Integer> map=new HashMap<> ();
int c=0;
for(int i=0;i<s.length();i++)
{
    int n=s.charAt(i);
    if(map.containsKey(n))
    {
        int n1=map.get(n);
        n1++;
        map.replace(n, n1-1, n1);
    }
    else
        map.put(n,1);
}
int count=0;
for(int i=0;i<s.length();i++)
{
    int n2=map.get((int) (s.charAt(i)));
    if(n2==1)
    {
        count++;
    }
}
return count;
}
static void conversion(String s)
{
    int n=Integer.parseInt(s);
    int temp=n;
    int bin=0,c=0;
    while(temp!=0) {
        int rem=temp%2;
        bin=bin+(int) (Math.pow(10,c)*rem);
        c++;
        temp=temp/2;
    }
    System.out.println(bin);
    temp=bin;
```


Satyam Tripathi
202151141

```
        c=0;
        int deci=0;
        while(temp!=0){
            int rem=temp%10;
            deci=deci+(int) (Math.pow(2,c))*rem;
        }
        System.out.println(deci);
    }
    static void pattern()
    {
        int n=5;
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<=i;j++)
                System.out.print("*");
            System.out.println();
        }
    }
}
```

//Main Class for calling arithmetic functions using switch case

```
import java.util.*;

public class arithmetic {

    public static void main(String[] args) {

        System.out.println("1. Division 2. gcd 3. lcm 4. power 5. max 6. min 7. abs 8. factorial 9. sum 10.sumofdigits 11.sqrt 12. isPrime 13. isLeapYear 14. isPalindrome 15. isArmstrong 16. APsum 17. GPsum");

        Scanner sc=new Scanner(System.in);

        int ch=sc.nextInt();

        int ans=0;

        supplement x1=new supplement();

        int a=0,b=0,c=0;

        String s="";

        int arr[];

        switch(ch)

        {

            case 1:

                a=sc.nextInt();

                b=sc.nextInt();

                double div=x1.divide(a,b); //division

                System.out.println(a+"/"+b+" = "+div);

                break;

            case 2:

                a=sc.nextInt();

                b=sc.nextInt();

                int gcdans=x1.findGCD(a,b);

                System.out.println(a+" "+b+" gcd is = "+gcdans);

                break;

            case 3:

                a=sc.nextInt();

                b=sc.nextInt();

                long lcm=x1.lcm(a,b);

                System.out.println(a+" "+b+" lcm is = "+lcm);

                break;

            case 4:

                a=sc.nextInt();

                b=sc.nextInt();

                long power=x1.power(a,b);
```

```
System.out.println(a+" raised to power "+b+" is = "+power);
break;
case 5:
a=sc.nextInt();
b=sc.nextInt();
int max=x1.max(a,b);
System.out.println("Max no. is "+max);
break;
case 6:
a=sc.nextInt();
b=sc.nextInt();
int min=x1.min(a,b);
System.out.println("Min no. is "+min);
break;
case 7:
a=sc.nextInt();
int abs=x1.abs(a);
System.out.println(abs);
break;
case 8:
a=sc.nextInt();
long fact=x1.factorial(a);
System.out.println("Factorial of "+a+" is "+fact);
break;
case 9:
System.out.println("Enter length of array");
int len=sc.nextInt();
arr=new int[len];
for(int i=0;i<len;i++)
{
    arr[i]=sc.nextInt();
}
int sum=x1.sum(arr);
System.out.println("Sum of the array is "+sum);
break;
case 10:
a=sc.nextInt();
sum=x1.sum(a);
System.out.println("Sum of digits of "+a+" is "+sum);
break;
```

