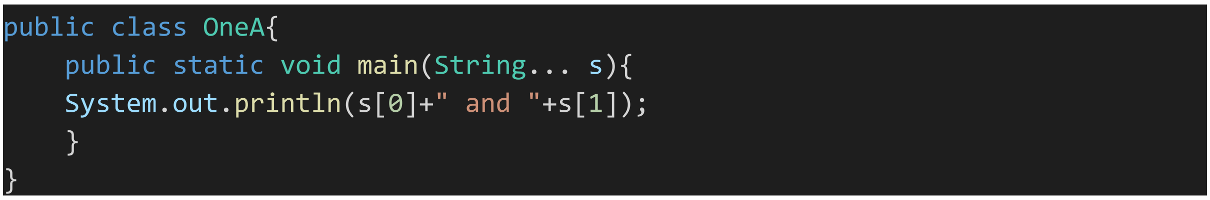
1. a) Implement the following programs using command line arguments.

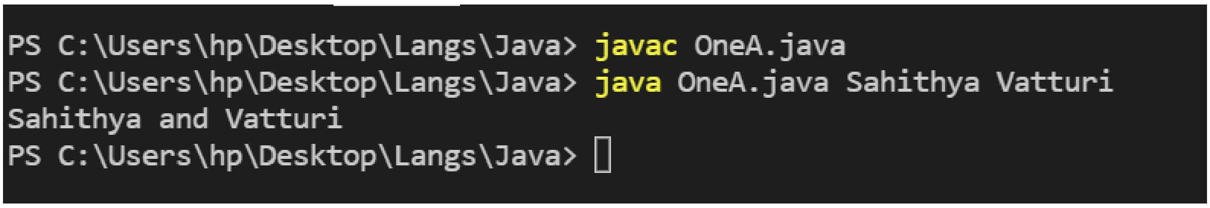
i. Accept two strings from the user and print it on console with concatenation of "and" in the middle of the strings.

Code:



```
public class OneA{
    public static void main(String... s){
    System.out.println(s[0]+" and "+s[1]);
    }
}
```

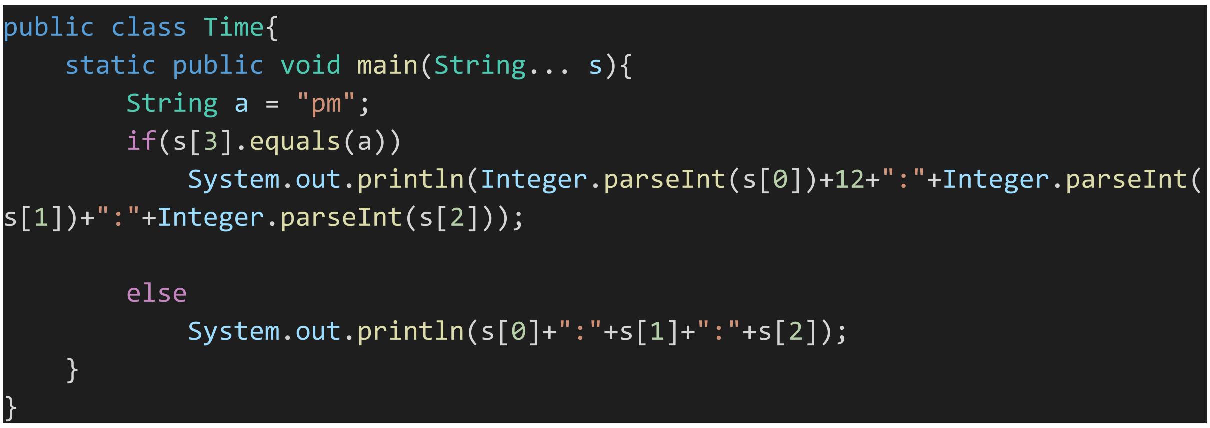
Input & Output:



```
PS C:\Users\hp\Desktop\Langs\Java> javac OneA.java
PS C:\Users\hp\Desktop\Langs\Java> java OneA.java Sahithya Vatturi
Sahithya and Vatturi
PS C:\Users\hp\Desktop\Langs\Java> 
```

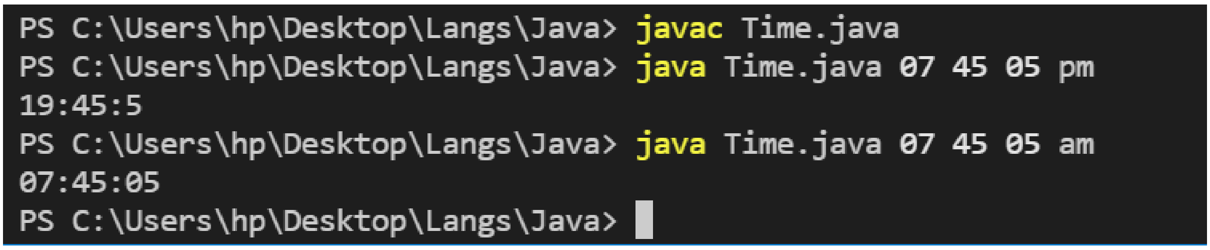
ii. Accept 12-hour time zone and convert into its corresponding 24-hour time zone. Note: Accept hours, minutes and seconds separately from the user (e.g., 07 05 45 PM should be displayed as 19:05:45).

CODE:



```
public class Time{
    static public void main(String... s){
        String a = "pm";
        if(s[3].equals(a))
            System.out.println(Integer.parseInt(s[0])+12+":"+Integer.parseInt(
s[1])+":"+Integer.parseInt(s[2]));

        else
            System.out.println(s[0]+":"+s[1]+":"+s[2]);
    }
}
```

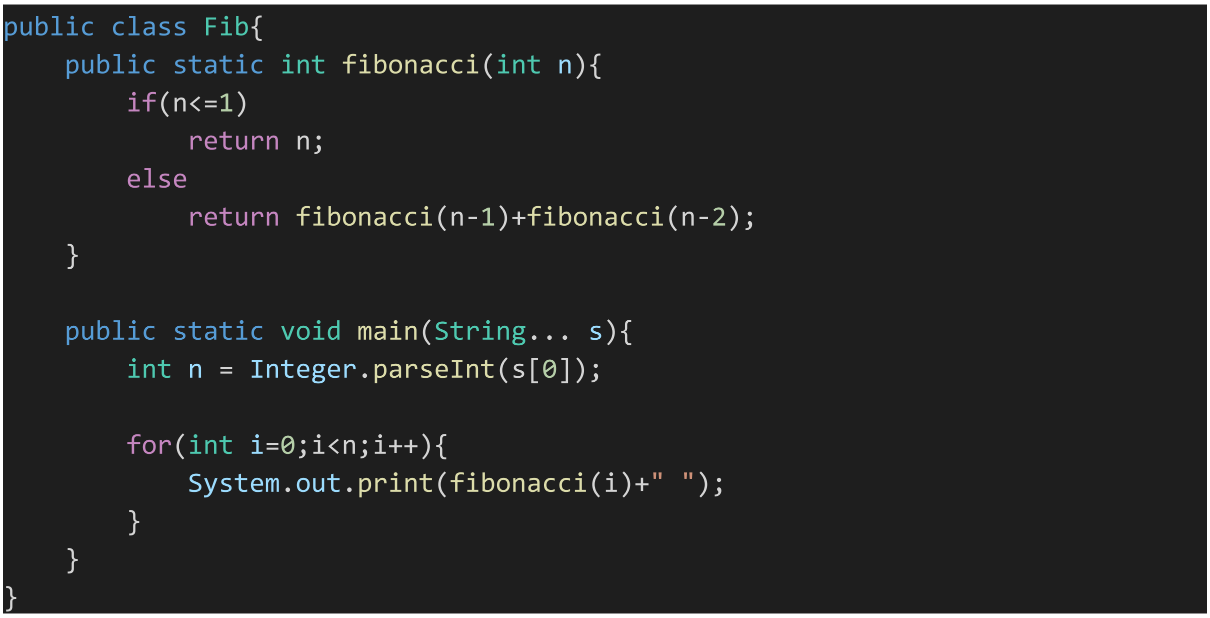
INPUT & OUTPUT:



```
PS C:\Users\hp\Desktop\Langs\Java> javac Time.java
PS C:\Users\hp\Desktop\Langs\Java> java Time.java 07 45 05 pm
19:45:5
PS C:\Users\hp\Desktop\Langs\Java> java Time.java 07 45 05 am
07:45:05
PS C:\Users\hp\Desktop\Langs\Java> 
```

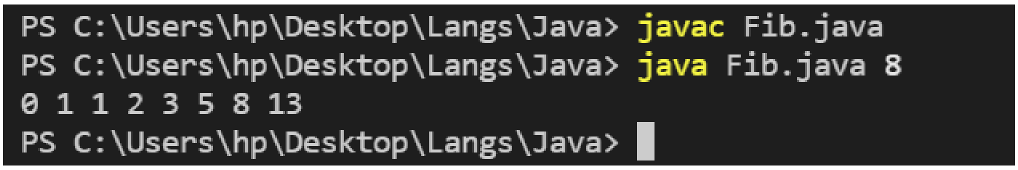
iii. Accept a number "n" and print the list of "n" Fibonacci terms recursively.

CODE:



```
public class Fib{
    public static int fibonacci(int n){
        if(n<=1)
            return n;
        else
            return fibonacci(n-1)+fibonacci(n-2);
    }

    public static void main(String... s){
        int n = Integer.parseInt(s[0]);

        for(int i=0;i<n;i++){
            System.out.print(fibonacci(i)+" ");
        }
    }
}
```

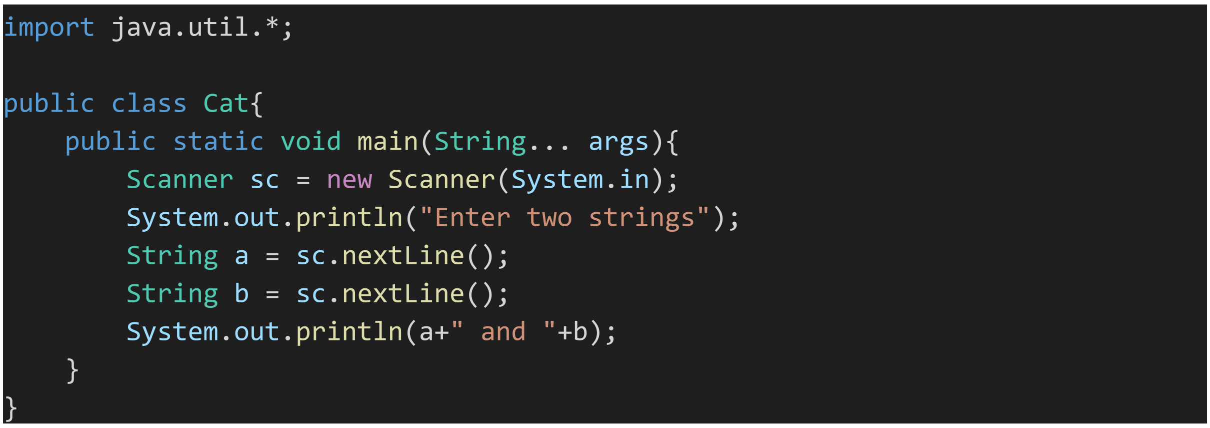
INPUT & OUTPUT:



```
PS C:\Users\hp\Desktop\Langs\Java> javac Fib.java
PS C:\Users\hp\Desktop\Langs\Java> java Fib.java 8
0 1 1 2 3 5 8 13
PS C:\Users\hp\Desktop\Langs\Java>
```

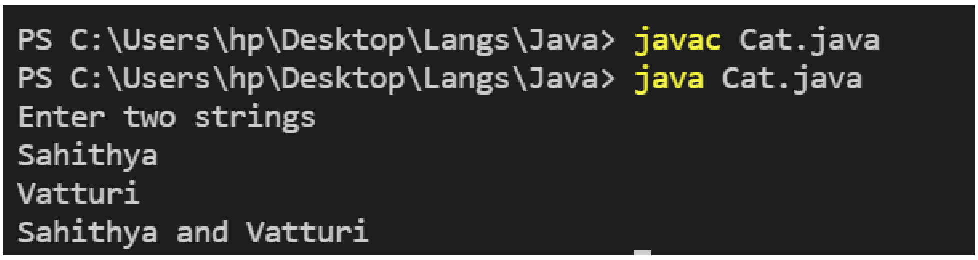
b) Perform the above programs using Scanner class.

