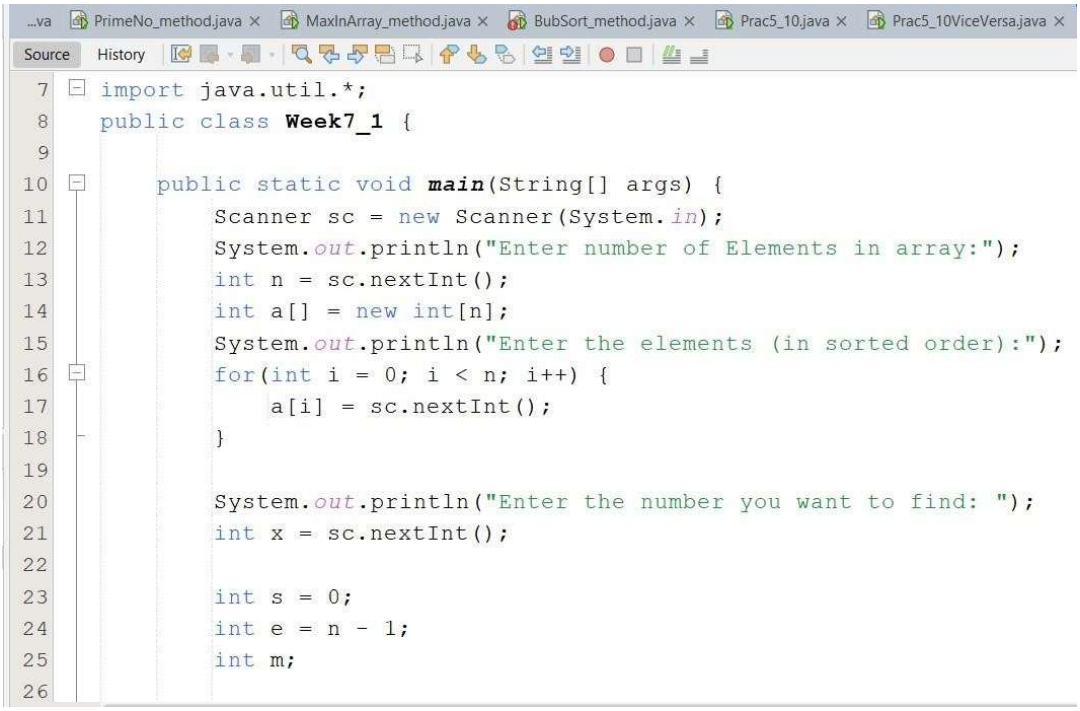
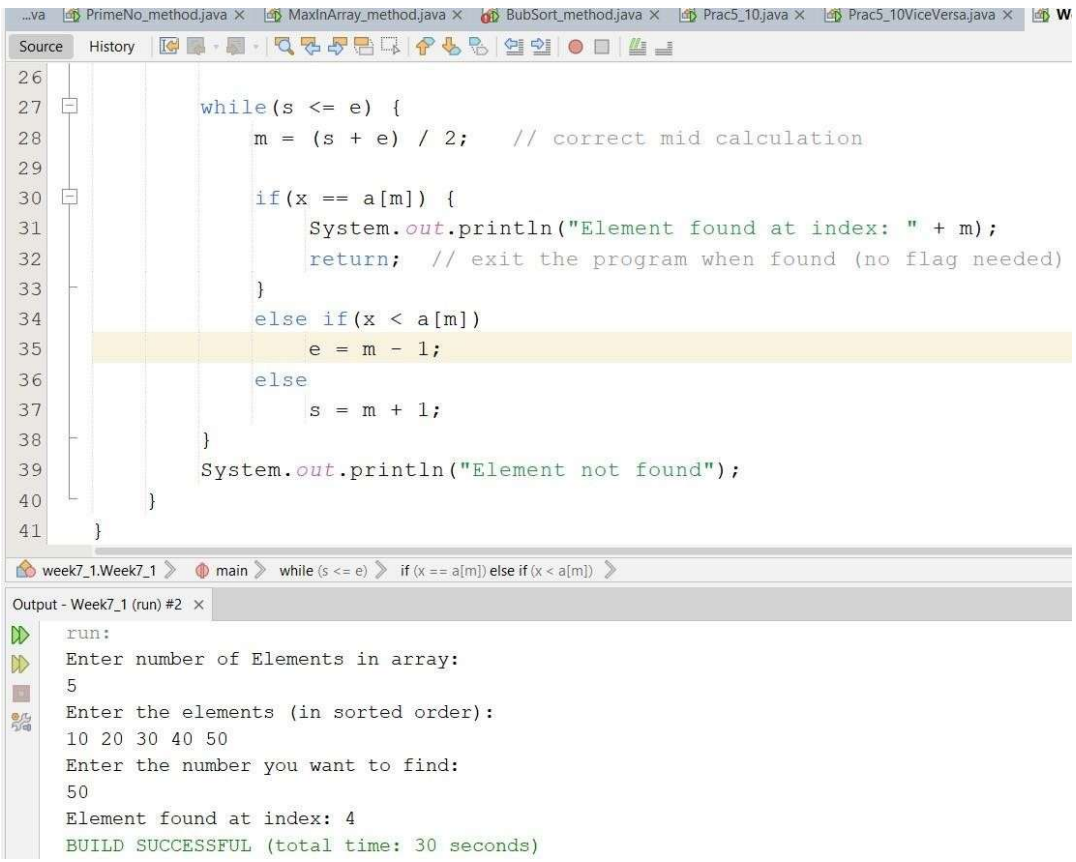
WEEK-7

