```
    }
```

93. Define a class with a method to sort an array by bubble sort method.

Ans.

```
import java.io.*;
public class test
{
    public void sort(int A[])
    {
        int a, b, c;
        for(a = 0; a < A.length - 1; a++)
        {
            for(b = a + 1; b < A - length; b + +)
            {
                if(A[a] > A[b])
                {
                c = A[a];
                A[a] = A[b];
                A[b] = C;
                }
            }
        }
    }
}
```

94. Write a program to store weight of 10 students into an array and display those ages which are greater than average age.

Ans.
```
import java.io.*;
public class Test
{
    public static void main(String args[])
    {
        double weight[] = {40.8, 44.9, 50.5, 70.00, 80.5, 38.5, 50.00, 55.00, 30.5, 34.0};
        double sum = 0, Avg = 0;
        for(int i = 0; i < weight.length; i++)
        {
            sum = sum + weight[i];
        }
        Avg = sum/10.0;
        for(i = 0; i < weight - length; i++)
        {
            if(weight[i] > Avg)
            System.out.println(weight [i]);
        }
    }
}
```

95. Define a method to search a value into an array using binary search method.

Ans.
```
import java.io.*;
public class test
{
    public void search(int A [], int n) // n is the value to be search in an Array A.
    {
        int low = 0, high = A.length - 1;
        int f = 0;
        while(low <= high && f == 0)
        {
            mid = (low + high)/2;
            if(A[mid] == n)
            {
                f = 1;
                System.out.println(n+ "found at" + mid + "th position");
                break;
            }
            else if(A[mid] > n)
            high = mid - 1;
            else
            low = mid + 1;
        }
        if(f == 0)
        System.out.print(n + "is not found in array");
    }
}
```

96. Define a class to sort an array using bubble sort method.

Ans.
```
import java.io.*;
public class test
{
        public void sort(int A[])
        {
            int a, b, c;
                int l = A.length - 1;
                for(a = 0; a <= l; a++)
                {
                        for(b = 0; a <= l - 1 - a; b++)
                        {
                                if(A[b] > A[b + 1])
                                {
                                c = A[b];
                                A[b] = A[b + 1];
                                A[b + 1] = c;
                                }
                        }
                }
        }
}
```

97. Define a class to search a value into an array using linear search techniques.

Ans.
```
import java.io.*;
public class test
{
        public void search(int A[], int n) // n is the value to be search and || A[] is the Array.
        {
        int f = 0;
        int l = A.length;
            for(int i = 0; i <= l - 1 && f == 0; i++)
            {
                if (A[i] == n)
                {
                    System.out.print(n + "is found at" + i + "positon");
                    f = 1;
                }
            }
        if(f == 0)
        System.out.print(n+ "not found in list");
        }
}
```

98. Write a Java program to create a single dimensional array for N numbers and transfer the smallest element to the left side and the second smallest element to the right side and as so on up to n numbers.

Ans.
```
import java. io.*;
public class transfer
{
    public static void main(String args[]) throws IOException
    {
        int a[] = new int[50];
        BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
        int n;
```

```
        System.out.print("Enter Number of Elements You Want To Input");
        String d = input.readLine();
        n = Integer.parseInt(d);
        String numb;
        for(int i = 0; i < n; i++)
        {
            System.out.print("Enter any number");
            numb = input.readLine();
            a[i] = Integer.parseInt(numb);
        }
        int s, e, m, p, min = 0, t, j, i;
        s = 0;
        e = n - 1;
        for(i = 1; i <= n; i++)
        {
            min = a[s];
            p = s;
            for(j = s; j <= e; j++)
            {
                if(a[j] < min)
                {
                    min = a[j];
                    p = j;
                }
            }
            if(i % 2 == 1)
            {
                t = a[s];
                a[s] = a[p];
                a[p] = t;
                s++;
            }
            else
            {
                t = a[e];
                a[e] = a[p];
                a[p] = t;
                e--;
            }
        }
        System.out.println("Now Output is");
        for(m = 0; m < n; m++)
        {
            System.out.print(a[m] + " ");
        }
    }
}
```

Output :

```
Enter Number of Elements You Want To Input - - > 5
Enter Any Number - - > 25
```

```
Enter Any Number - - > 36
Enter Any Number - - > 15
Enter Any Number - - > 2
Enter Any Number - - > 26
Now Output Is :
2 25 36 26 15
```

99. Write a Java program which searches an element from an array and locates the position of the searched element.

Ans.

```
// Program for searching an element in an Array.
import java.io.*;
public class search
{
        public static void main(String args[]) throws IOException
        {
                int i;
                int[] array = new int[50];
                int look = 0;
                int location = 0;
                BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
                System.out.print("Enter Numbers in Array = ");
                String x = input.readLine();
                String numb;
                int n = Integer.parseInt(x);
                for(i = 0; i < n; i++)
                {
                    System.out.print("Enter Number =" + (i + 1) + " : : ");
                    numb = input.readLine();
                    array[i] = Integer.parseInt(numb);
                }
                System.out.print("Enter Number to be searched = ");
                String a = input.readLine();
                look = Integer.parseInt(a);
                for(i = 0; i < n; i++)
                {
                    if(array[i] == look)
                    location = i + 1;
                }
                if(location > 0)
                    System.out.println("The Required Number " + look + " Found at = " + location);
                else
                    System.out.println("The Required Number" + look + "Not Found !!");
        }
}
Output :
Enter Number in Array = 6
Enter Number = 1 : : 7
Enter Number = 2 : : 9
Enter Number = 3 : : 10
Enter Number = 4 : : 5
Enter Number = 5 : : 77
Enter Number = 6 : : 22
Enter Number to be Searched = 77
The Required Number 77 Found at = 5
```