Concatenation Code:



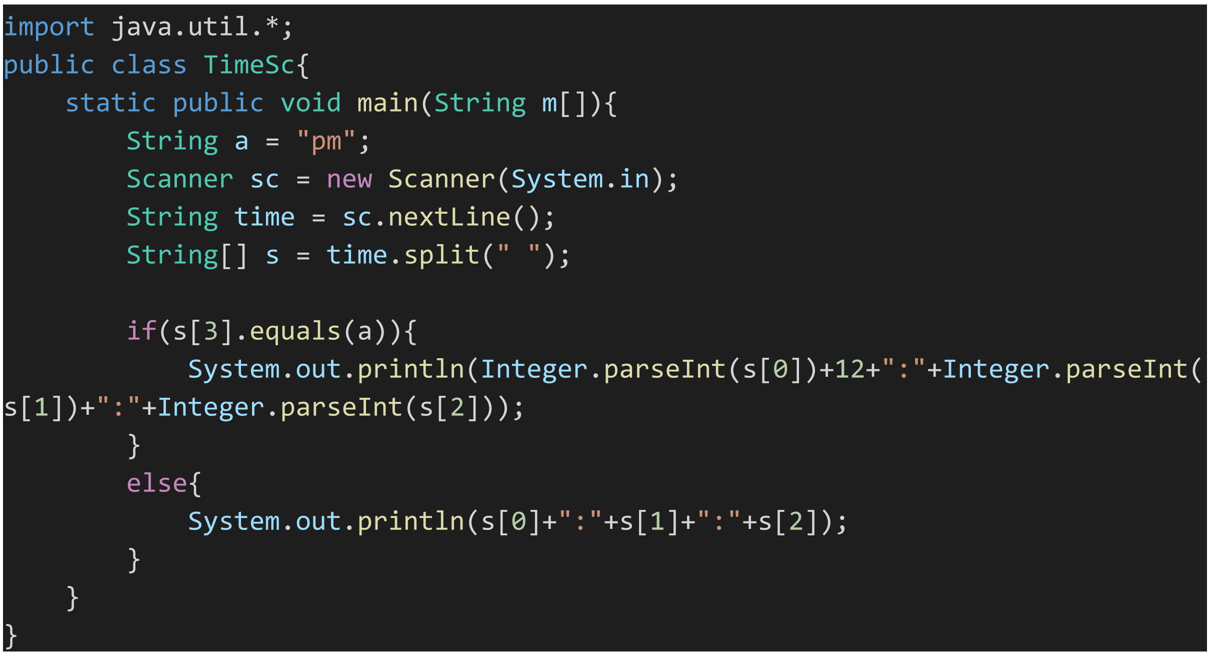
```
import java.util.*;

public class Cat{
    public static void main(String... args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter two strings");
        String a = sc.nextLine();
        String b = sc.nextLine();
        System.out.println(a+" and "+b);
    }
}
```

INPUT & OUTPUT:



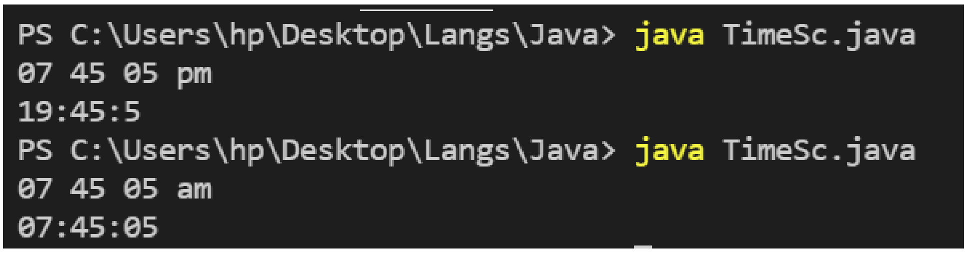
```
PS C:\Users\hp\Desktop\Langs\Java> javac Cat.java
PS C:\Users\hp\Desktop\Langs\Java> java Cat.java
Enter two strings
Sahithya
Vatturi
Sahithya and Vatturi
```

12 Hour to 24 Hour format Code:



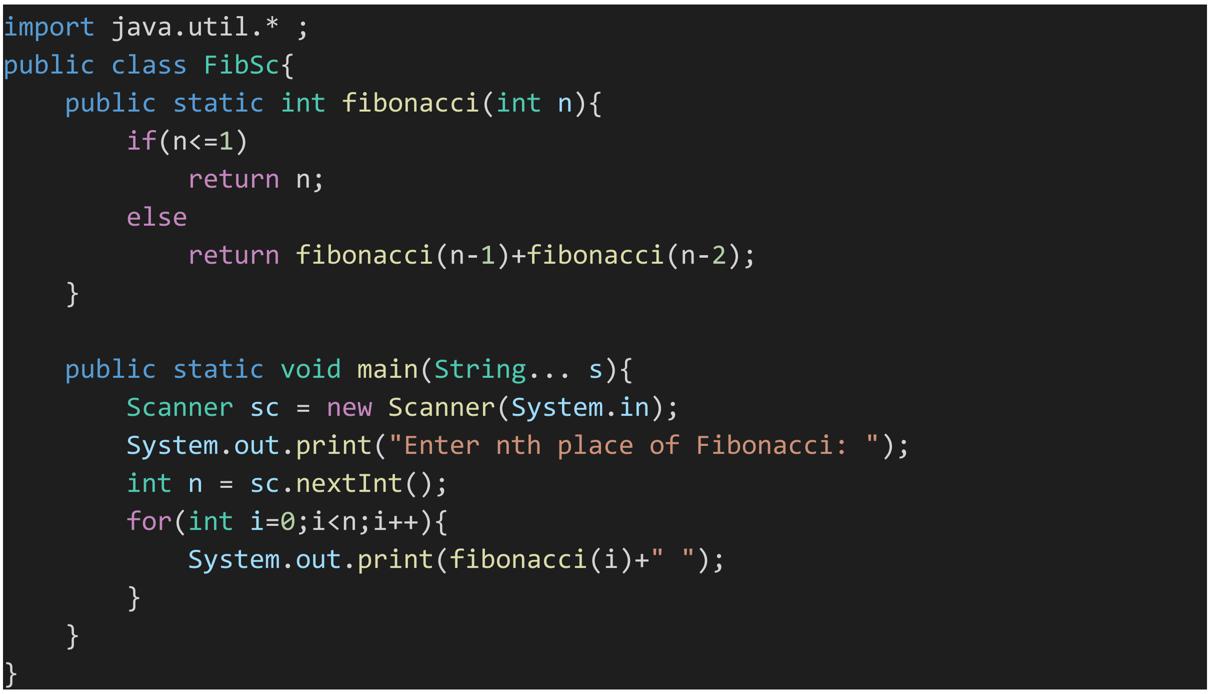
```
import java.util.*;
public class TimeSc{
    static public void main(String m[]){
        String a = "pm";
        Scanner sc = new Scanner(System.in);
        String time = sc.nextLine();
        String[] s = time.split(" ");

        if(s[3].equals(a)){
            System.out.println(Integer.parseInt(s[0])+12+":"+Integer.parseInt(
s[1])+":"+Integer.parseInt(s[2]));
        }
        else{
            System.out.println(s[0]+":"+s[1]+":"+s[2]);
        }
    }
}
```

INPUT & OUTPUT:



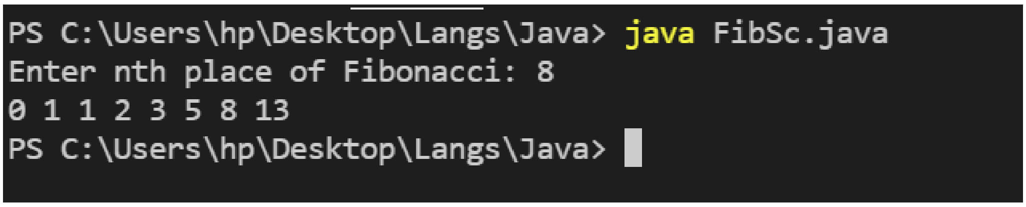
```
PS C:\Users\hp\Desktop\Langs\Java> java TimeSc.java
07 45 05 pm
19:45:5
PS C:\Users\hp\Desktop\Langs\Java> java TimeSc.java
07 45 05 am
07:45:05
```

Fibonacci Series Code:



```
import java.util.* ;
public class FibSc{
    public static int fibonacci(int n){
        if(n<=1)
            return n;
        else
            return fibonacci(n-1)+fibonacci(n-2);
    }

    public static void main(String... s){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter nth place of Fibonacci: ");
        int n = sc.nextInt();
        for(int i=0;i<n;i++){
            System.out.print(fibonacci(i)+" ");
        }
    }
}
```

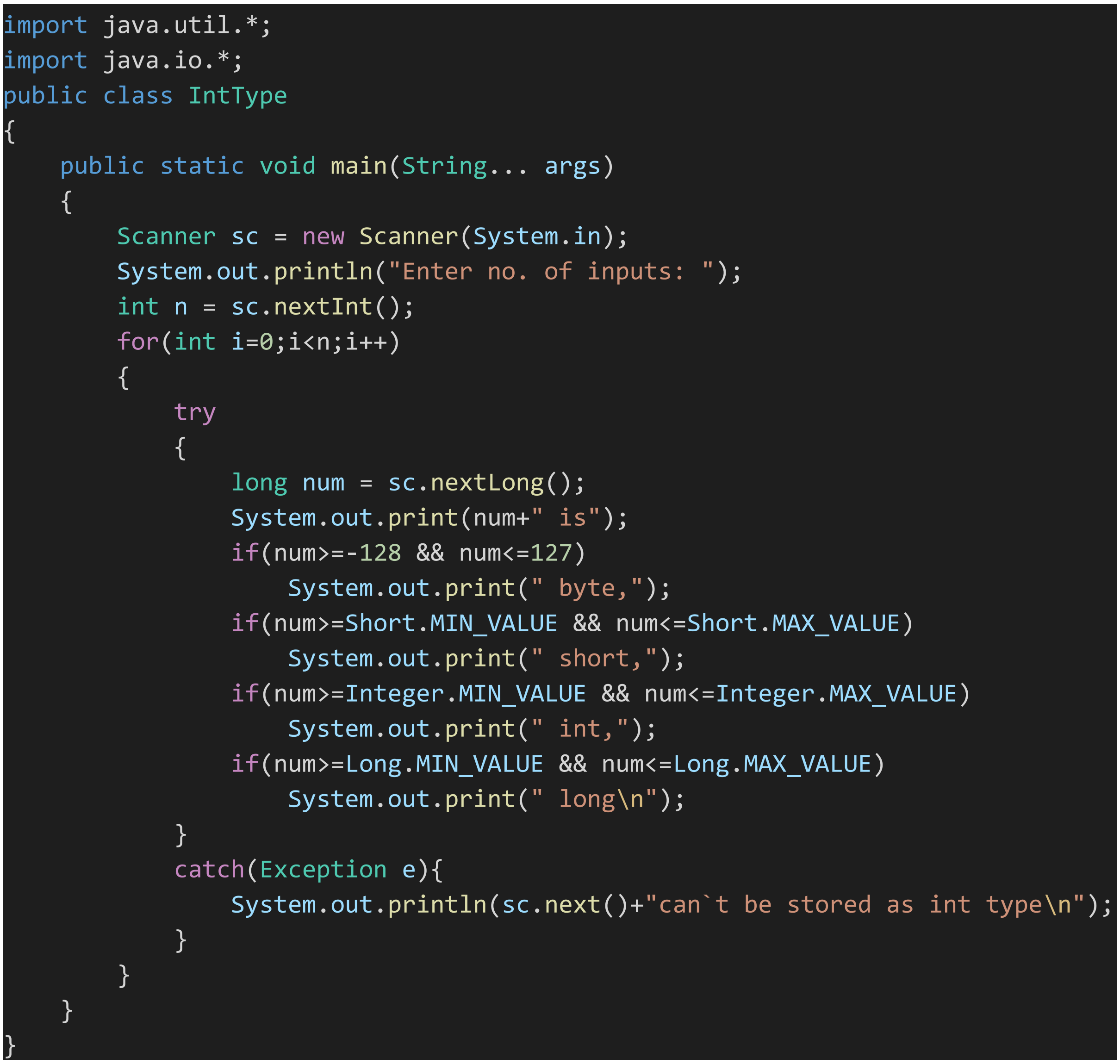
INPUT & OUTPUT:



```
PS C:\Users\hp\Desktop\Langs\Java> java FibSc.java
Enter nth place of Fibonacci: 8
0 1 1 2 3 5 8 13
PS C:\Users\hp\Desktop\Langs\Java> 
```