Problems Based on if statement/Looping/Array in JAVA

1# Write a Java function to implement binary search.




```java
import java.util.*;
public class Week7_1 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of Elements in array:");
        int n = sc.nextInt();
        int a[] = new int[n];
        System.out.println("Enter the elements (in sorted order):");
        for(int i = 0; i < n; i++) {
            a[i] = sc.nextInt();
        }

        System.out.println("Enter the number you want to find: ");
        int x = sc.nextInt();

        int s = 0;
        int e = n - 1;
        int m;
```




```java

        while(s <= e) {
            m = (s + e) / 2;   // correct mid calculation

            if(x == a[m]) {
                System.out.println("Element found at index: " + m);
                return;  // exit the program when found (no flag needed)
            }
            else if(x < a[m])
                e = m - 1;
            else
                s = m + 1;
        }
        System.out.println("Element not found");
    }
}
```

week7_1.Week7_1 > main > while (s <= e) > if (x == a[m]) else if (x < a[m]) >

Output - Week7_1 (run) #2

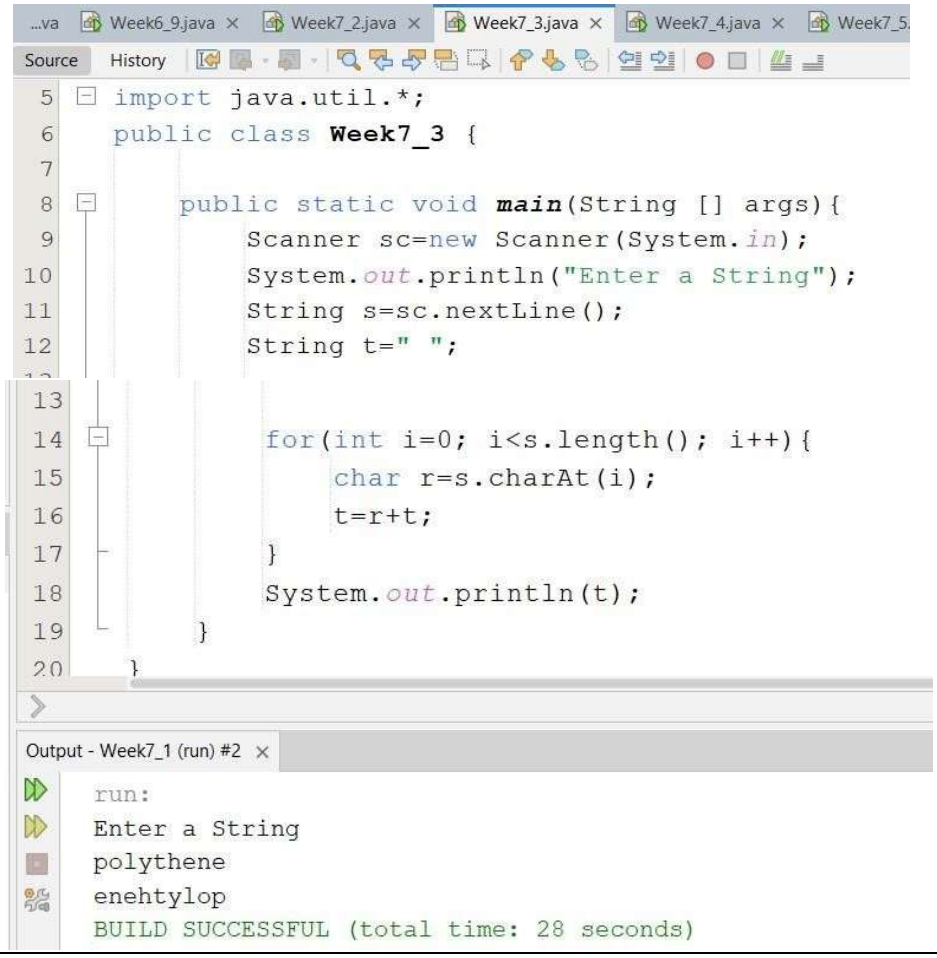
```
run:
Enter number of Elements in array:
5
Enter the elements (in sorted order):
10 20 30 40 50
Enter the number you want to find:
50
Element found at index: 4
BUILD SUCCESSFUL (total time: 30 seconds)
```

```java
import java.util.*;
public class Week7_2 {

    public static void main(String [] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of Elements in array:");
        int n = sc.nextInt();

        int a[] = new int[n];
        System.out.println("Enter the elements:");
        for(int i = 0; i < n; i++) {
            a[i] = sc.nextInt();
        }

        for(int i=0; i<n-1; i++){
            for(int j=0; j<n-i-1; j++){
                if(a[j]>a[j+1]){
                    int t=a[j];
                    a[j]=a[j+1];
                    a[j+1]=t;
                }
            }
        }
```

```
Output - Week7_1 (run) #2
run:
Enter number of Elements in array:
5
Enter the elements:
30 20 50 40 10
Array in ascending
order
10 20 30 40 50 BUILD SUCCESSFUL (total time: 14 seconds)
```

