

1. Write a program to create an arraylist of double element and add the elements. sort the elements in descending order and print it.

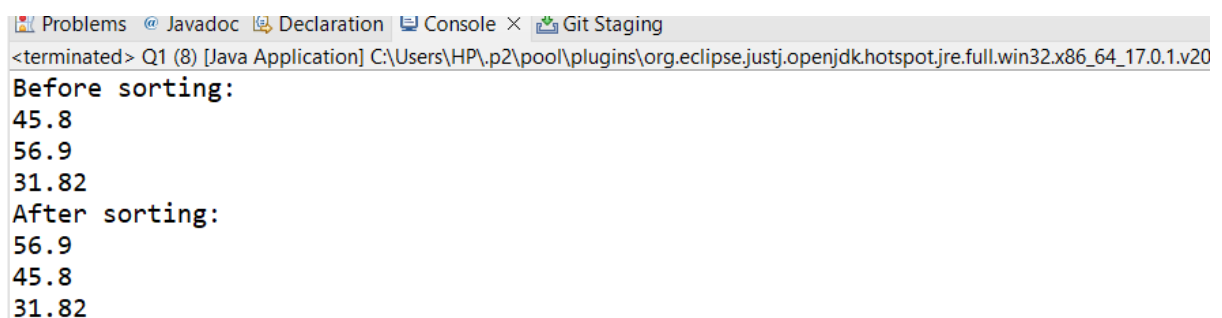
CODE:

```
package sba1;

import java.util.*;
import java.util.Collections;

public class Q1 {
    public static void main(String[] args) {
        ArrayList<Double>list =new ArrayList<Double>();
        list.add(45.8);
        list.add(56.90);
        list.add(31.82);
        System.out.println("Before sorting:");
        for(double newlist:list)
        {
            System.out.println(newlist);
        }
        Collections.sort(list,Collections.reverseOrder());
        System.out.println("After sorting:");
        for(double newlist:list)
        {
            System.out.println(newlist);
        }
    }
}
```

OUTPUT:

A screenshot of the Eclipse IDE's console window. The title bar shows 'Problems', 'Javadoc', 'Declaration', 'Console', and 'Git Staging'. The console text shows the execution of the Java program. It starts with '<terminated> Q1 (8) [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20'. Below this, it prints 'Before sorting:' followed by the values 45.8, 56.9, 31.82 on separate lines. Then it prints 'After sorting:' followed by the values 56.9, 45.8, 31.82 on separate lines, demonstrating a descending sort.

```
<terminated> Q1 (8) [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20
Before sorting:
45.8
56.9
31.82
After sorting:
56.9
45.8
31.82
```

CODE:

OUTPUT:

3. Create two arraylist of strings to take First_name and Last_name of the students, and print their whole name.

CODE:

```
package sba1;

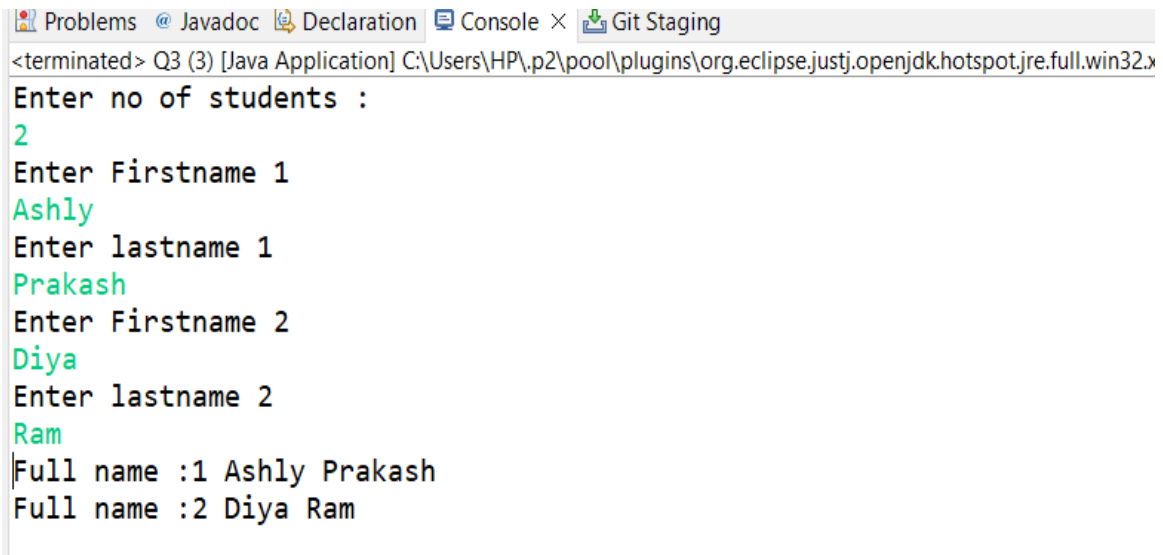
import java.util.ArrayList;

public class Q2 {

    public static void main(String[] args) {
        ArrayList<Integer> sum = new ArrayList<Integer>();
        sum.add(2);
        sum.add(8);
        sum.add(4);
        sum.add(6);
        int total=0;
        double avrg;
        for(int i=0;i<sum.size();i++)

            total= total+sum.get(i);
            avrg= total/sum.size();
            System.out.println("sum "+total);
            System.out.println("Average "+avrg);
    }
}
```

OUTPUT:



```
<terminated> Q3 (3) [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x
Enter no of students :
2
Enter Firstname 1
Ashly
Enter lastname 1
Prakash
Enter Firstname 2
Diya
Enter lastname 2
Ram
Full name :1 Ashly Prakash
Full name :2 Diya Ram
```

4. Write a program to check for the occurrence of a particular character in a string and display how many times it has occurred. note: take the String and the character to be checked as a input from the user.