```
        case 11:
            a=sc.nextInt();
            int sqr=x1.sqrt(a);
            System.out.println(sqr);
            break;
        case 12:
            a=sc.nextInt();
            if(x1.isPrime(a))
                System.out.println(a+" is prime");
            else
                System.out.println(a+" is not prime");

            break;
        case 13:
            a=sc.nextInt();
            if(x1.isLeapYear(a))
                System.out.println(a+" is a leap year");
            else
                System.out.println(" is not a leap year");
            break;
        case 14:
            a=sc.nextInt();
            if(x1.isPalindrome(a))
                System.out.println(a+" is a Palindrome");
            else
                System.out.println(a+" is not a palindrome");
            break;
        case 15:
            a=sc.nextInt();
            if(x1.isArmstrong(a))
                System.out.println(a+" is a armstrong number");
            else
                System.out.println(a+" is not a palindrome");
            break;
        case 16:
            a=sc.nextInt();
            b=sc.nextInt();
            c=sc.nextInt();
            System.out.println("Sum of AP is "+x1.ArithmeticSequenceSum(a,
b, c));
```

Satyam Tripathi
202151141

```
        break;
        case 17:
            a=sc.nextInt();
            b=sc.nextInt();
            c=sc.nextInt();
            System.out.println("Sum of GP is "+x1.GeometricSequenceSum(a,
b, c));
            break;
        }
    }
}
```

//Main class for calling Array Problems

```
import java.util.Scanner;

public class arrayfunc {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        supplement x1=new supplement();
        int n=sc.nextInt();
        System.out.println("Enter length of array");
        int arr[]=new int[n];
        System.out.println("Enter elements of array");
        for(int i=0;i<n;i++)
        {
            arr[i]=sc.nextInt();
        }
        System.out.println("1. Linear Search 2. Reverse Array 3. Find Max
absolute difference");
        int ch=sc.nextInt();
        switch(ch)
        {
            case 1:
                System.out.println("Enter number to search");
                int num=sc.nextInt();
                System.out.println(x1.linearSearch(arr, num));
                break;
            case 2:
                int a[]=x1.reverseArray(arr);
                System.out.println(arr.toString());
                break;
            case 3:
                System.out.println(x1.maximumAbsoluteDifference(arr));
                break;
        }
    }
}
```

//Main class for calling Matrix Problem functions

```
import java.util.Scanner;

public class matrixproblems {
    public static void main(String[] args) {
        supplement x1=new supplement();
        Scanner sc=new Scanner(System.in);
        int arr1[][] ,arr2[][];
        System.out.println("1. Addition of Matrices  2. Multiplication of matrix");
        int ch=sc.nextInt();
        switch(ch){
            case 1:
                System.out.println("Enter Dimensions of both matrix");
                int r=sc.nextInt();
                int c=sc.nextInt();

                arr1=new int[r][c];
                arr2=new int[r][c];
                System.out.println("Enter First matrix");
                int ans[][]=new int[r][c];
                for(int i=0;i<r;i++)
                {
                    for(int j=0;j<c;j++)
                    {
                        arr1[i][j]=sc.nextInt();
                    }
                }
                System.out.println("Enter second matrix");
                for(int i=0;i<r;i++)
                {
                    for(int j=0;j<c;j++)
                    {
                        arr2[i][j]=sc.nextInt();
                    }
                }
                System.out.println();
            }
        }
    }
}
```

```
ans=x1.additionMatrices(arr1,arr2);
for(int i=0;i<r;i++)
{
    for(int j=0;j<c;j++)
    {
        System.out.print(ans[i][j]+" ");

    }
    System.out.println();

}
break;
case 2:
System.out.println("Enter Dimensions of first matrix");
int r1=sc.nextInt();
int c1=sc.nextInt();
System.out.println("Enter Dimensions of second matrix");
int r2=sc.nextInt();
int c2=sc.nextInt();
arr1=new int[r1][c1];
arr2=new int[r2][c2];
System.out.println("Enter First matrix");
ans=new int[r1][c2];
for(int i=0;i<r1;i++)
{
    for(int j=0;j<c1;j++)
    {
        arr1[i][j]=sc.nextInt();

    }
}
System.out.println("Enter second matrix");
for(int i=0;i<r2;i++)
{
    for(int j=0;j<c2;j++)
    {
        arr2[i][j]=sc.nextInt();

