

PROGRAM FOR PRIORITY QUEUE USING ADT

```
import java.util.PriorityQueue;
import java.util.Iterator;
class PriorityQ
{
    public static void main(String[] args)
    {
        PriorityQueue<Integer> numbers = new PriorityQueue<>();
        numbers.add(4);
        numbers.add(2);
        System.out.println("PriorityQueue: " + numbers);
        numbers.offer(1);
        System.out.println("Updated PriorityQueue: " + numbers);
        int number = numbers.peek();
        System.out.println("Accessed Element: " + number);
        boolean result = numbers.remove(2);
        System.out.println("Is the element 2 removed? " + result);
        System.out.println("PriorityQueue: " + numbers);
        number = numbers.poll();
        System.out.println("Removed Element Using poll(): " + number);
        Iterator<Integer> iterate = numbers.iterator();
        while(iterate.hasNext())
        {
            System.out.print(iterate.next());
            System.out.print(", ");
        }
    }
}
```

LINEAR SEARCH

```
import java.io.*;
class LinearSearch
{
public static void main(String args[])
{
int i, n, search, a[];
try
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter number of elements");
n = Integer.parseInt(br.readLine());
a = new int[n];
System.out.println("Enter those " + n + " elements");
for (i = 0; i< n; i++)
a[i] = Integer.parseInt(br.readLine());
System.out.println("Enter value to find");
search = Integer.parseInt(br.readLine());
for (i = 0; i< n; i++)
{
if (a[i] == search)
{
System.out.println(search + " is present at location " + (i+1) + ".");
break;
}
}
if (i == n)
System.out.println(search + " is not present in array.");
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

BINARY SEARCH

```
import java.lang.reflect.Array;
import java.util.Arrays;
import java.io.*;
class BinarySearch
{
public static void main(String args[])
{
int i, first, last, middle, n, search, a[];
try
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter number of elements");
n = Integer.parseInt(br.readLine());
a = new int[n];
System.out.println("Enter " + n + " integers");
for (i = 0; i < n; i++)
a[i] = Integer.parseInt(br.readLine());
Arrays.sort(a);
System.out.println("Sorting Array is :-");
for (i = 0; i < n; i++)
System.out.println(a[i]);
System.out.println("Enter value to find");
search = Integer.parseInt(br.readLine());
first = 0;
last = n - 1;
middle = (first + last)/2;
while( first <= last )
{
if ( a[middle] < search )
first = middle + 1;
else if ( a[middle] == search )
{
System.out.println(search + " found at location " + (middle+1) + ".");
break;
}
else
last = middle - 1;
middle = (first + last)/2;
}
if (first > last)
System.out.println(search + " is not present in the list.\n");
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

SELECTION SORT

```
import java.io.*;
import java.io.*;
class Selectionsort
{
public static void main(String args[])
{
int i,j,n,pos,temp;
try
{
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.print("Enter the number of elements you want to store: ");
n = Integer.parseInt(br.readLine());
System.out.println("Enter the elements of the array: ");
int[] a = new int[n];
for(i=0; i<n; i++)
a[i]= Integer.parseInt(br.readLine());
for(i=0;i<n;i++)
{
pos = smallest(a,n,i);
temp = a[i];
a[i]=a[pos];
a[pos] = temp;
}
System.out.println("\n Printing sorted elements...\n");
for (i=0;i<n;i++)
System.out.println(" "+a[i]);
}
catch(Exception e)
{
System.out.println(e);
}
}
public static int smallest(int a[], int no, int i)
{
int small,pos,j,n;
n=no;
small = a[i];
pos = i;
for(j=i+1;j<n;j++)
{
if(a[j]<small)
{
small = a[j];
pos=j;
}
}
return pos;
} }
```

INSERTION SORT

```
import java.io.*;
class InsertionSort
{
public static void insertionSort(int arr[])
{
int n = arr.length;
for(int j = 1; j < n; j++)
{
int key = arr[j];
int i = j-1;
while ( (i > -1) && ( arr [i] > key ) )
{
arr [i+1] = arr [i];
i--;
}
arr[i+1] = key;
}
}
public static void main(String args[])
{
int n;
try
{
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.print("Enter the number of elements you want to store: ");
n = Integer.parseInt(br.readLine());
int a[]= new int[n];
System.out.println("Enter the elements of the array: ");
for(int i=0; i<n; i++)
a[i]= Integer.parseInt(br.readLine());
System.out.println("Array Before Insertion Sort");
for(int i=0; i<n; i++)
System.out.println(" "+a[i]);
for (int j=1;j<n;j++)
{
int key = a[j];
int i = j-1;
while((i > -1) && ( a [i] > key ))
{
a[i+1] = a[i];
i--;
}
a[i+1] = key;
}
System.out.println("After Insertion Sort");
for(int i=0; i<n; i++)
System.out.println(" "+a[i]);
}
}
```

```
catch(Exception e)
{
System.out.println(e);
}
}
```

BUBBLE SORT

```
import java.io.*;
public class bubblesort
{
    public static void main(String args[])
    {
        int i,j,n,temp;
        try
        {
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            System.out.print("Enter the number of elements you want to store: ");
            n= Integer.parseInt(br.readLine());
            int array[] = new int[n];
            System.out.println("Enter the elements of the array: ");
            for(i=0; i<n; i++)
                array[i]= Integer.parseInt(br.readLine());
            System.out.println("Array Before Bubble Sort");
            for(i=0;i<n;i++)
                System.out.println(array[i]);
            for(i=0;i<n-1;i++)
            {
                for(j=i+1;j<n;j++)
                {
                    if(array[i]>array[j])
                    {
                        temp=array[i];
                        array[i]=array[j];
                        array[j]=temp;
                    }
                }
            }
            System.out.println();
            System.out.println("Array After Bubble Sort");
            for(i=0;i<n;i++)
                System.out.println(array[i]);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```