CODE:

```
package sba1;

import java.util.Scanner;

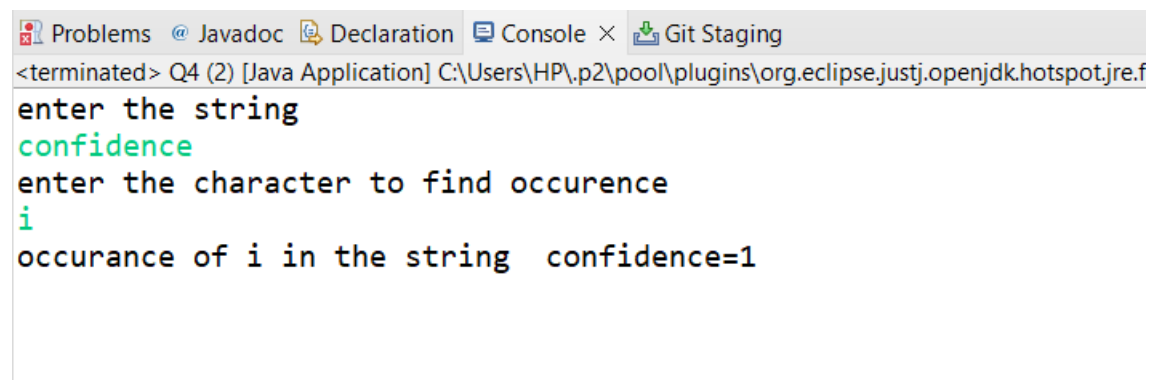
public class Q4 {

    public static void main(String[] args) {
        int count=0;
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the string ");
        String s=sc.next();
        System.out.println("enter the character to find occurrence");
        char s2=sc.next().charAt(0);
        for( int i=0;i<s.length();i++) {
            if(s.charAt(i)==s2) {

                count++;

            }
        }
        System.out.println("occurrence of " +s2+" in the string " +s+
"=" +count);
    }
}
```

OUTPUT:



```
<terminated> Q4 (2) [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.f
enter the string
confidence
enter the character to find occurrence
i
occurrence of i in the string confidence=1
```

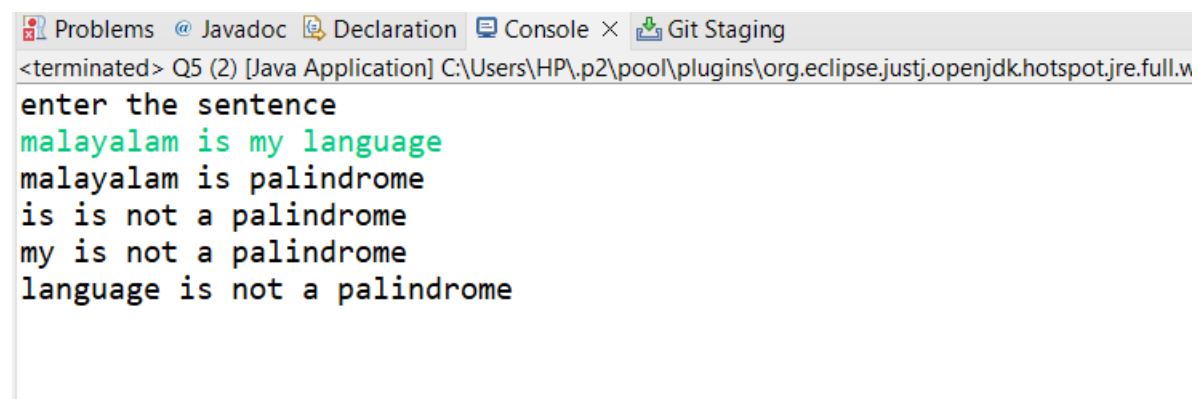
5. Write a program to take an input of a string with multiple words and convert it into a string array, and check if every element of that array is a Palindrome.

Note: Palindrome is a word which when reversed also is the same.

CODE:

```
package sba1;
import java.util.Scanner;
public class Q5 {
    public static boolean checkpalindrome(String str)
    {
        int len =str.length();
        for(int i=0;i<len/2;i++) {
            if(str.charAt(i)!=str.charAt(len-i-1))
                return false;
        }
        return true;
    }
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the sentence");
        String str=sc.nextLine();
        String[] arr=str.split(" ");
        int n=arr.length;
        for(int i=0;i<n;i++)
            if(Q5.checkpalindrome(arr[i])) {
                System.out.println(arr[i]+" is palindrome");
            }
            else
                System.out.println(arr[i]+" is not a
palindrome");
    }
}
```

OUTPUT:

The screenshot shows the Eclipse IDE's console window. At the top, there are tabs for 'Problems', 'Javadoc', 'Declaration', 'Console', and 'Git Staging'. The 'Console' tab is active, displaying the output of the Java application. The output starts with a prompt 'enter the sentence' followed by four lines of user input: 'malayalam is my language', 'malayalam is palindrome', 'is is not a palindrome', and 'my is not a palindrome'. For each input, the program's response is shown on the next line: 'malayalam is my language' is followed by 'malayalam is palindrome', while the other three inputs are followed by 'is is not a palindrome'. The console path at the top reads: '<terminated> Q5 (2) [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.w'.