3# Write a program to reverse a given string.

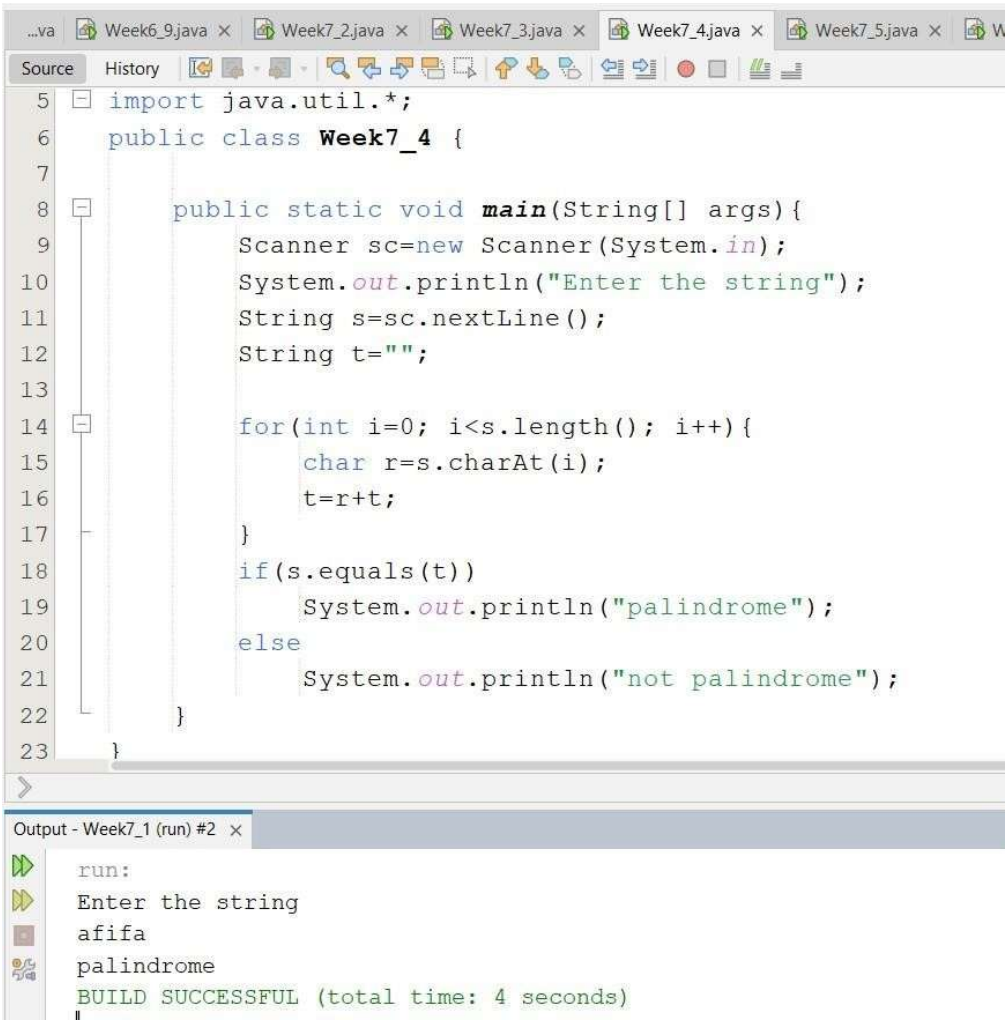
2# Write a Java function to arrange the elements of an array in ascending order (Sorting).



```
...va  Week6_9.java  Week7_2.java  Week7_3.java  Week7_4.java  Week7_5.
Source  History
5  import java.util.*;
6  public class Week7_3 {
7
8      public static void main(String [] args){
9          Scanner sc=new Scanner(System.in);
10         System.out.println("Enter a String");
11         String s=sc.nextLine();
12         String t=" ";
13
14         for(int i=0; i<s.length(); i++){
15             char r=s.charAt(i);
16             t=r+t;
17         }
18         System.out.println(t);
19     }
20 }
```

```
Output - Week7_1 (run) #2
run:
Enter a String
polythene
enehtylop
BUILD SUCCESSFUL (total time: 28 seconds)
```