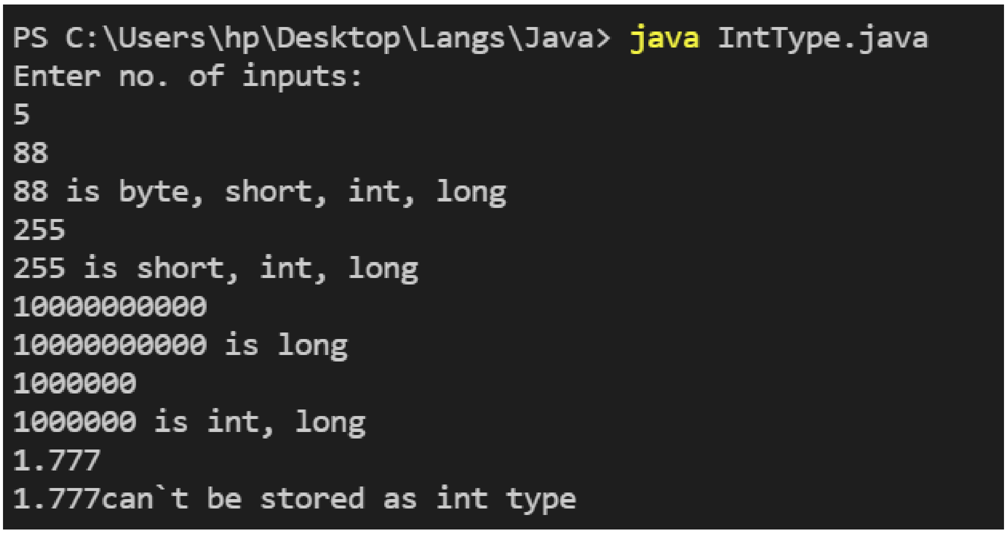
2 a) Write a program that accepts the set of inputs from the user of various integer data types and determines the primitive data type that is capable of properly storing that input.

CODE:



```
import java.util.*;
import java.io.*;
public class IntType
{
    public static void main(String... args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter no. of inputs: ");
        int n = sc.nextInt();
        for(int i=0;i<n;i++)
        {
            try
            {
                long num = sc.nextLong();
                System.out.print(num+" is");
                if(num>=-128 && num<=127)
                    System.out.print(" byte,");
                if(num>=Short.MIN_VALUE && num<=Short.MAX_VALUE)
                    System.out.print(" short,");
                if(num>=Integer.MIN_VALUE && num<=Integer.MAX_VALUE)
                    System.out.print(" int,");
                if(num>=Long.MIN_VALUE && num<=Long.MAX_VALUE)
                    System.out.print(" long\n");
            }
            catch(Exception e){
                System.out.println(sc.next()+"can`t be stored as int type\n");
            }
        }
    }
}
```

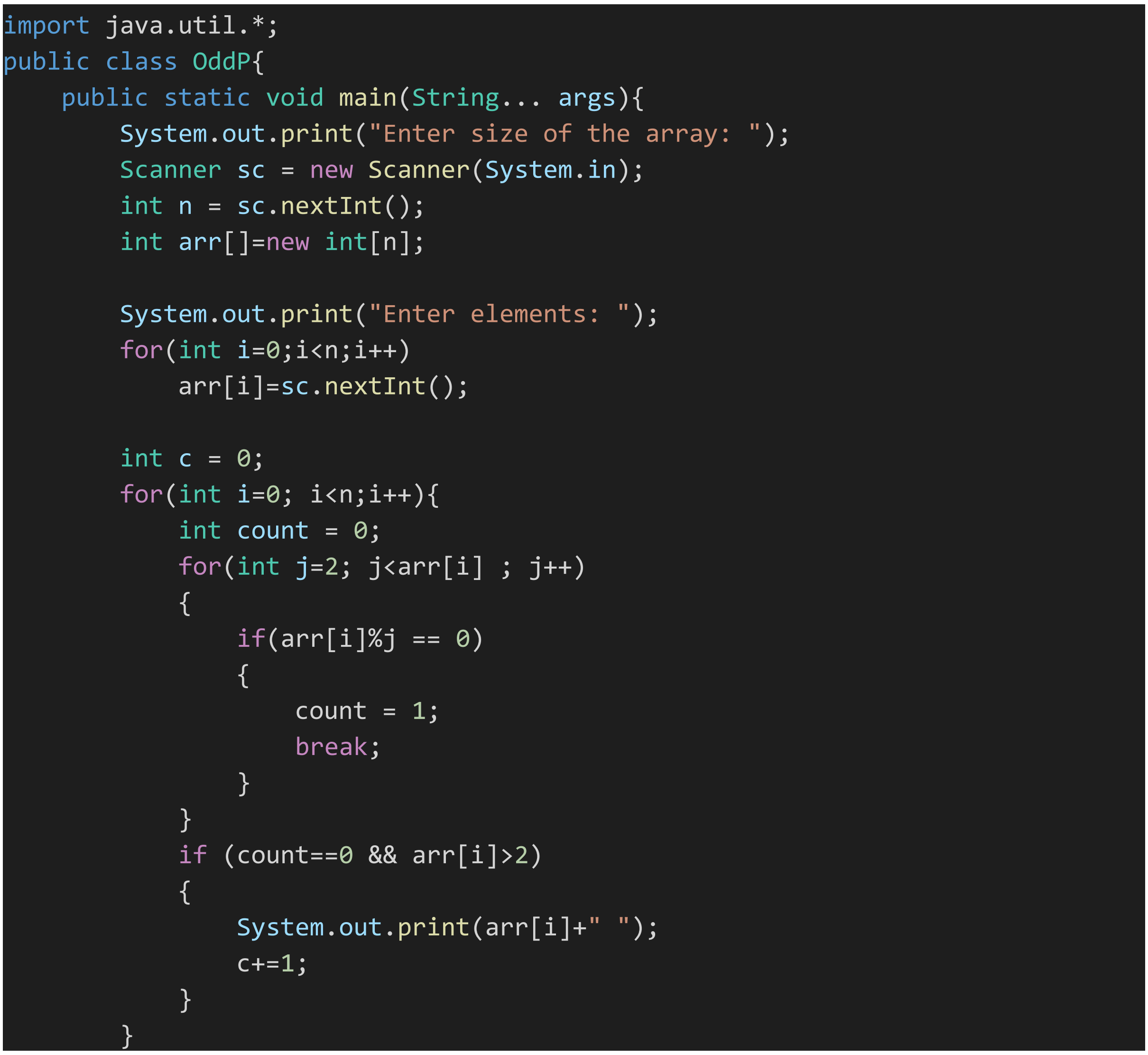
INPUT & OUTPUT:



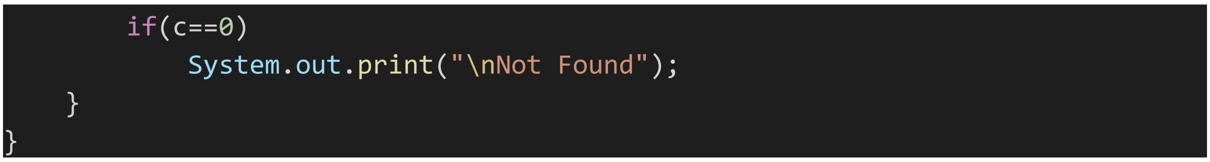
```
PS C:\Users\hp\Desktop\Langs\Java> java IntType.java
Enter no. of inputs:
5
88
88 is byte, short, int, long
255
255 is short, int, long
10000000000
10000000000 is long
1000000
1000000 is int, long
1.777
1.777can`t be stored as int type
```

b) Write a program that accepts an array of integers and print those which are both odd and prime. If no such element in that array print “Not found”.

CODE:

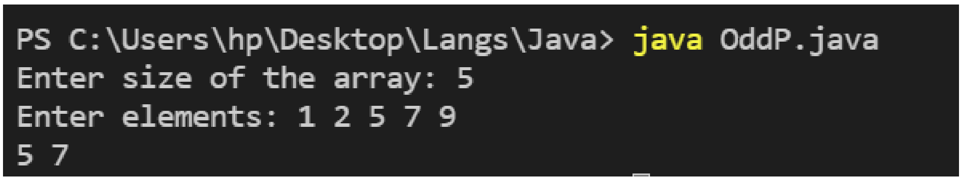


```
import java.util.*;
public class OddP{
    public static void main(String... args){
        System.out.print("Enter size of the array: ");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int arr[]=new int[n];

        System.out.print("Enter elements: ");
        for(int i=0;i<n;i++)
            arr[i]=sc.nextInt();

        int c = 0;
        for(int i=0; i<n;i++){
            int count = 0;
            for(int j=2; j<arr[i] ; j++)
            {
                if(arr[i]%j == 0)
                {
                    count = 1;
                    break;
                }
            }
            if (count==0 && arr[i]>2)
            {
                System.out.print(arr[i]+" ");
                c+=1;
            }
        }
```



```
        if(c==0)
            System.out.print("\nNot Found");
    }
}
```

