

Heap Sort:

```
import java.util.Scanner;
public class HeapSort
{
    public void sort(int arr[])
    {
        int n = arr.length;
        for (int i = n / 2 - 1; i >= 0; i--)
            heapify(arr, n, i);
        for (int i=n-1; i>0; i--)
        {
            int temp = arr[0];
            arr[0] = arr[i];
            arr[i] = temp;
            heapify(arr, i, 0);
        }
    }
    void heapify(int arr[], int n, int i)
    {
        int largest = i;
        int l = 2*i + 1;
        int r = 2*i + 2;
        if (l < n && arr[l] > arr[largest])
            largest = l;
        if (r < n && arr[r] > arr[largest])
            largest = r;
        if (largest != i)
        {
            int swap = arr[i];
            arr[i] = arr[largest];
            arr[largest] = swap;
            heapify(arr, n, largest);
        }
    }
    static void printArray(int arr[])
    {
        int n = arr.length;
        for (int i=0; i<n; ++i)
            System.out.print(arr[i]+" ");
        System.out.println();
    }
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter Heap Size");
        int n = sc.nextInt();
        System.out.println("Enter Heap Elements");
        int arr[] = new int[n];
        for(int i=0;i<n;i++)
        {
            arr[i]=sc.nextInt();
        }
    }
}
```

```
        HeapSort ob = new HeapSort();  
        ob.sort(arr);  
        System.out.println("Sorted array is");  
        printArray(arr);  
    }  
}
```

Output:

```
Enter Heap Size  
5  
Enter Heap Elements  
73  
99  
40  
27  
143  
Sorted array is  
27 40 73 99 143
```

LINEAR SEARCH:

CODE

```
import java.util.Scanner;

public class Main
{
    public static void main (String args[])
    {
        int c, n, search, array[];

        Scanner in = new Scanner (System.in);
        System.out.println ("Enter number of elements");    n =
in.nextInt ();    array = new int[n];
        System.out.println ("Enter those " + n + " elements");

        for (c = 0; c < n; c++)
            array[c] = in.nextInt ();

        System.out.println ("Enter value to find");
        search = in.nextInt ();    for
(c = 0; c < n; c++)
        {
            if (array[c] == search)
            {
                System.out.println (search + " is present at location " +
(c + 1) + ".");    break;
            }
        }
        if (c == n)
            System.out.println (search + " isn't present in array.");
    }
}
```

}

}

Output:

```
Enter number of elements
5
Enter those 5 elements
1
2
3
4
5
Enter value to find
5
5 is present at location 5.
```

```
Enter number of elements
5
Enter those 5 elements
1
2
3
4
5
Enter value to find
6
6 isn't present in array.
```

Binary search:

Code:

```
import java.util.Scanner; public
class Main
{
public static void main(String args[])
{
int counter, num, item, array[], first, last, middle;
Scanner input = new Scanner(System.in);
System.out.println("Enter number of elements:");
num = input.nextInt(); array = new int[num];
System.out.println("Enter " + num + " integers"); for
(counter = 0; counter < num; counter++)
array[counter] = input.nextInt();
```

```
System.out.println("Enter the search value:"); item =
input.nextInt();
first = 0; last = num - 1;
middle = (first + last)/2;
while( first <= last )
{
if ( array[middle] < item ) first =
middle + 1;
else if ( array[middle] == item )
{
System.out.println(item + " found at location " + (middle + 1) +
".");
break;
}
else
{
last = middle - 1;
}
}
```

OUTPUT:

```
Enter number of elements:
5
Enter 5 integers
1 2 3 4 5
Enter the search value:
5
5 found at location 5.

...Program finished with exit code 0
Press ENTER to exit console. □
```

Enter number of elements:

4

Enter 4 integers

1

2

3

4

Enter the search value:

5

5 is not found.

...Program finished with exit code 0

Press ENTER to exit console.