**ASSIGNMENT:4** MID:M1095607

**Data structures and algorithms**

Use bubble sort algorithm to solve below questions

1. Sort below array in ascending order

[ 1 , 4 , 3 , 5 , 2 ]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** BubbleSort1 {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of elements");

**int** numbers=in.nextInt();

System.***out***.println("Enter the elements");

**int** arr[]=**new** **int**[numbers];

**for**(**int** i=0;i<numbers;i++) {

arr[i]=in.nextInt();

}

**int**[] sortedArray=*bubbleSortAlgo*(arr);

System.***out***.println("Sorted Array using bubble sort in ascending order");

**for**(**int** i=0;i<sortedArray.length;i++) {

System.***out***.print(sortedArray[i]+" ");

}

}

**public** **static** **int**[] bubbleSortAlgo(**int**[] arr) {

**for**(**int** i=0;i<arr.length;i++) {

**for**(**int** j=1;j<arr.length-i;j++) {

**if**(arr[j-1]>arr[j]) {

**int** temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

}

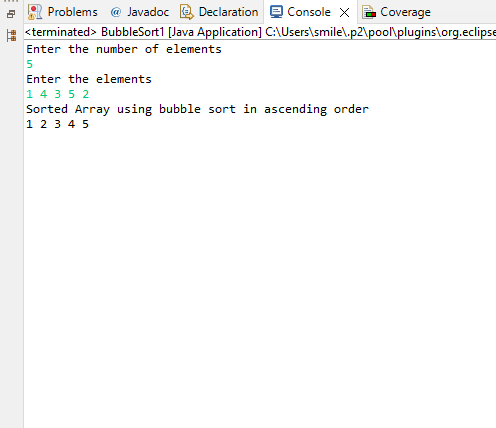
}

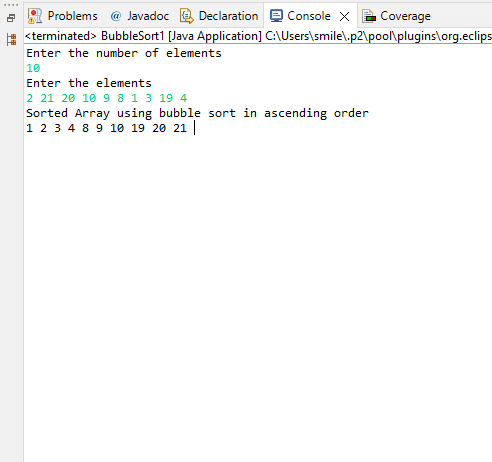
**return** arr;

}

}

**OUTPUT:**

****

****

1. Sort below array in descending order

[ 1 , 4 , 3 , 5 , 2 ]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** BubbleSort1 {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of elements");

**int** numbers=in.nextInt();

System.***out***.println("Enter the elements");

**int** arr[]=**new** **int**[numbers];

**for**(**int** i=0;i<numbers;i++) {

arr[i]=in.nextInt();

}

**int**[] sortedArray=*bubbleSortAlgo*(arr);

System.***out***.println("Sorted Array using bubble sort in descending order");

**for**(**int** i=0;i<sortedArray.length;i++) {

System.***out***.print(sortedArray[i]+" ");

}

}

**public** **static** **int**[] bubbleSortAlgo(**int**[] arr) {

**for**(**int** i=0;i<arr.length;i++) {

**for**(**int** j=1;j<arr.length-i;j++) {

**if**(arr[j-1]<arr[j]) {

**int** temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

}

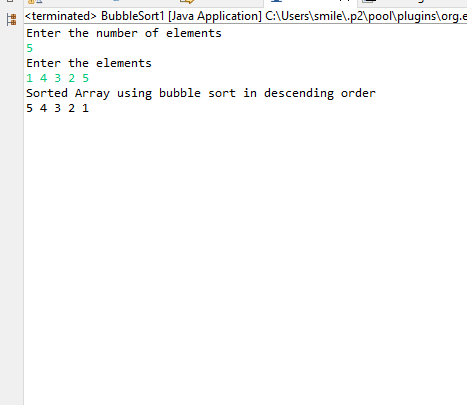
}

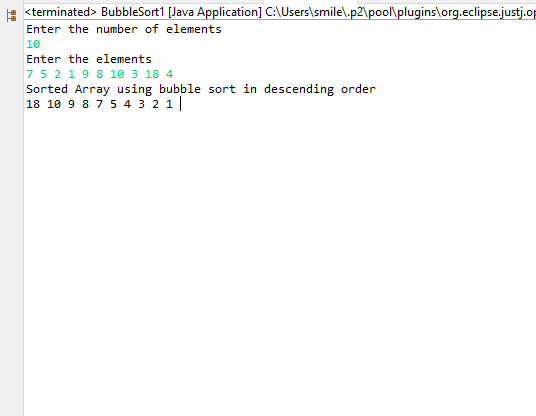
**return** arr;