4# Write a program to check whether a given string is palindrome or not.



5# Write a program to implement factorial of a number through recursion.

```java
import java.util.*;
public class Week7_5 {

    public static int fact(int n){
        if(n==0){
            return 1;
        }
        return n*fact(n-1);
    }
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of Elements in array:");
        int n = sc.nextInt();

        System.out.println(fact(n));
    }
}
```

```
run:
Enter number of Elements in array:
5
120
BUILD SUCCESSFUL (total time: 4 seconds)
```

6# Write a program to implement Fibonacci series of a number with and without recursion.

Without Recursion

```java
package sessional1_;
import java.util.Scanner;
public class Week6_9 {

    public static void fibonacci(int N){
        int first=0;
        int second=1;
        int next;

        for(int i=1; i<=N; i++){
            System.out.println(first+" ");
            next=first+second;
            first=second;
            second=next;
        }
    }
```

```
5
8
13
21
34
BUILD SUCCESSFUL (total time: 6 seconds)
```

With Recursion

```java
import java.util.*;
public class Week7_6 {//with recursion
    public static int fibonacci(int n){
        if(n==0)
            return 0;
        else if(n==1)
            return 1;
        return(fibonacci(n-1)+fibonacci(n-2));
    }

    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number n: ");
        int n=sc.nextInt();
        System.out.println("The Fibonacci series is: ");
        for(int i=0; i<n; i++)
            System.out.print(fibonacci(i)+" ");
    }
```

```
run:
Enter a number n:
6
The Fibonacci series is:
0 1 1 2 3 5 BUILD SUCCESSFUL (total time: 2 seconds)
```

Optional:

7# Write a Java function to find the greatest common divisor (GCD) of two numbers with and without using recursion.

Without recursion

```
import java.util.*;
public class Week7_7WOR {
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the first number: ");
        int a=sc.nextInt();
        System.out.println("Enter the second number: ");
        int b=sc.nextInt();

        int GCD=0;
        for(int i=12; i>=1; i--){
            if((b%i==0) && (a%i==0)){
                GCD=i;
                break;
            }
        }
        System.out.println(GCD);
```