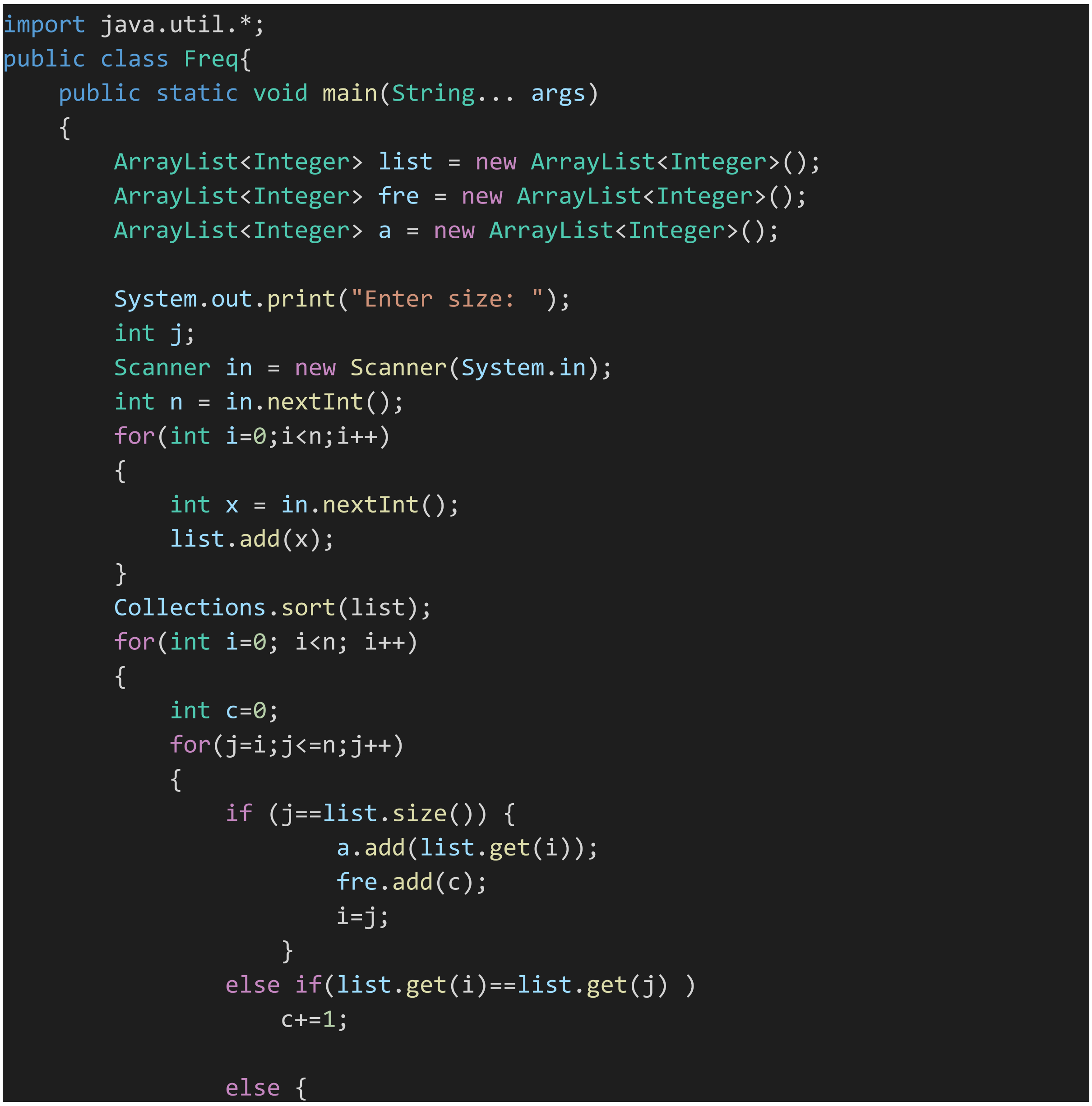
INPUT & OUTPUT:



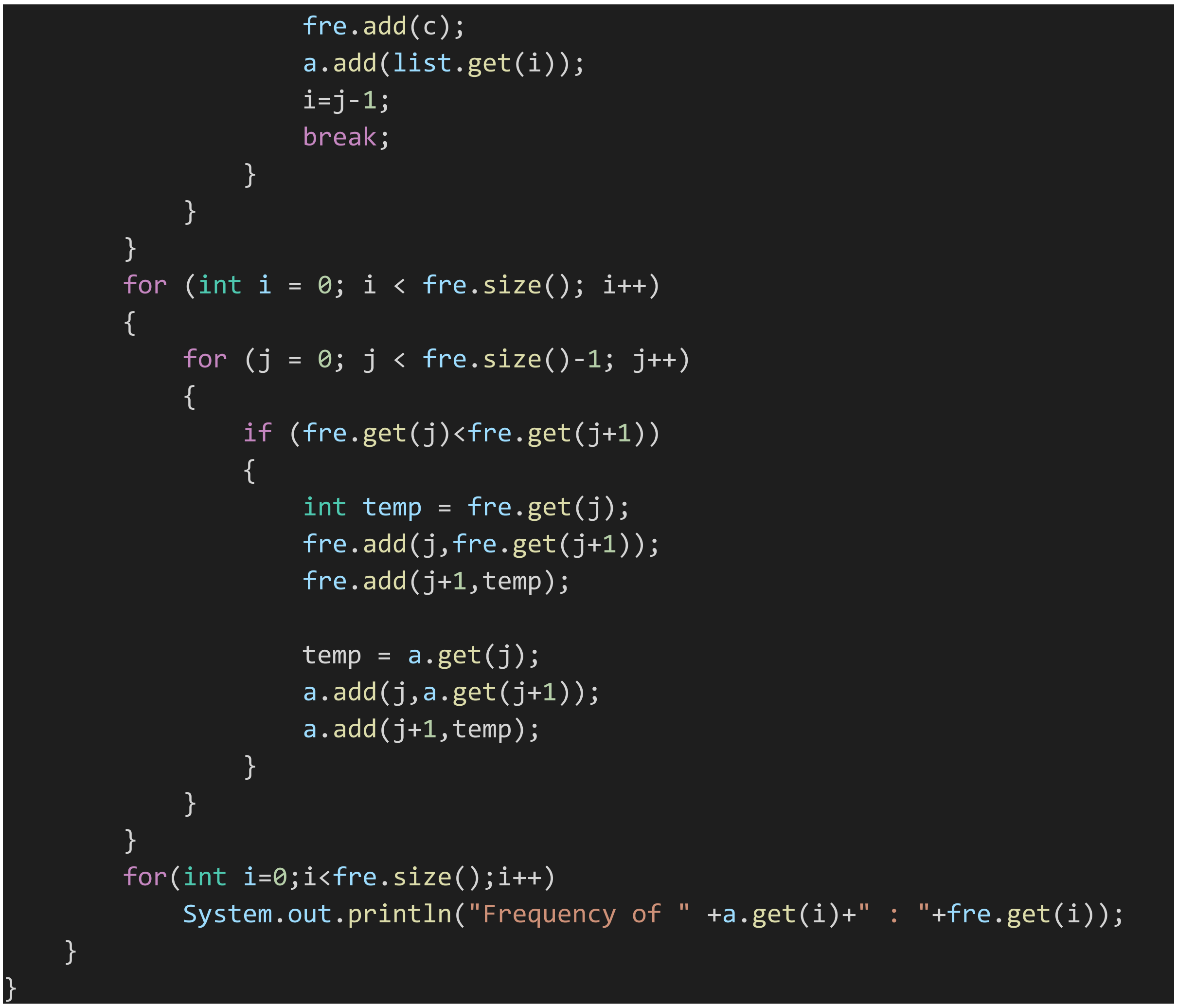
```
PS C:\Users\hp\Desktop\Langs\Java> java OddP.java
Enter size of the array: 5
Enter elements: 1 2 5 7 9
5 7
```

c) Write a program to accept contents into an Integer Array and print the frequency of each number in the order of their number of occurrences.

CODE:

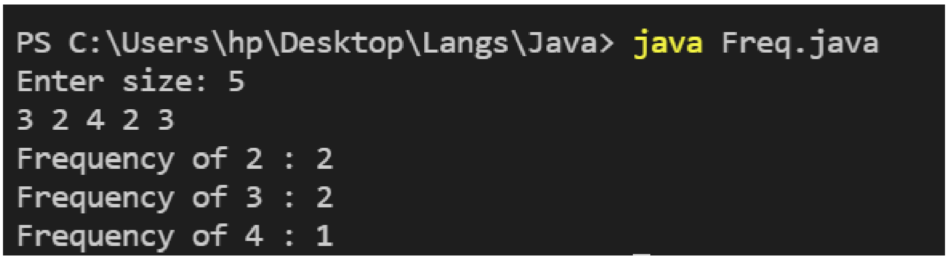


```
import java.util.*;
public class Freq{
    public static void main(String... args)
    {
        ArrayList<Integer> list = new ArrayList<Integer>();
        ArrayList<Integer> fre = new ArrayList<Integer>();
        ArrayList<Integer> a = new ArrayList<Integer>();

        System.out.print("Enter size: ");
        int j;
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        for(int i=0;i<n;i++)
        {
            int x = in.nextInt();
            list.add(x);
        }
        Collections.sort(list);
        for(int i=0; i<n; i++)
        {
            int c=0;
            for(j=i;j<=n;j++)
            {
                if (j==list.size()) {
                        a.add(list.get(i));
                        fre.add(c);
                        i=j;
                    }
                else if(list.get(i)==list.get(j) )
                    c+=1;

                else {
```



```
                    fre.add(c);
                    a.add(list.get(i));
                    i=j-1;
                    break;
                }
            }
        }
        for (int i = 0; i < fre.size(); i++)
        {
            for (j = 0; j < fre.size()-1; j++)
            {
                if (fre.get(j)<fre.get(j+1))
                {
                    int temp = fre.get(j);
                    fre.add(j,fre.get(j+1));
                    fre.add(j+1,temp);

                    temp = a.get(j);
                    a.add(j,a.get(j+1));
                    a.add(j+1,temp);
                }
            }
        }
        for(int i=0;i<fre.size();i++)
            System.out.println("Frequency of " +a.get(i)+" : "+fre.get(i));
    }
}
```

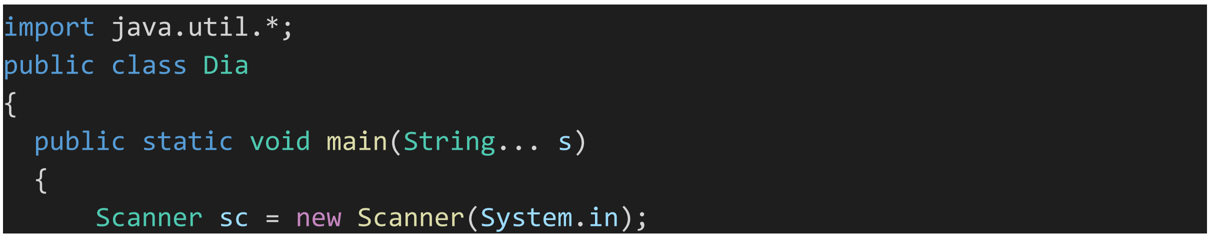
INPUT & OUTPUT:



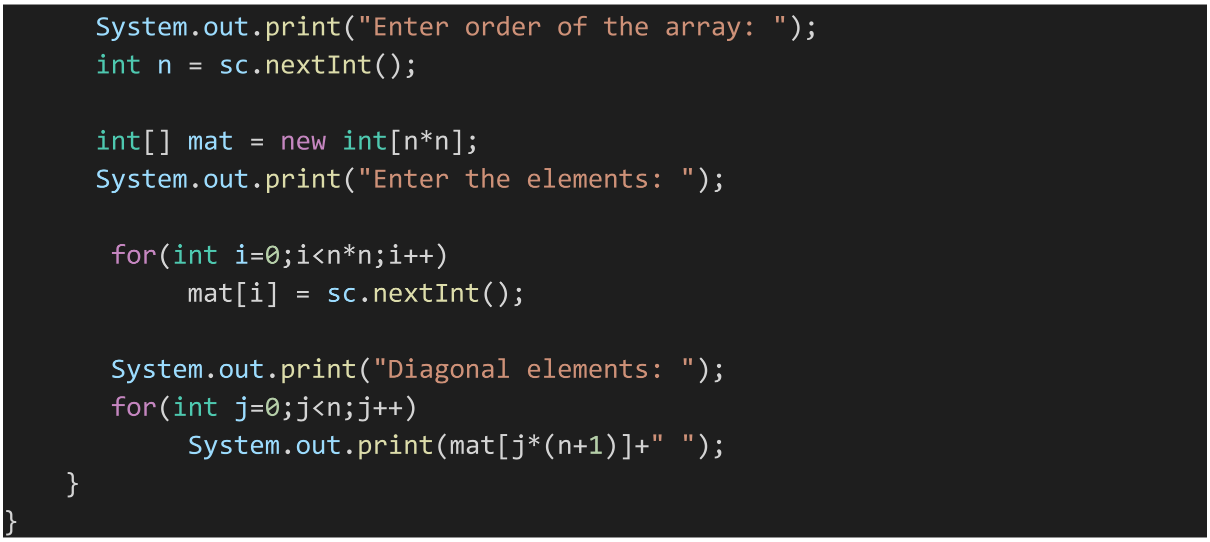
```
PS C:\Users\hp\Desktop\Langs\Java> java Freq.java
Enter size: 5
3 2 4 2 3
Frequency of 2 : 2
Frequency of 3 : 2
Frequency of 4 : 1
```

3. a) Write a program that accepts an ‘n’ ordered square matrix elements into a single dimension array and print the elements of leading diagonal (top left to bottom right).