}

}

**OUTPUT:**





1. Sort below array in ascending order

[Pepsi, Coca-cola, Limka, Fanta, Sprite]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** BubbleSort3 {

**public** **static** **void** main(String args[]) {

Scanner scan=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of Strings");

**int** number=scan.nextInt();

String[] arr=**new** String[number];

System.***out***.println("Enter the Strings");

**for**(**int** i=0;i<number;i++) {

arr[i]=scan.nextLine();

}

String[] sortedArray=*bubbleSortAlgo*(arr);

System.***out***.println("Sorted Array using bubble sort in ascending order");

**for**(**int** i=0;i<sortedArray.length;i++) {

**if**(i<sortedArray.length-1) {

System.***out***.print(sortedArray[i]+",");

}

**else** {

System.***out***.print(sortedArray[i]);

}

}

}

**public** **static** String[] bubbleSortAlgo(String[] arr) {

**for**(**int** i=0;i<arr.length;i++) {

**for**(**int** j=1;j<arr.length-i;j++) {

**if**(arr[j-1].compareTo(arr[j])>0) {

String temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

}

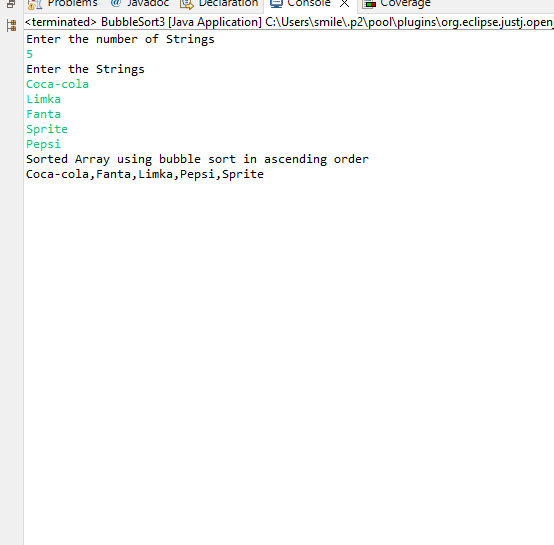
}

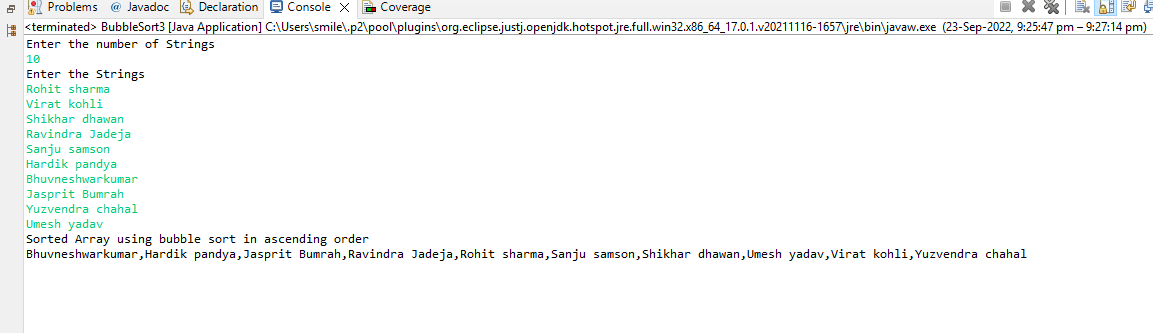
**return** arr;

}

}

**OUTPUT:**





1. Sort below array in descending order

[Pepsi, Coca-cola, Limka, Fanta, Sprite]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.Scanner;

**public** **class** BubbleSort4 {

**public** **static** **void** main(String args[]) {

Scanner scan=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of Strings");

**int** number=scan.nextInt();

String[] arr=**new** String[number];

System.***out***.println("Enter the Strings");

**for**(**int** i=0;i<number;i++) {

arr[i]=scan.nextLine();

}

String[] sortedArray=*bubbleSortAlgo*(arr);

System.***out***.println("Sorted Array using bubble sort in descending order");

**for**(**int** i=0;i<sortedArray.length;i++) {

**if**(i<sortedArray.length-1) {

System.***out***.print(sortedArray[i]+",");

}

**else** {

System.***out***.print(sortedArray[i]);