    }
}
}
```


Satyam Tripathi
202151141

```
        System.out.println();  
        ans=x1.multiplicationMatrices(arr1, arr2);  
        for(int i=0;i<r1;i++)  
        {  
            for(int j=0;j<c2;j++)  
            {  
                System.out.print(ans[i][j]+" ");  
            }  
            System.out.println();  
        }  
    }  
}
```

//Main class for calling String Problem functions

```
import java.util.Scanner;

import javax.lang.model.util.ElementScanner14;

public class stringFunctions {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println(" 1. Check Palindrome 2. Check if String has  
only numbers 3. Check if two Strings are equal 4. Sort the characters in  
String 5. Check whether 2 strings are anagram 6. Count single occurring 7.  
Convert Decimal to Binary and vice-versa");
        int ch=sc.nextInt();
        String temp=sc.nextLine();
        System.out.println("Enter a String");
        String s=sc.nextLine();
        supplement x1 =new supplement();
        String s1="";
        switch(ch)
        {
            case 1:
                if(x1.isPalindrome(s))
                    System.out.println(s+" is a Palindrome");
                else{
                    System.out.println(s+" is not a palindrome");
                }
                break;
            case 2:
                if(x1.Numeric(s))
                    System.out.println(s+" is numeric");
                else{
                    System.out.println(s+" is not numeric");
                }
                break;
            case 3:
                System.out.println("Enter another String");
                s1=sc.nextLine();
                if(x1.chkEqualString(s,s1))
                    System.out.println(s+" and "+s1+" are equal");
```

Satyam Tripathi
202151141

```
        else{
            System.out.println(s+" and "+s1+" are not equal");
        }
        break;
        case 4:
            s1=x1.sortString(s);
            System.out.println(s);
            break;
        case 5:
            System.out.println("Enter another String");
            s1=sc.nextLine();
            if(x1.checkAnagram(s,s1))
                System.out.println(s+" and "+s1+" are anagram");
            else{
                System.out.println(s+" and "+s1+" are not anagram");
            }
            break;
        case 6:
            int count = x1.countCharacters(s);
            System.out.println(count);
            break;
        case 7:
            x1.conversion(s);
            break;
    }
}
}
```

Satyam Tripathi
202151141

//Main class to call Pattern function

```
public class pattern {  
    public static void main(String[] args) {  
        supplement x1=new supplement();  
        x1.pattern();  
    }  
}
```

Satyam Tripathi

202151141

OUTPUT

Sum of AP Sequence

```
1. Division 2. gcd 3. lcm 4. power 5. max 6. min 7. abs 8. factorial 9. sum 10.sumofdigits 11.sqrt 12. isPrime 13. isLeapYe
ar 14. isPalindrome 15. isArmstrong 16. APsum 17. GPsum
16
2 4 4
Sum of AP is 32
```

Check if a number is Prime or not

```
1. Division 2. gcd 3. lcm 4. power 5. max 6. min 7. abs 8. factorial 9. sum 10.sumofdigits 11.sqrt 12. isPrime 13. isLeapYe
ar 14. isPalindrome 15. isArmstrong 16. APsum 17. GPsum
12
111
111 is prime
```

Linear Search

```
5
Enter length of array
Enter elements of array
4 1 7 3 8
1. Linear Search 2. Reverse Array 3. Find Max absolute difference
1
Enter number to search
7
2
```

Satyam Tripathi

202151141

Multiplication of matrices

```
1. Addition of Matrices  2. Multiplication of matrix
2
Enter Dimensions of first matrix
3 3
Enter Dimensions of second matrix
3 2
Enter First matrix
1 0 0
0 1 0
0 0 1
Enter second matrix
2 3
4 5
6 7

2 3
4 5
6 7
```

Star Pattern

```
*
**
***
****
*****
```

Sort characters in string

```
4
Enter a String
hello
ehllo
```