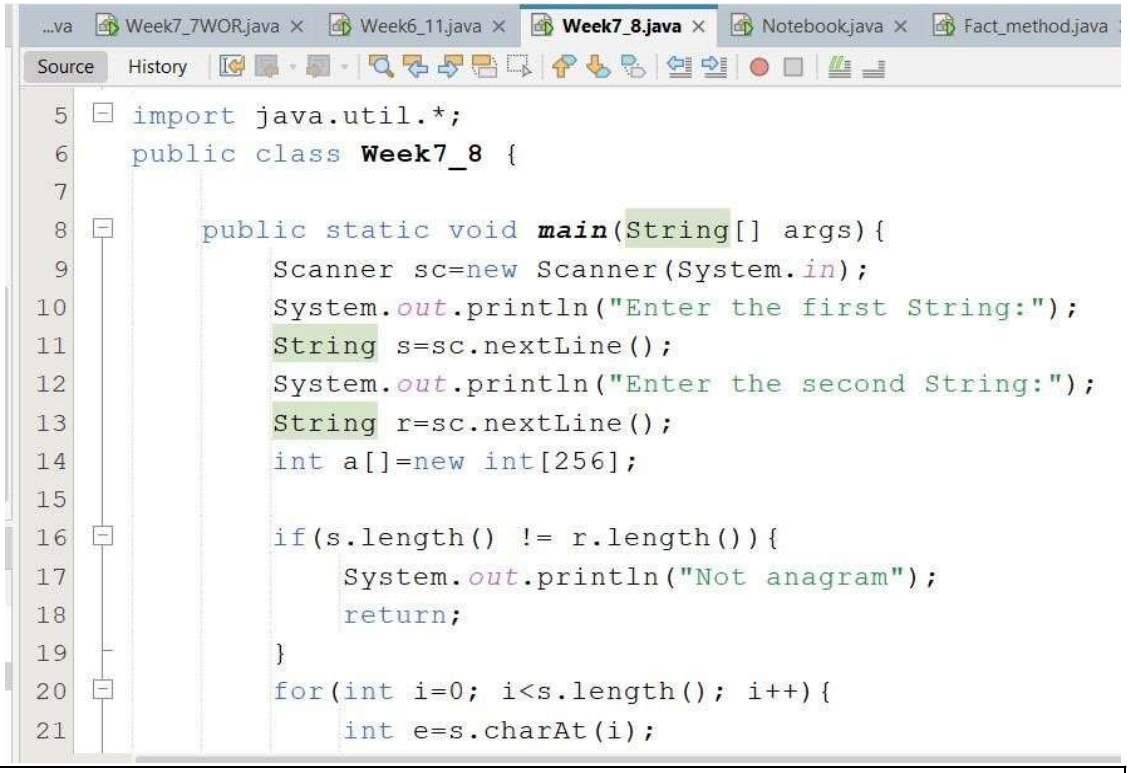
```
run:
Enter the first number:
20
Enter the second number:
12
4
```