}

}

}

**public** **static** String[] bubbleSortAlgo(String[] arr) {

**for**(**int** i=0;i<arr.length;i++) {

**for**(**int** j=1;j<arr.length-i;j++) {

**if**(arr[j-1].compareTo(arr[j])<0) {

String temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

}

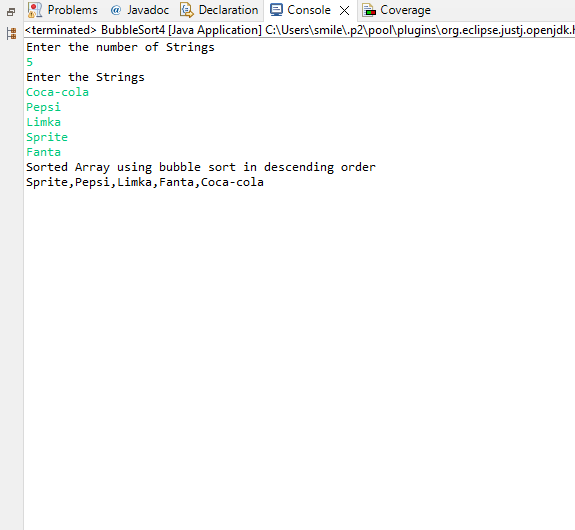
}

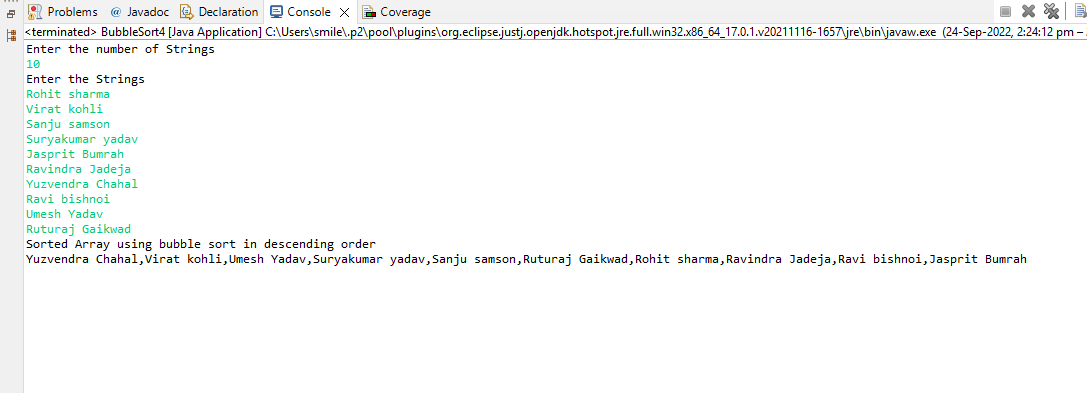
**return** arr;

}

}

**OUTPUT:**

****

****

Use Insertion sort algorithm to solve below questions

1. Sort below array in ascending order

[ 1 , 4 , 3 , 5 , 2 ]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** InsertionSort {

**public** **static** **void** main(String args[]) {

Scanner scan=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of elements");

**int** number=scan.nextInt();

**int** arr[]=**new** **int**[number];

System.***out***.println("Enter the elements");

**for**(**int** i=0;i<number;i++) {

**int** element=scan.nextInt();

arr[i]=element;

}

System.***out***.println("Sorted Array using Insertion sort in ascending order");

**int**[] sortedArray=*insertionSort*(arr);

**for**(**int** i=0;i<sortedArray.length;i++) {

**if**(i<sortedArray.length-1) {

System.***out***.print(sortedArray[i]+",");

}

**else** {

System.***out***.print(sortedArray[i]);

}

}

}

**public** **static** **int**[] insertionSort(**int**[] arr) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**int** j=i;

**while**(j>0) {

**if**(arr[j-1]>arr[j]) {

**int** temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

j=j-1;

