Assignment 8 Solution

Theory Questions with Answers:

1. Differentiate between:

a. Selection Sort Vs Bubble Sort-

Ans:

Selection Sort	***Bubble Sort***
• In selection sort, the minimum element is selected from the array and swap with an element which is at the beginning of the unsorted sub array.	• In bubble sort, two adjacent elements are compared. If the adjacent elements are not at the correct position, swapping would be performed.
• It is faster than the bubble sort as a lesser number of comparisons is required.	• It is slower than the selection sort as a greater number of comparisons is required.

b. Linear Search Vs Binary Search-

Ans:

Linear Search	***Binary Search***
• The linear search starts searching from the first element and compares each element with a searched element till the element is not found.	• It finds the position of the searched element by finding the middle element of the array.
• In a linear search, the elements don't need to be arranged in sorted order.	• The pre-condition for the binary search is that the elements must be arranged in a sorted order.
• It is preferrable for the small-sized data sets.	• It is preferrable for the large-size data sets.

c. Searching Vs Sorting

Ans:

Searching	***Sorting***
• Searching an array means to find a particular element in the array. The search operation is used to return the position of the element or check if it exists in the array.	• The sorting is a way to arrange elements of a list or array in a certain order. The order may be in ascending or descending order.
• There are two types of searching : Linear Search Binary Search	• Types of Sorting: Selection Sort Bubble Sort

2. **WAP to create function boolean isNumberFound(int n, int[ ] arr) & check whether an element is present in an array. Read the input from the user and call the function.**

```
import java.util.*;
class Search {
   public static boolean isNumberFound(int n, int[ ] arr) {
      int flag=0;
      int l = arr.length;
      for ( int i=0 ; i<l; i++) {
        if ( n==arr[i]) {
          flag =1;
          break;
       }
    }
    if (flag==1){
      return true;
    }
    else{
      return false;
   }
}
  public static void main(String[] args){
     Scanner sc = new Scanner(System.in);
     int n= sc.nextInt();
     int[] A= new int[n];
     for(int i=0;i<n;i++)
        A[i]= sc.nextInt();
     System.out.println("Enter the element to be found");
     int x= sc.nextInt();
     boolean ans = isNumberFound(x,A);
     System.out.println(ans);
  }
}
```

3. WAP to call these functions from the main() function:

- int product(int[] arr)- to return the product of the array elements
- void sortArray(int[] arr)- to sort the passed array in the ascending using selection sort.

Ans:

```
import java.util.*;
class Sort {
   public static int product(int[ ] arr) {
      int p=1;
      int l = arr.length;
      for (int i=0 ; i<l; i++) {
        p=p*arr[i];
    }
    return p;
   }
   private static void sortArray(int[] A ){
      int L = A.length;
      for (int i=0 ;i<L-1; i++) {
           int min = i;
        for (int j=i+1; j<L; j++) {
             if (A[j] < A[min])
               min = j;
         }
        int temp = A[i];
        A[i] = A[min];
        A[min] = temp;
    }
   }
  public static void main(String[] args){
     Scanner sc = new Scanner(System.in);
     int n= sc.nextInt();
     int[] A= new int[n];
     for(int i=0;i<n;i++)
       A[i]= sc.nextInt();
     int product = product(A);
     System.out.println(product);
     sortArray(A);
     for(int i=0;i<n;i++)
       System.out.println(A[i]);
  }
}
```

4. **Write a program to input an integer array sorted in increasing order, and create a function which returns the array of the squares of each number sorted in increasing order.**

**Input: nums = [-4,-1,0,3,10]
**

Output: [0,1,9,16,100]

Explanation: After squaring, the array becomes [16,1,0,9,100].
After sorting, it becomes [0,1,9,16,100].

Ans:

```
import java.util.*;
class SquaresSort {
   public static int[] squareSort(int[ ] arr) {
      int l = arr.length;
      for (int i=0 ; i<l; i++) {
        arr[i]= arr[i]*arr[i];
    }
    Arrays.sort(arr);
    return arr;
   }
   public static void main(String[] args){
     Scanner sc = new Scanner(System.in);
     int n= sc.nextInt();
     int[] A= new int[n];
     for(int i=0;i<n;i++)
       A[i]= sc.nextInt();
     int[] B = squareSort(A);
     for(int i=0;i<n;i++)
       System.out.print(B[i]+" ");
  }
}
```

5. **Predict the output:**

```
int [ ] numbers = { 1, 2, 3, 4, 5, 6, 7,  8, 9, 10};
        for (int j = 0; j < numbers.length; j++)
          System.out.print(numbers[ j ] + " ");
```

Ans: 1 2 3 4 5 6 7 8 9 10