With recusion

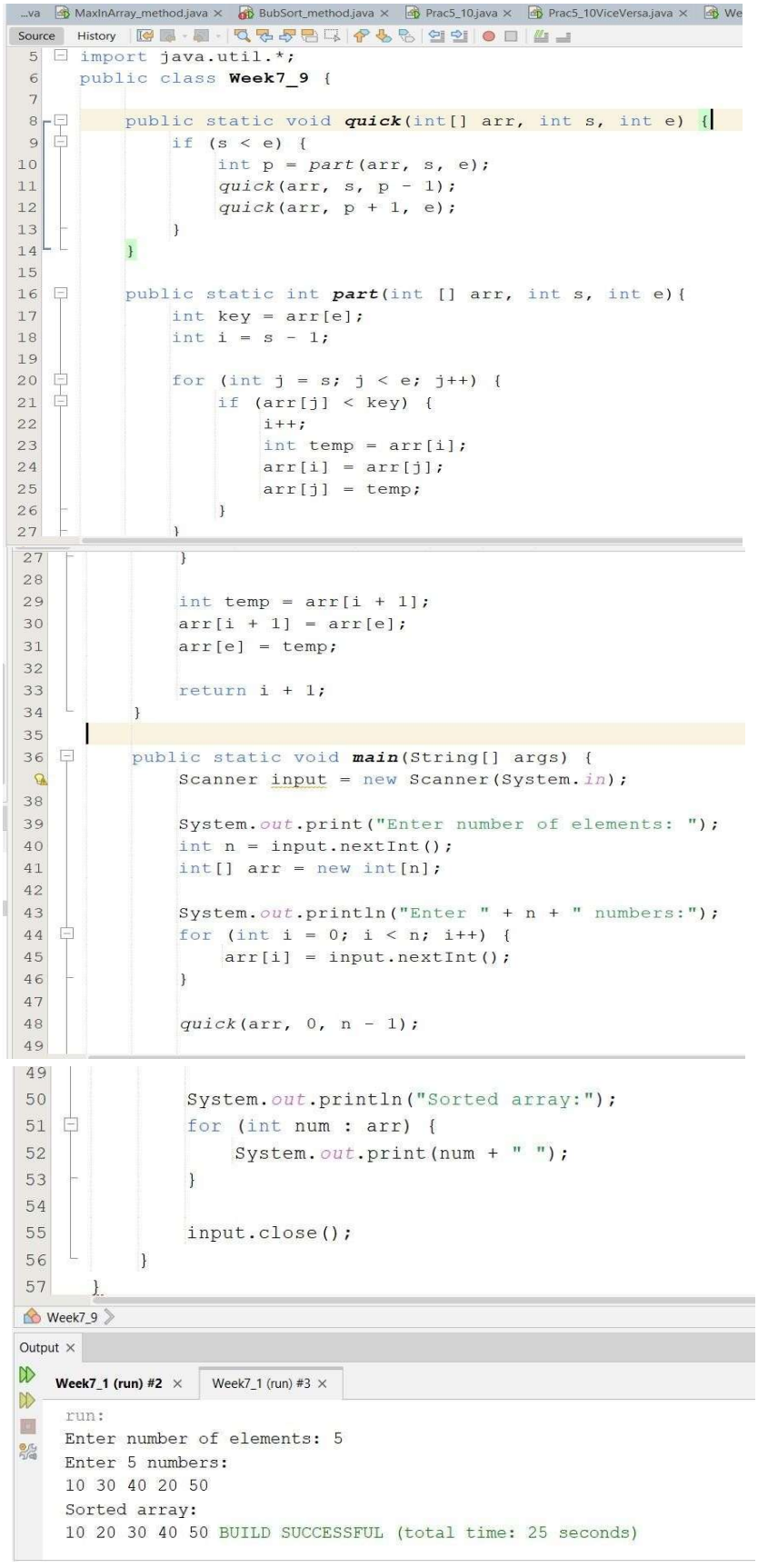
```
import java.util.*;
public class Week7_7WR {

    public static int GCD(int n, int d){
        if(n%d==0)
            return d;
        return(GCD(n, n%d));
    }
    public static void main(String [] args){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the first number:");
        int n=sc.nextInt();
        System.out.println("Enter the second number:");
        int d=sc.nextInt();
        System.out.println(GCD(n,d));
    }
}
```

```
Enter the first number:
20
Enter the second number:
12
4
BUILD SUCCESSFUL (total time: 6 seconds)
```

8# Write a program to check whether two strings are anagrams of each other (“listen” and “silent” are anagrams).



9# Write a Java program to implement Fibonacci series up to N terms (0,1,1,2,3,5....).



```
...va  MaxInArray_method.java ×  BubSort_method.java ×  Prac5_10.java ×  Prac5_10ViceVersa.java ×  We
Source  History
```

```java
import java.util.*;
public class Week7_9 {

    public static void quick(int[] arr, int s, int e) {
        if (s < e) {
            int p = part(arr, s, e);
            quick(arr, s, p - 1);
            quick(arr, p + 1, e);
        }
    }

    public static int part(int [] arr, int s, int e){
        int key = arr[e];
        int i = s - 1;

        for (int j = s; j < e; j++) {
            if (arr[j] < key) {
                i++;
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }

        int temp = arr[i + 1];
        arr[i + 1] = arr[e];
        arr[e] = temp;

        return i + 1;
    }

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

        System.out.print("Enter number of elements: ");
        int n = input.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter " + n + " numbers:");
        for (int i = 0; i < n; i++) {
            arr[i] = input.nextInt();
        }

        quick(arr, 0, n - 1);

        System.out.println("Sorted array:");
        for (int num : arr) {
            System.out.print(num + " ");
        }

        input.close();
    }
}
```

Week7_9

Output ×

Week7_1 (run) #2 × | Week7_1 (run) #3 ×

```
run:
Enter number of elements: 5
Enter 5 numbers:
10 30 40 20 50
Sorted array:
10 20 30 40 50 BUILD SUCCESSFUL (total time: 25 seconds)
```