}

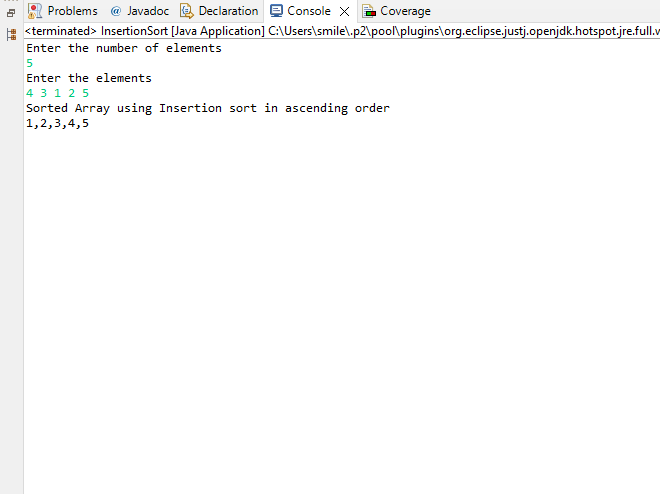
}

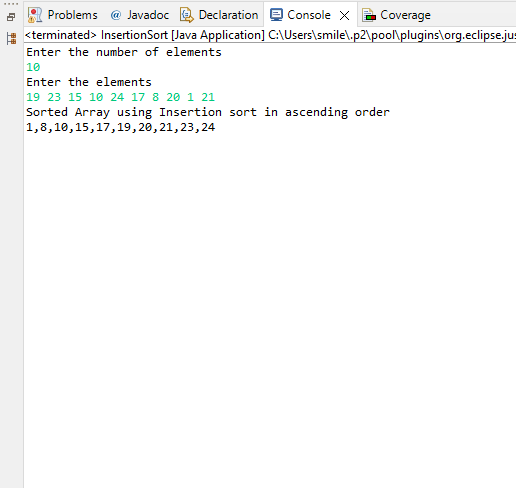
**return** arr;

}

}

**OUTPUT:**

****

****

1. Sort below array in descending order

[ 1 , 4 , 3 , 5 , 2 ]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.Scanner;

**public** **class** InsertionSort2 {

**public** **static** **void** main(String args[]) {

Scanner scan=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of elements");

**int** number=scan.nextInt();

**int** arr[]=**new** **int**[number];

System.***out***.println("Enter the elements");

**for**(**int** i=0;i<number;i++) {

**int** element=scan.nextInt();

arr[i]=element;

}

System.***out***.println("Sorted Array using Insertion sort in descending order");

**int**[] sortedArray=*insertionSort*(arr);

**for**(**int** i=0;i<sortedArray.length;i++) {

**if**(i<sortedArray.length-1) {

System.***out***.print(sortedArray[i]+",");

}

**else** {

System.***out***.print(sortedArray[i]);

}

}

}

**public** **static** **int**[] insertionSort(**int**[] arr) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**int** j=i;

**while**(j>0) {

**if**(arr[j-1]<arr[j]) {

**int** temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

j=j-1;

}

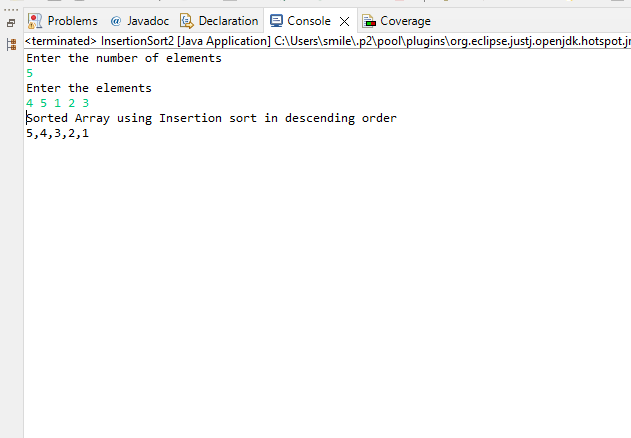
}

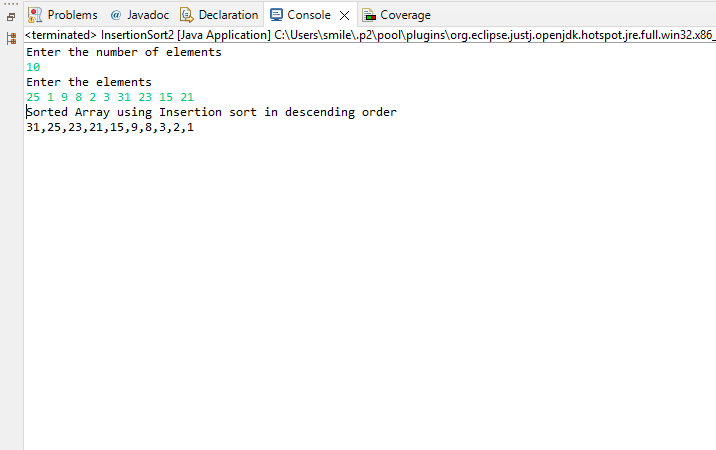
**return** arr;

}

}

**OUTPUT:**

****

****

1. Sort below array in ascending order

[Pepsi, Coca-cola, Limka, Fanta, Sprite]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** InsertionSort3 {

**public** **static** **void** main(String args[]) {

Scanner scan=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of Strings");

**int** number=scan.nextInt();

String arr[]=**new** String[number];

System.***out***.println("Enter the Strings");

**for**(**int** i=0;i<number;i++) {

String element=scan.nextLine();

arr[i]=element;