CODE:

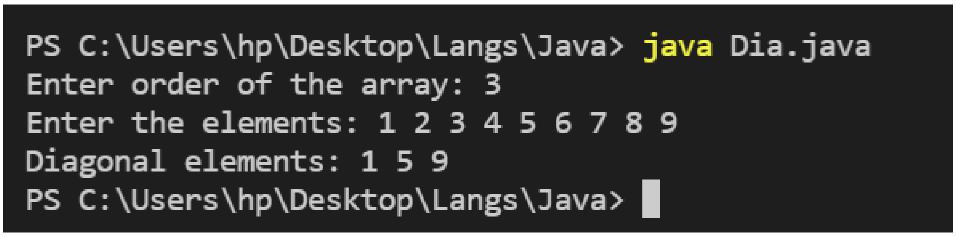


```
import java.util.*;
public class Dia
{
  public static void main(String... s)
  {
      Scanner sc = new Scanner(System.in);
```



```
      System.out.print("Enter order of the array: ");
      int n = sc.nextInt();

      int[] mat = new int[n*n];
      System.out.print("Enter the elements: ");

       for(int i=0;i<n*n;i++)
            mat[i] = sc.nextInt();

       System.out.print("Diagonal elements: ");
       for(int j=0;j<n;j++)
            System.out.print(mat[j*(n+1)]+" ");
    }
}
```

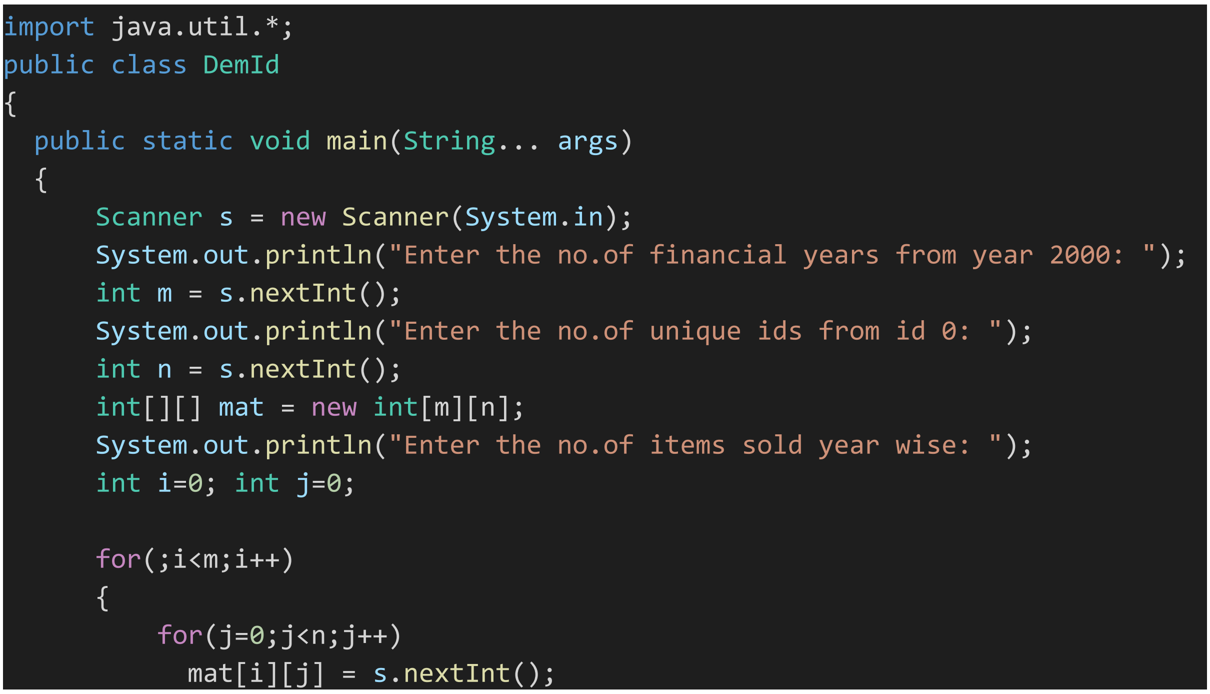
INPUT & OUTPUT:



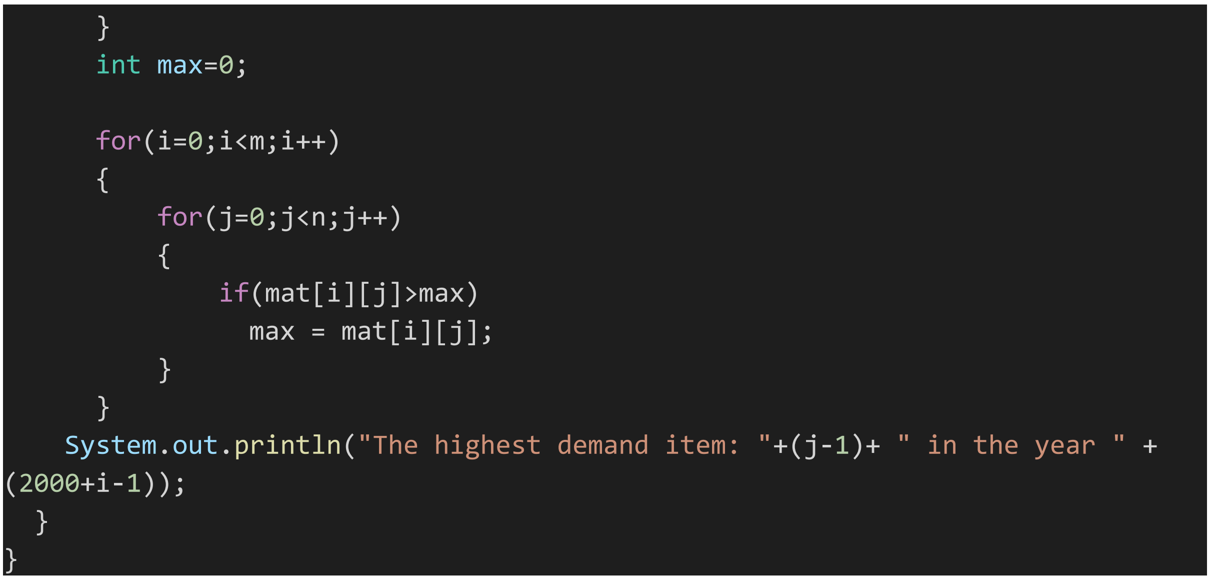
```
PS C:\Users\hp\Desktop\Langs\Java> java Dia.java
Enter order of the array: 3
Enter the elements: 1 2 3 4 5 6 7 8 9
Diagonal elements: 1 5 9
PS C:\Users\hp\Desktop\Langs\Java>
```

b) Write a program that accepts an “m x n‟ double dimension array, where “m‟ represents financial years and “n‟ represents Ids of the items sold. Each element in the array represents the number of items sold in a particular year. Identify the year and id of the item which has more demand.

CODE:

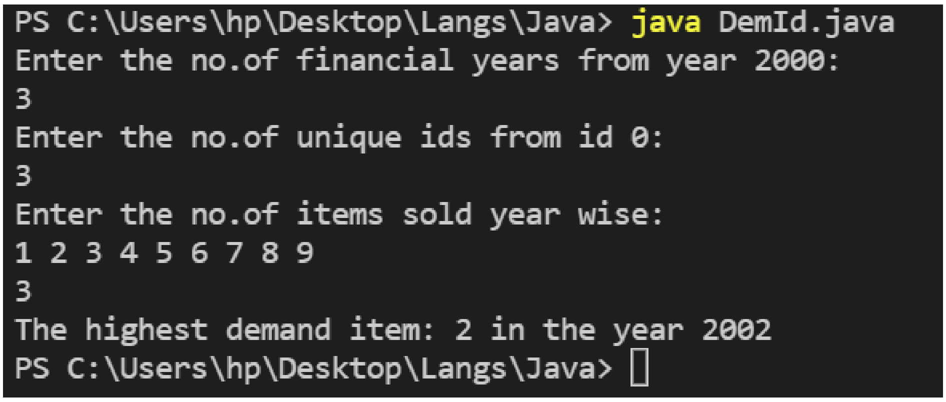


```
import java.util.*;
public class DemId
{
  public static void main(String... args)
  {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter the no.of financial years from year 2000: ");
      int m = s.nextInt();
      System.out.println("Enter the no.of unique ids from id 0: ");
      int n = s.nextInt();
      int[][] mat = new int[m][n];
      System.out.println("Enter the no.of items sold year wise: ");
      int i=0; int j=0;

      for(;i<m;i++)
      {
          for(j=0;j<n;j++)
            mat[i][j] = s.nextInt();
```



```
      }
      int max=0;

      for(i=0;i<m;i++)
      {
          for(j=0;j<n;j++)
          {
              if(mat[i][j]>max)
                max = mat[i][j];
          }
      }
    System.out.println("The highest demand item: "+(j-1)+ " in the year " +
(2000+i-1));
  }
}
```

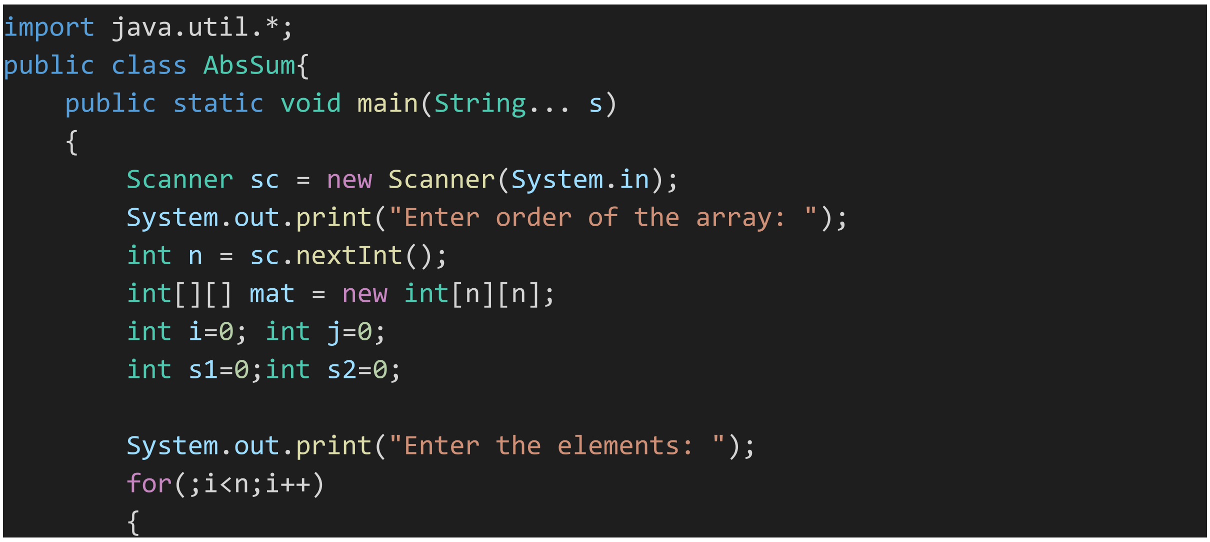
INPUT & OUTPUT:



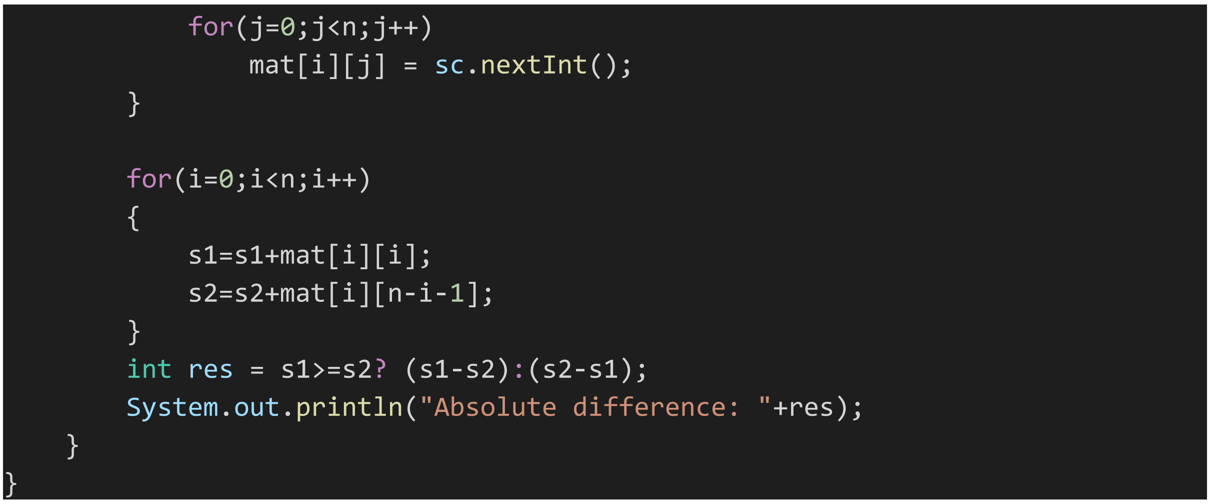
```
PS C:\Users\hp\Desktop\Langs\Java> java DemId.java
Enter the no.of financial years from year 2000:
3
Enter the no.of unique ids from id 0:
3
Enter the no.of items sold year wise:
1 2 3 4 5 6 7 8 9
3
The highest demand item: 2 in the year 2002
PS C:\Users\hp\Desktop\Langs\Java> 
```

c) Write a program that accepts an "n" ordered square matrix and calculate the absolute difference between the sums of elements in their diagonals.

CODE:

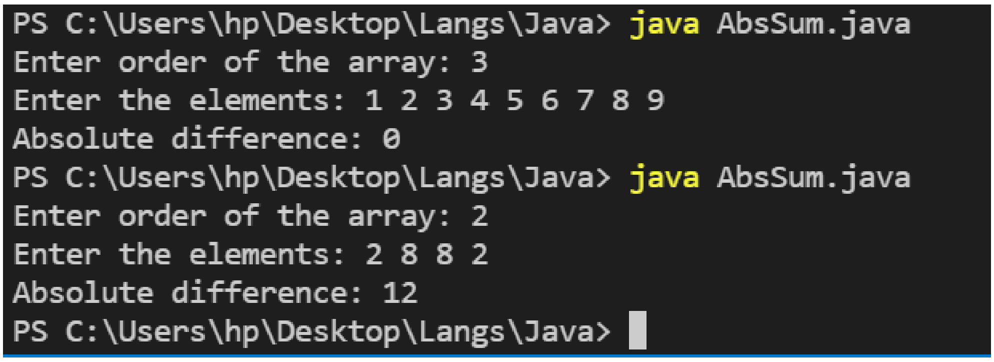


```
import java.util.*;
public class AbsSum{
    public static void main(String... s)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter order of the array: ");
        int n = sc.nextInt();
        int[][] mat = new int[n][n];
        int i=0; int j=0;
        int s1=0;int s2=0;

        System.out.print("Enter the elements: ");
        for(;i<n;i++)
        {
```



```
            for(j=0;j<n;j++)
                mat[i][j] = sc.nextInt();
        }

        for(i=0;i<n;i++)
        {
            s1=s1+mat[i][i];
            s2=s2+mat[i][n-i-1];
        }
        int res = s1>=s2? (s1-s2):(s2-s1);
        System.out.println("Absolute difference: "+res);
    }
}
```

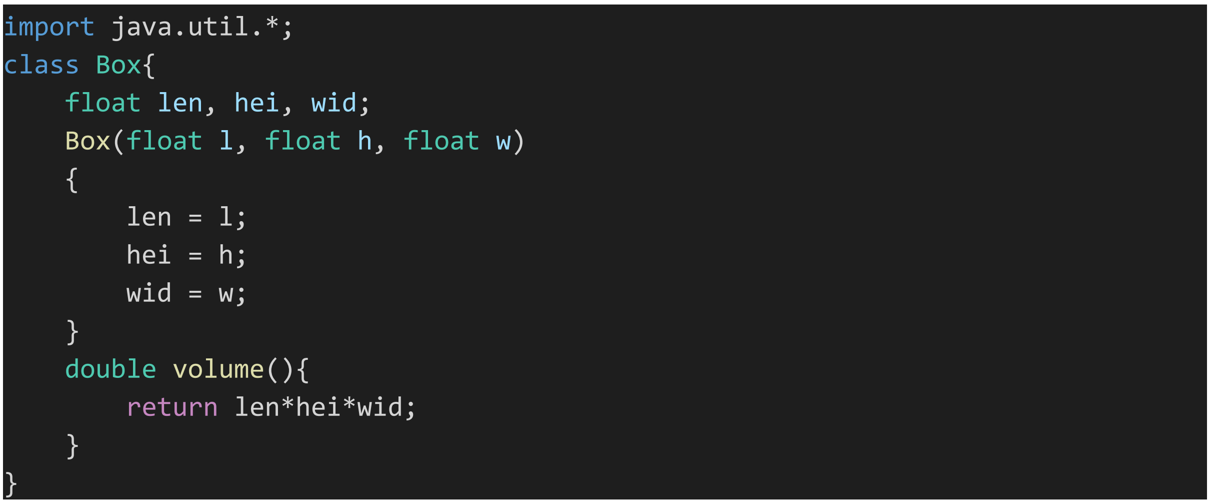
INPUT & OUTPUT:



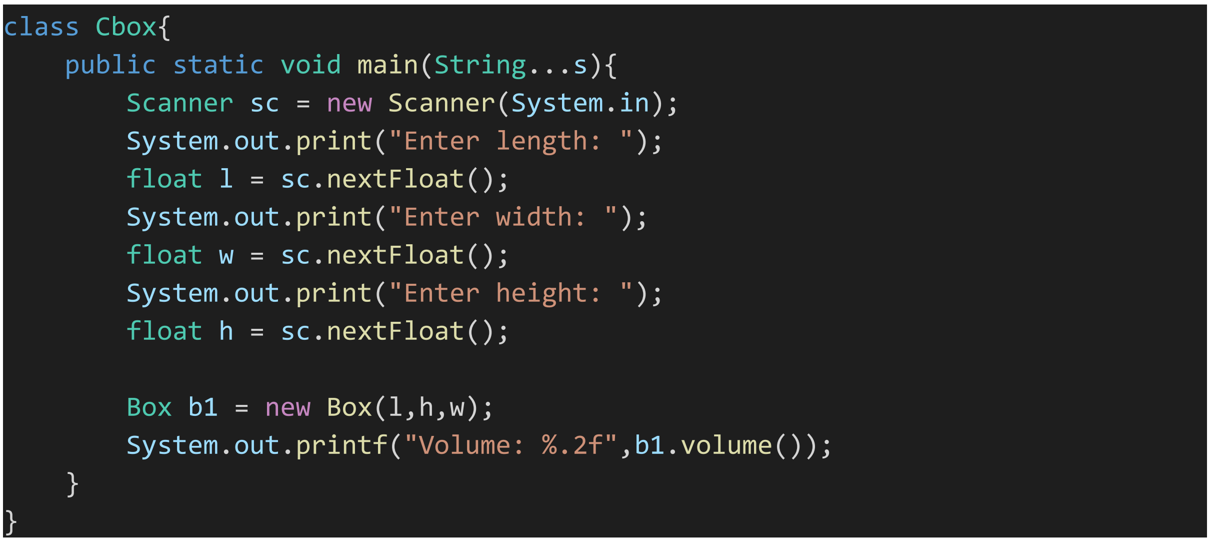
```
PS C:\Users\hp\Desktop\Langs\Java> java AbsSum.java
Enter order of the array: 3
Enter the elements: 1 2 3 4 5 6 7 8 9
Absolute difference: 0
PS C:\Users\hp\Desktop\Langs\Java> java AbsSum.java
Enter order of the array: 2
Enter the elements: 2 8 8 2
Absolute difference: 12
PS C:\Users\hp\Desktop\Langs\Java>
```

4. a) Create a class Box that uses a parameterized constructor to initialize the dimensions of a box. The dimensions of the Box are width, height, depth. The class should have a method that can return the volume of the box. Create an object of the Box class and test the functionality.

CODE:

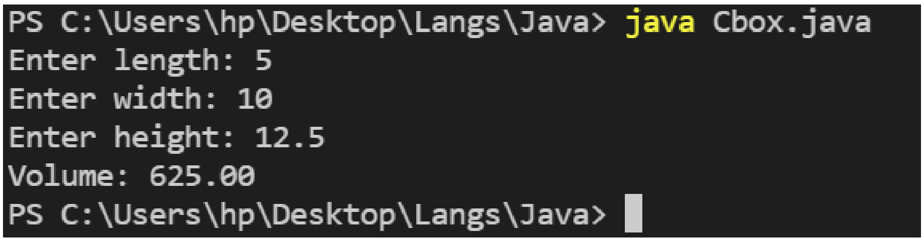


```
import java.util.*;
class Box{
    float len, hei, wid;
    Box(float l, float h, float w)
    {
        len = l;
        hei = h;
        wid = w;
    }
    double volume(){
        return len*hei*wid;
    }
}
```



```
class Cbox{
    public static void main(String...s){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter length: ");
        float l = sc.nextFloat();
        System.out.print("Enter width: ");
        float w = sc.nextFloat();
        System.out.print("Enter height: ");
        float h = sc.nextFloat();

        Box b1 = new Box(l,h,w);
        System.out.printf("Volume: %.2f",b1.volume());
    }
}
```

INPUT & OUTPUT:

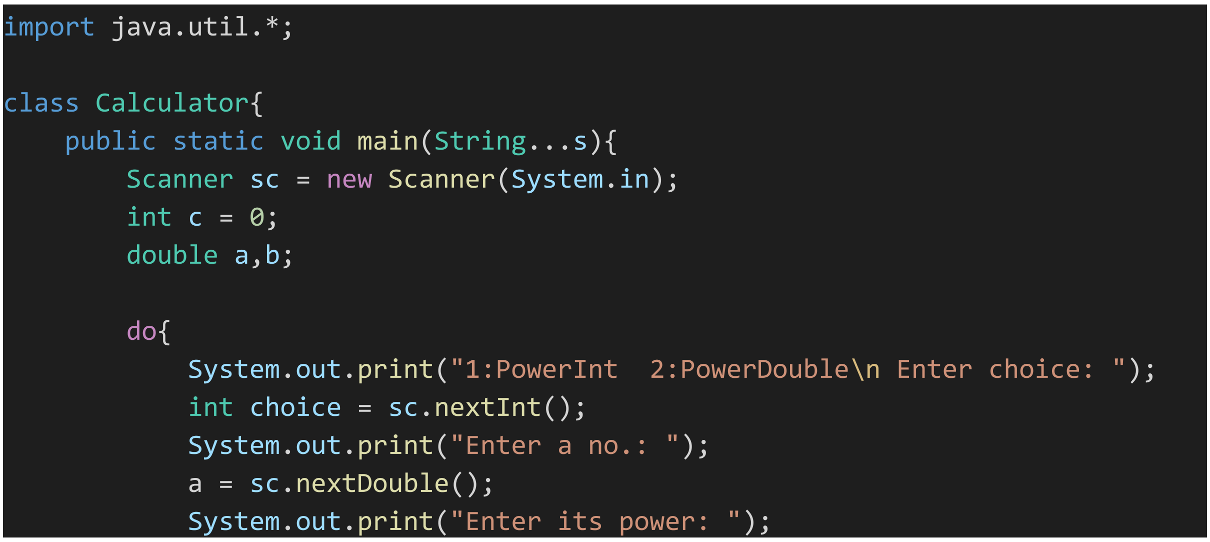


```
PS C:\Users\hp\Desktop\Langs\Java> java Cbox.java
Enter length: 5
Enter width: 10
Enter height: 12.5
Volume: 625.00
PS C:\Users\hp\Desktop\Langs\Java>
```

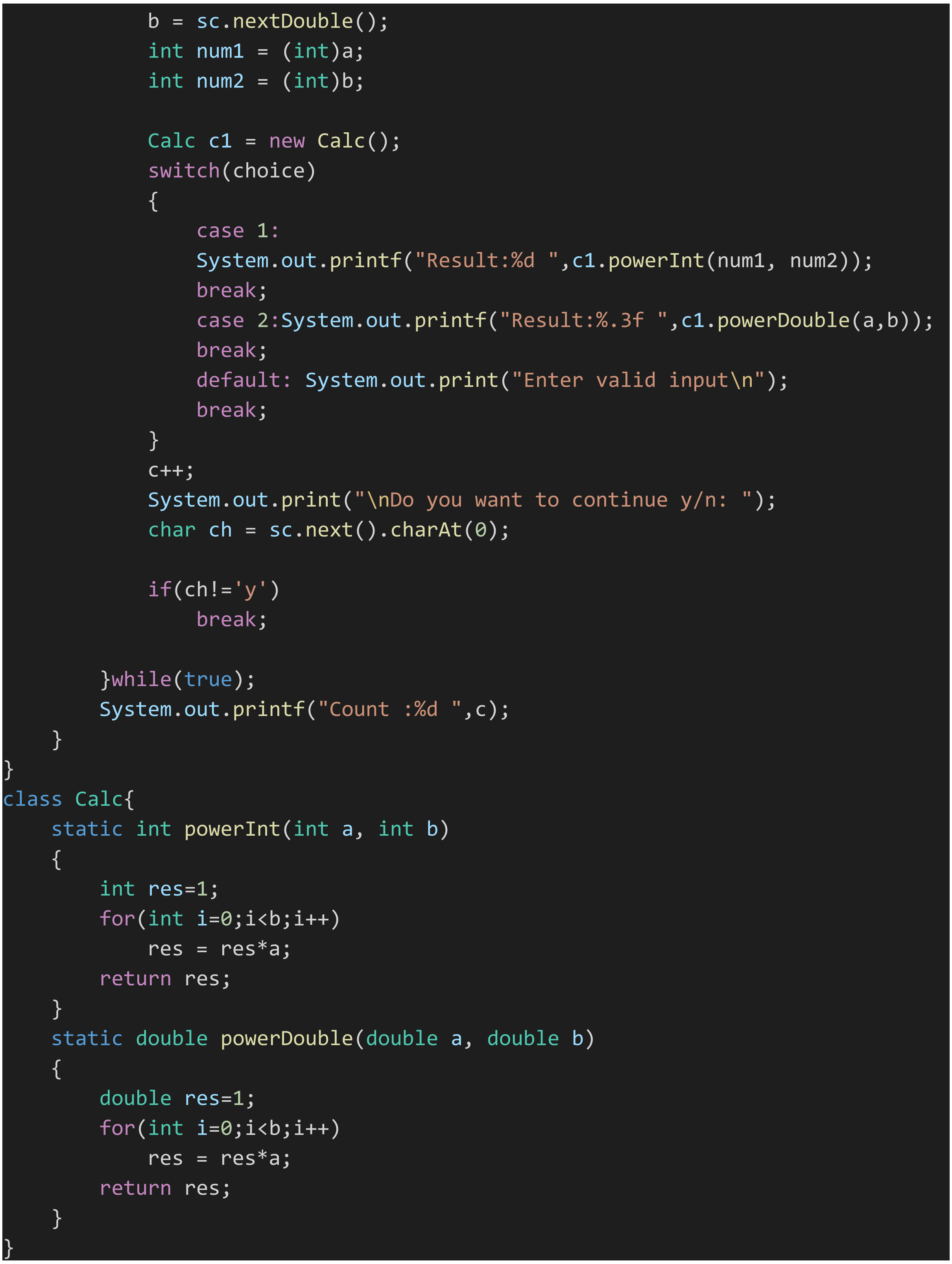
b) Create a new class called Calculator with the following methods:

- A static method called powerInt(int num1,int num2) This method should return num1 to the power num2.
- A static method called powerDouble(double num1,double num2). This method should return num1 to the power num2.
- Invoke both the methods and test the functionality. Also count the number of objects created.

CODE:

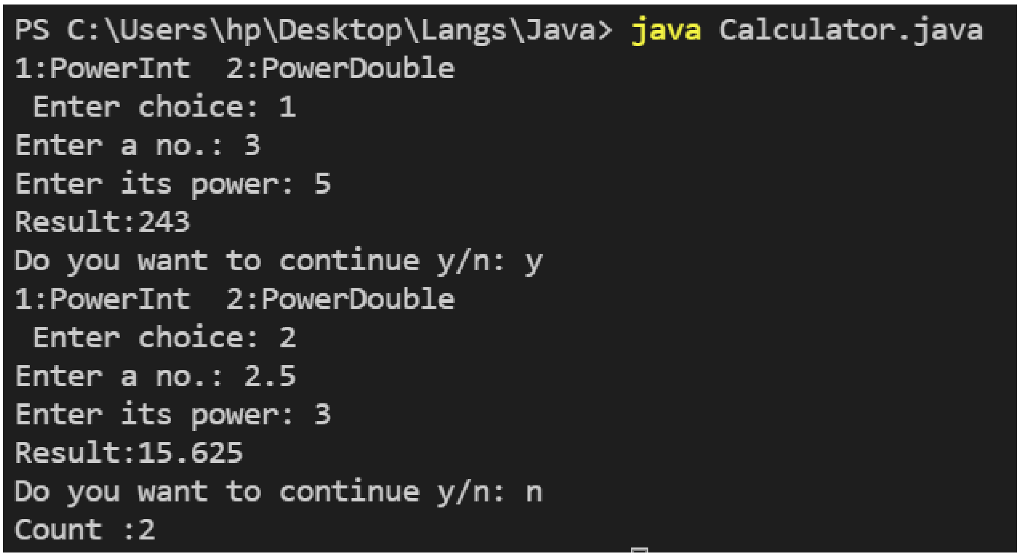


```
import java.util.*;

class Calculator{
    public static void main(String...s){
        Scanner sc = new Scanner(System.in);
        int c = 0;
        double a,b;

        do{
            System.out.print("1:PowerInt  2:PowerDouble\n Enter choice: ");
            int choice = sc.nextInt();
            System.out.print("Enter a no.: ");
            a = sc.nextDouble();
            System.out.print("Enter its power: ");
```



```
            b = sc.nextDouble();
            int num1 = (int)a;
            int num2 = (int)b;

            Calc c1 = new Calc();
            switch(choice)
            {
                case 1:
                System.out.printf("Result:%d ",c1.powerInt(num1, num2));
                break;
                case 2:System.out.printf("Result:%.3f ",c1.powerDouble(a,b));
                break;
                default: System.out.print("Enter valid input\n");
                break;
            }
            c++;
            System.out.print("\nDo you want to continue y/n: ");
            char ch = sc.next().charAt(0);

            if(ch!='y')
                break;

        }while(true);
        System.out.printf("Count :%d ",c);
    }
}
class Calc{
    static int powerInt(int a, int b)
    {
        int res=1;
        for(int i=0;i<b;i++)
            res = res*a;
        return res;
    }
    static double powerDouble(double a, double b)
    {
        double res=1;
        for(int i=0;i<b;i++)
            res = res*a;
        return res;
    }
}
```

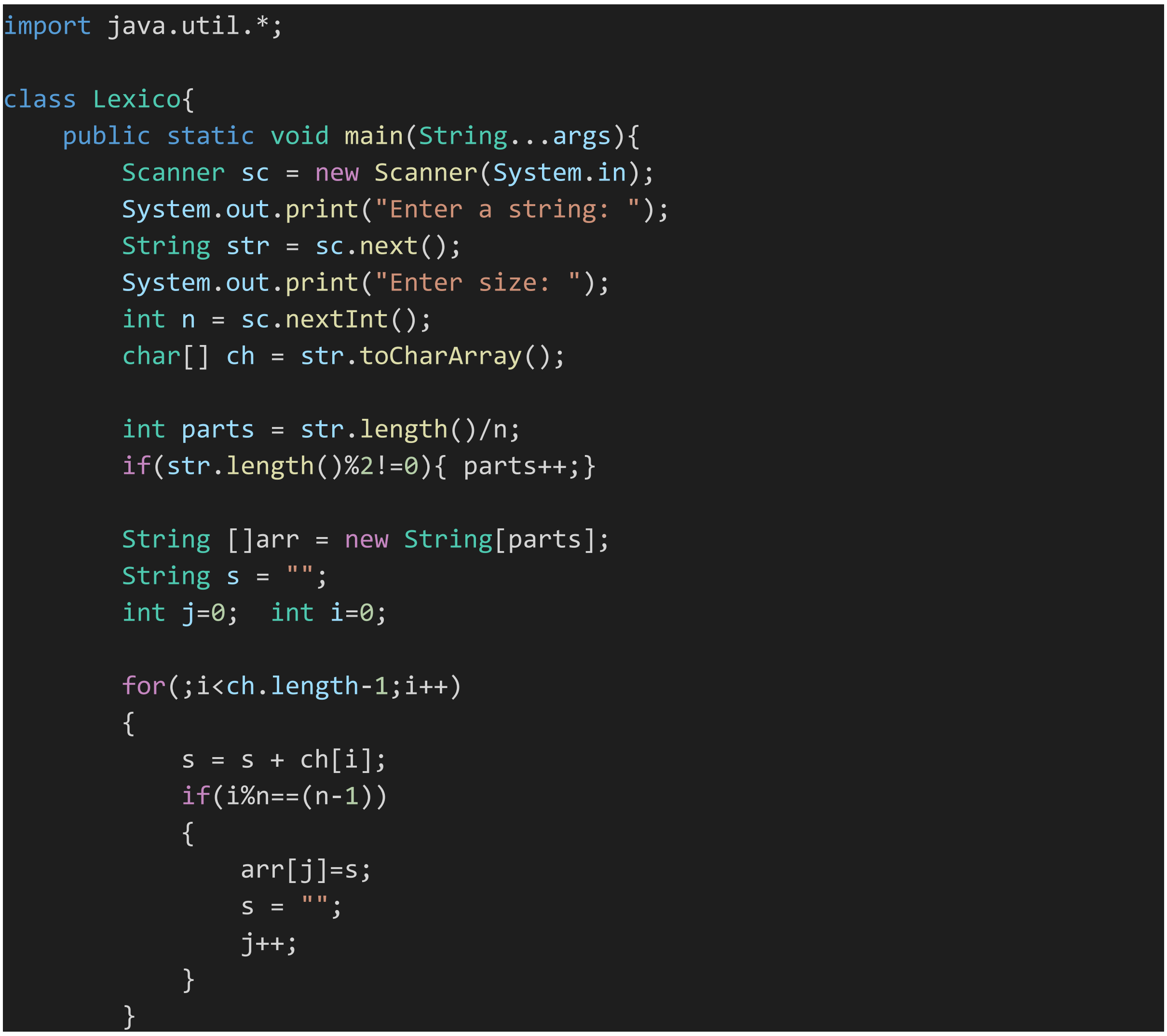
INPUT & OUTPUT:



```
PS C:\Users\hp\Desktop\Langs\Java> java Calculator.java
1:PowerInt  2:PowerDouble
 Enter choice: 1
Enter a no.: 3
Enter its power: 5
Result:243
Do you want to continue y/n: y
1:PowerInt  2:PowerDouble
 Enter choice: 2
Enter a no.: 2.5
Enter its power: 3
Result:15.625
Do you want to continue y/n: n
Count :2
```