}

System.***out***.println("Sorted Array using Insertion sort in ascending order");

String[] sortedArray=*insertionSort*(arr);

**for**(**int** i=0;i<sortedArray.length;i++) {

**if**(i<sortedArray.length-1) {

System.***out***.print(sortedArray[i]+",");

}

**else** {

System.***out***.print(sortedArray[i]);

}

}

}

**public** **static** String[] insertionSort(String[] arr) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**int** j=i;

**while**(j>0) {

**if**(arr[j-1].compareTo(arr[j])>0) {

String temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

j=j-1;

}

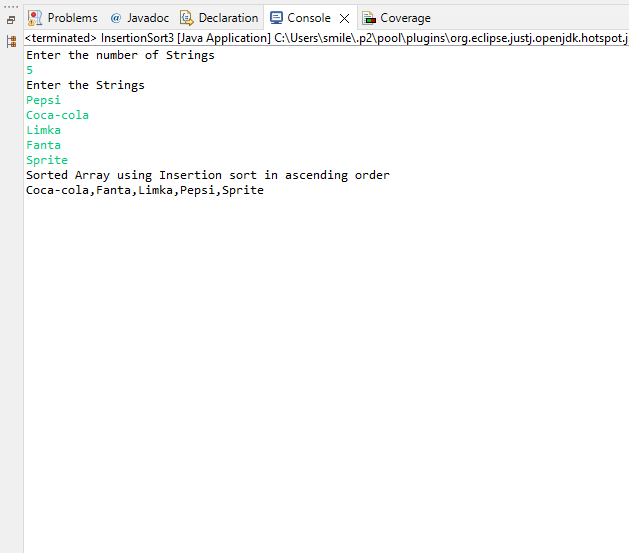
}

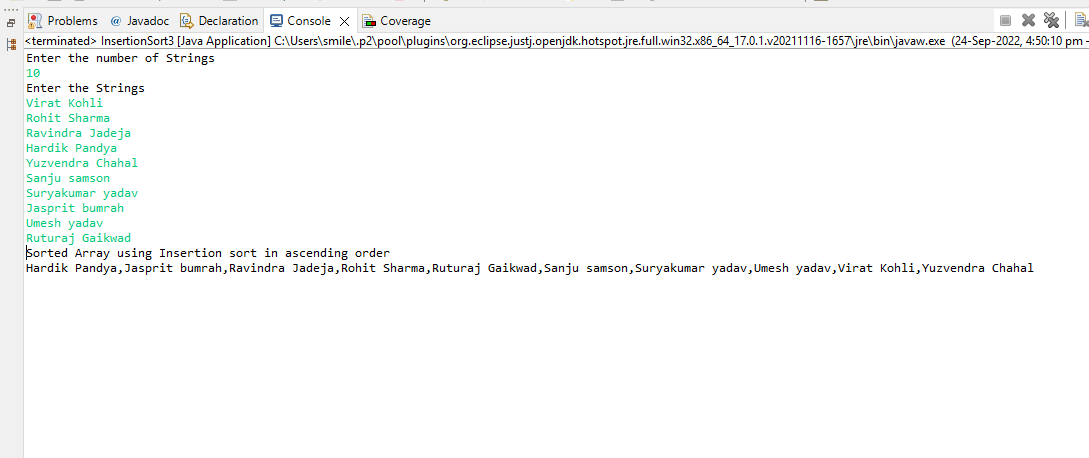
**return** arr;

}

}

**OUTPUT:**

****

****

1. Sort below array in descending order

[Pepsi, Coca-cola, Limka, Fanta, Sprite]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** InsertionSort4 {

**public** **static** **void** main(String args[]) {

Scanner scan=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of Strings");

**int** number=scan.nextInt();

String arr[]=**new** String[number];

System.***out***.println("Enter the Strings");

**for**(**int** i=0;i<number;i++) {

String element=scan.nextLine();

arr[i]=element;

}

System.***out***.println("Sorted Array using Insertion sort in descending order");

String[] sortedArray=*insertionSort*(arr);

**for**(**int** i=0;i<sortedArray.length;i++) {

**if**(i<sortedArray.length-1) {

System.***out***.print(sortedArray[i]+",");

}

**else** {

System.***out***.print(sortedArray[i]);

}

}

}

**public** **static** String[] insertionSort(String[] arr) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**int** j=i;

**while**(j>0) {

**if**(arr[j-1].compareTo(arr[j])<0) {

String temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

j=j-1;

}

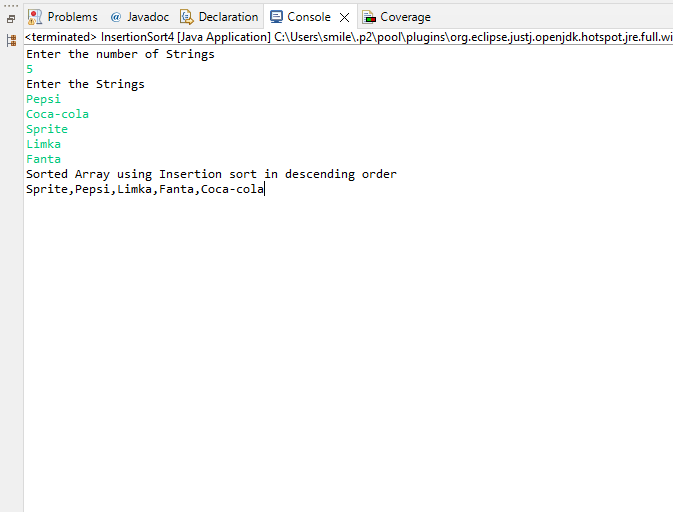
}

**return** arr;

}

}

**OUTPUT:**

****

****

Use Selection sort algorithm to solve below questions

1. Sort below array in ascending order

[ 1 , 4 , 3 , 5 , 2 ]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** SelectionSort {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of elements");

**int** n=in.nextInt();

**int**[] arr=**new** **int**[n];

System.***out***.println("Enter the elements");

**for**(**int** i=0;i<n;i++) {

**int** element=in.nextInt();

arr[i]=element;

}

System.***out***.println("Sorted array using Selection sort in ascending order");

**int**[] sortedArray=*selectionSortAlgo*(arr);

**for**(**int** i=0;i<sortedArray.length;i++) {

**if**(i<n-1) {

System.***out***.print(sortedArray[i]+",");

}

**else** {

System.***out***.print(sortedArray[i]);

}

}

}

**public** **static** **int**[] selectionSortAlgo(**int**[] arr) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**int** min=i;

**for**(**int** j=i;j<len;j++) {

**if**(arr[j]<arr[min]) {

min=j;

}

}

**int** temp=arr[i];

arr[i]=arr[min];

arr[min]=temp;

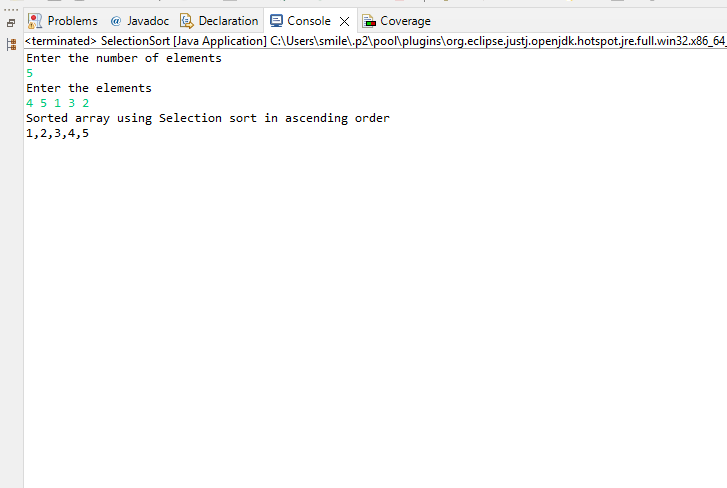
}

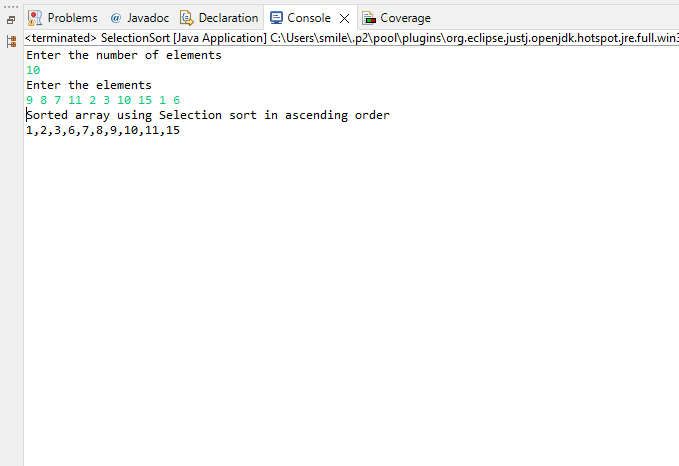
**return** arr;

}

}

**OUTPUT:**

****

****

1. Sort below array in descending order

[ 1 , 4 , 3 , 5 , 2 ]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** SelectionSort2 {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of elements");