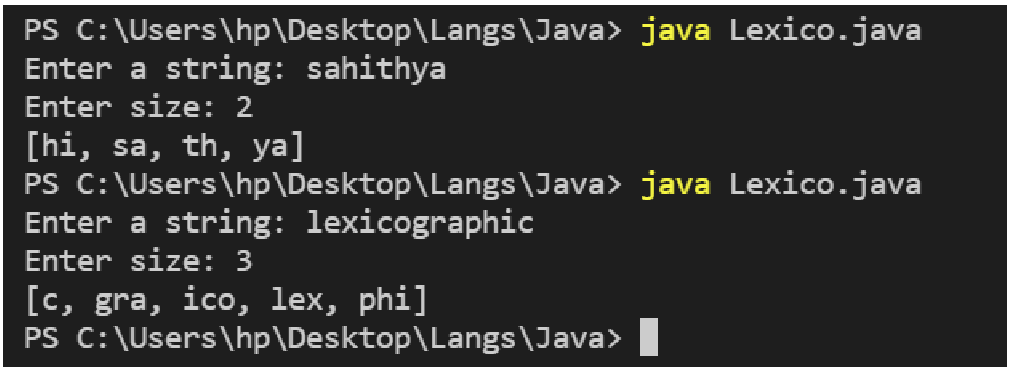
5. a) Accept a String and a number “n‟ from user. Divide the given string into substrings each of size “n‟ and sort them lexicographically.

CODE:



```
import java.util.*;

class Lexico{
    public static void main(String...args){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = sc.next();
        System.out.print("Enter size: ");
        int n = sc.nextInt();
        char[] ch = str.toCharArray();

        int parts = str.length()/n;
        if(str.length()%2!=0){ parts++;}

        String []arr = new String[parts];
        String s = "";
        int j=0;  int i=0;

        for(;i<ch.length-1;i++)
        {
            s = s + ch[i];
            if(i%n==(n-1))
            {
                arr[j]=s;
                s = "";
                j++;
            }
        }
        s = s + ch[i];
        arr[j]=s;
        Arrays.sort(arr);
        System.out.print(Arrays.toString(arr));
    }
}
```

INPUT & OUTPUT:



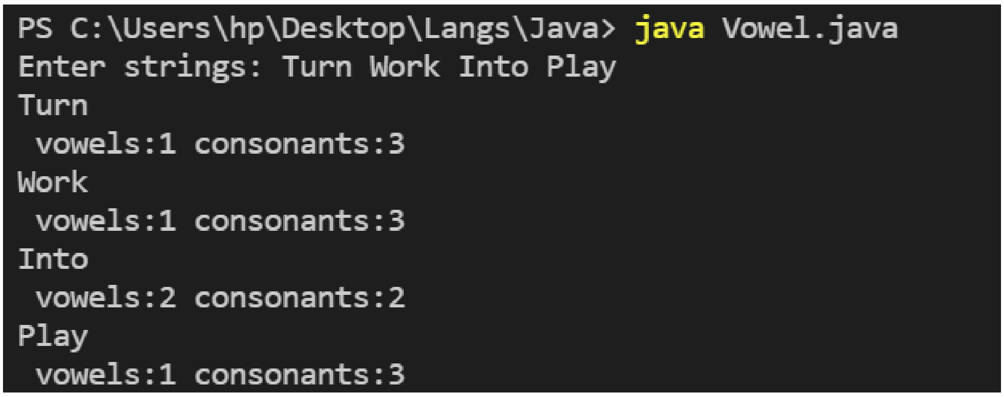
```
PS C:\Users\hp\Desktop\Langs\Java> java Lexico.java
Enter a string: sahithya
Enter size: 2
[hi, sa, th, ya]
PS C:\Users\hp\Desktop\Langs\Java> java Lexico.java
Enter a string: lexicographic
Enter size: 3
[c, gra, ico, lex, phi]
PS C:\Users\hp\Desktop\Langs\Java>
```

b) Accept an array of strings and display the number of vowels and consonants occurred in each string.

CODE:

```
import java.util.*;

class Vowel{
    public static void main(String...args){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter strings: ");
        String []str = sc.nextLine().split(" ");

        for(String i:str)
        {
          char[] ch = i.toLowerCase().toCharArray();
          int v = 0;
            for(char j:ch)
            {
                if(j=='a'|| j=='e'|| j=='i'|| j=='o'|| j=='u')
                    v++;
            }
            System.out.printf("%s\n vowels:%d consonants:%d\n",i, v, i.length(
)-v);
        }
    }
}
```

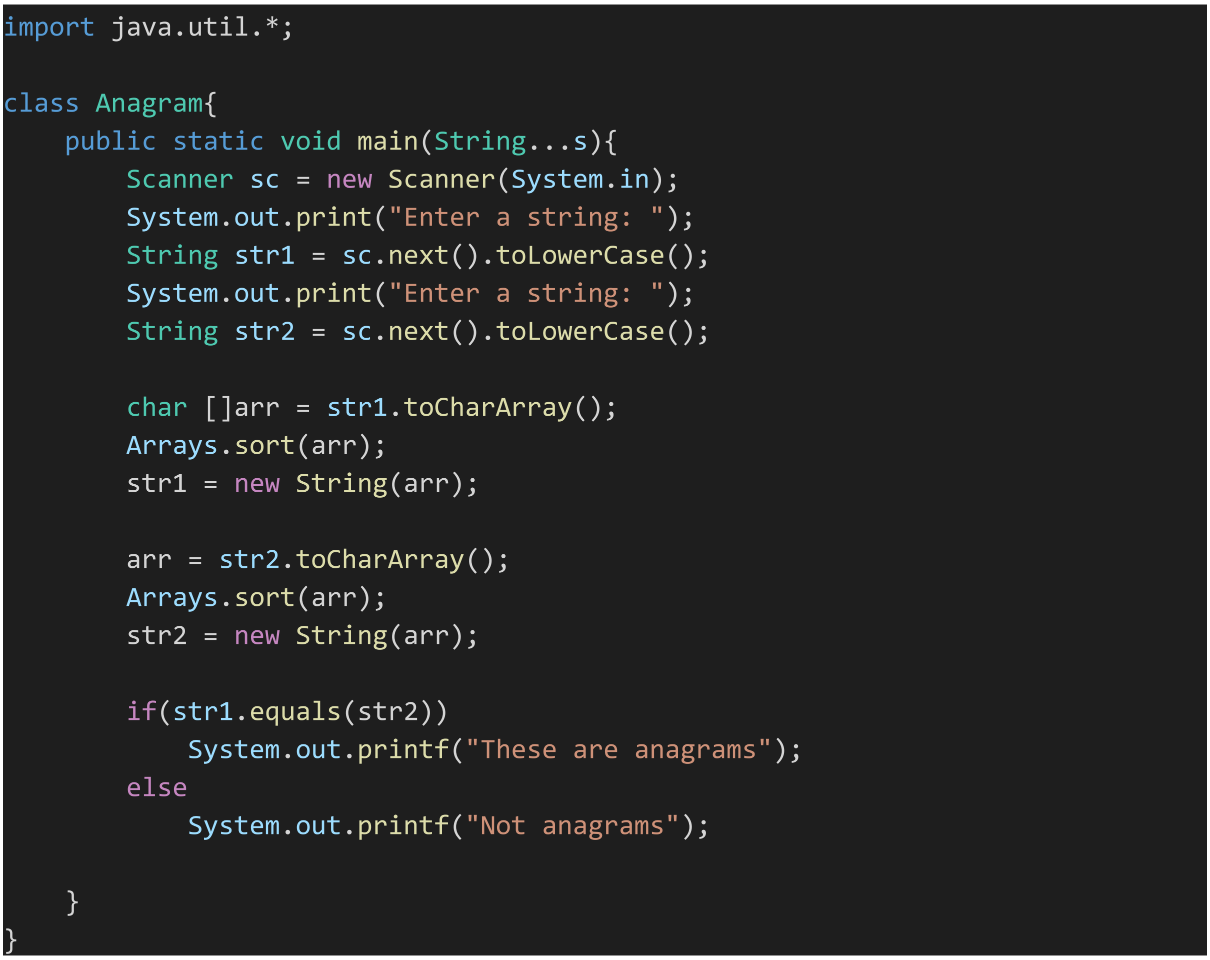
INPUT & OUTPUT:



```
PS C:\Users\hp\Desktop\Langs\Java> java Vowel.java
Enter strings: Turn Work Into Play
Turn
 vowels:1 consonants:3
Work
 vowels:1 consonants:3
Into
 vowels:2 consonants:2
Play
 vowels:1 consonants:3
```

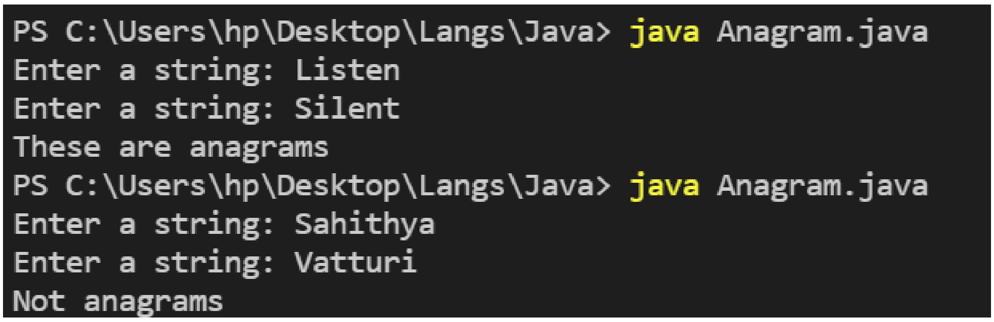
c) Accept two strings from the user and determine if the strings are anagrams or not.

CODE:



```
import java.util.*;

class Anagram{
    public static void main(String...s){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str1 = sc.next().toLowerCase();
        System.out.print("Enter a string: ");
        String str2 = sc.next().toLowerCase();

        char []arr = str1.toCharArray();
        Arrays.sort(arr);
        str1 = new String(arr);

        arr = str2.toCharArray();
        Arrays.sort(arr);
        str2 = new String(arr);

        if(str1.equals(str2))
            System.out.printf("These are anagrams");
        else
            System.out.printf("Not anagrams");

    }
}
```

INPUT & OUTPUT:



```
PS C:\Users\hp\Desktop\Langs\Java> java Anagram.java
Enter a string: Listen
Enter a string: Silent
These are anagrams
PS C:\Users\hp\Desktop\Langs\Java> java Anagram.java
Enter a string: Sahithya
Enter a string: Vatturi
Not anagrams
```