**int** n=in.nextInt();

**int**[] arr=**new** **int**[n];

System.***out***.println("Enter the elements");

**for**(**int** i=0;i<n;i++) {

**int** element=in.nextInt();

arr[i]=element;

}

System.***out***.println("Sorted array using Selection sort in descending order");

**int**[] sortedArray=*selectionSortAlgo*(arr);

**for**(**int** i=0;i<sortedArray.length;i++) {

**if**(i<n-1) {

System.***out***.print(sortedArray[i]+",");

}

**else** {

System.***out***.print(sortedArray[i]);

}

}

}

**public** **static** **int**[] selectionSortAlgo(**int**[] arr) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**int** min=i;

**for**(**int** j=i;j<len;j++) {

**if**(arr[j]>arr[min]) {

min=j;

}

}

**int** temp=arr[i];

arr[i]=arr[min];

arr[min]=temp;

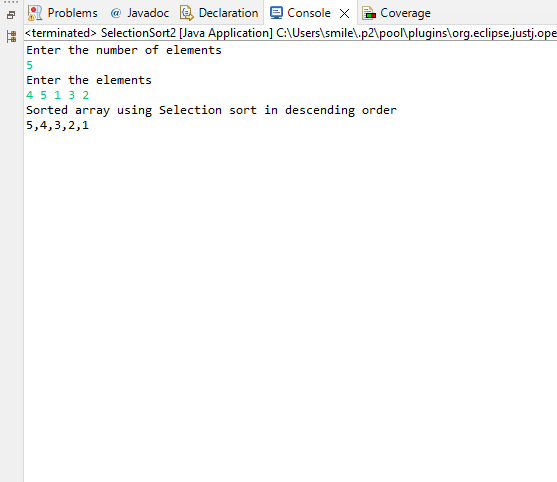
}

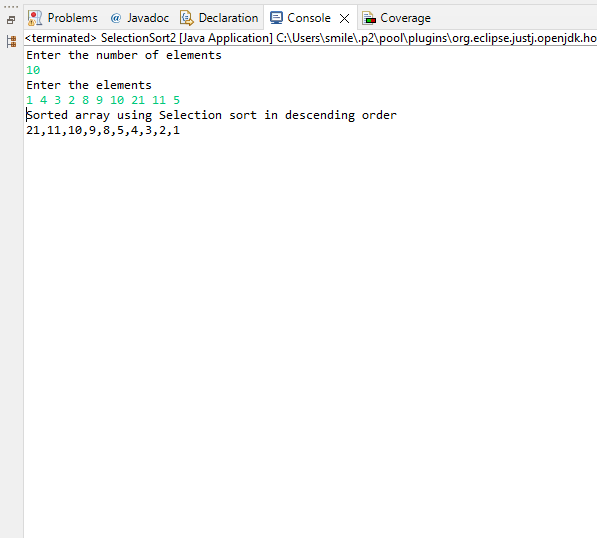
**return** arr;

}

}

**OUTPUT:**

****

****

1. Sort below array in ascending order

[Pepsi, Coca-cola, Limka, Fanta, Sprite]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** SelectionSort3 {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of strings");

**int** n=in.nextInt();

String[] arr=**new** String[n];

System.***out***.println("Enter the string");

Scanner scan=**new** Scanner(System.***in***);

**for**(**int** i=0;i<n;i++) {

String element=scan.nextLine();

arr[i]=element;

}

System.***out***.println("Sorted array using Selection sort in ascending order");

String[] sortedArray=*selectionSortAlgo*(arr);

**for**(**int** i=0;i<sortedArray.length;i++) {

**if**(i<n-1) {

System.***out***.print(sortedArray[i]+",");

}

**else** {

System.***out***.print(sortedArray[i]);

}

}

}

**public** **static** String[] selectionSortAlgo(String[] arr) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**int** min=i;

**for**(**int** j=i;j<len;j++) {

**if**(arr[j].compareTo(arr[min])<0) {

min=j;

}

}

String temp=arr[i];

arr[i]=arr[min];

arr[min]=temp;

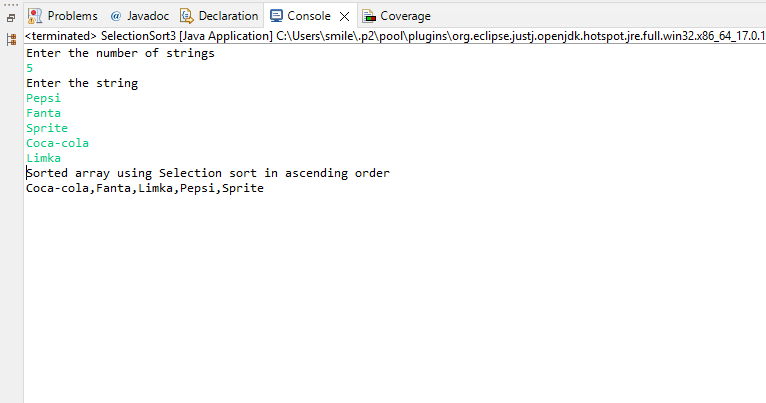
}

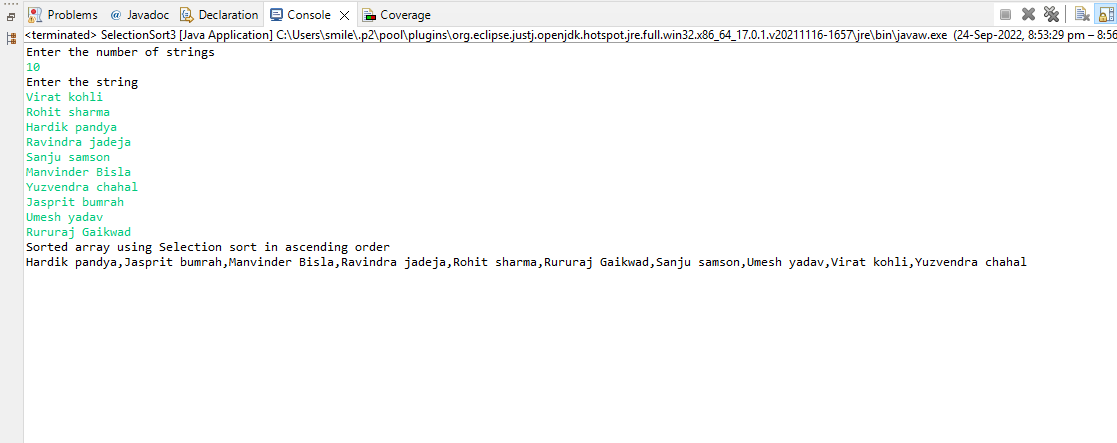
**return** arr;

}

}

**OUTPUT:**

****

****

1. Sort below array in ascending order

[Pepsi, Coca-cola, Limka, Fanta, Sprite]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** SelectionSort4 {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of strings");

**int** n=in.nextInt();

String[] arr=**new** String[n];

System.***out***.println("Enter the string");

Scanner scan=**new** Scanner(System.***in***);

**for**(**int** i=0;i<n;i++) {

String element=scan.nextLine();

arr[i]=element;

}

System.***out***.println("Sorted array using Selection sort in descending order");

String[] sortedArray=*selectionSortAlgo*(arr);

**for**(**int** i=0;i<sortedArray.length;i++) {

**if**(i<n-1) {

System.***out***.print(sortedArray[i]+",");

}

**else** {

System.***out***.print(sortedArray[i]);

}

}

}

**public** **static** String[] selectionSortAlgo(String[] arr) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**int** min=i;

**for**(**int** j=i;j<len;j++) {

**if**(arr[j].compareTo(arr[min])>0) {

min=j;

}

}

String temp=arr[i];

arr[i]=arr[min];

arr[min]=temp;

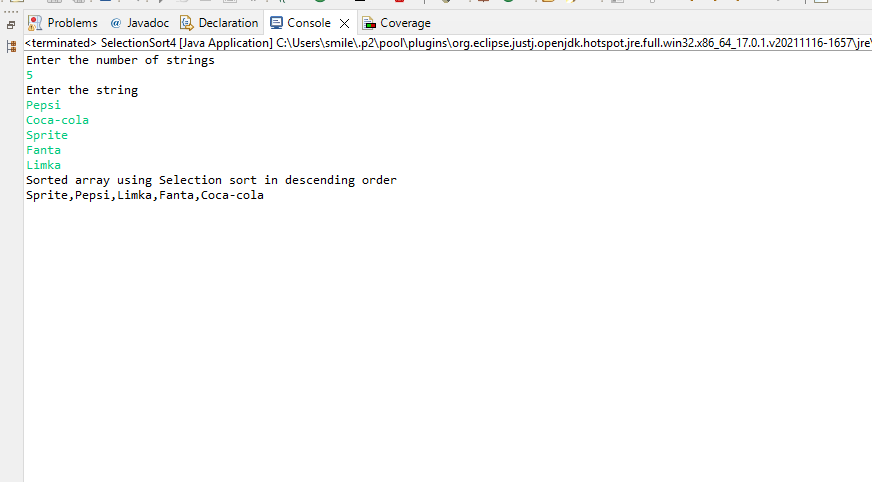
}

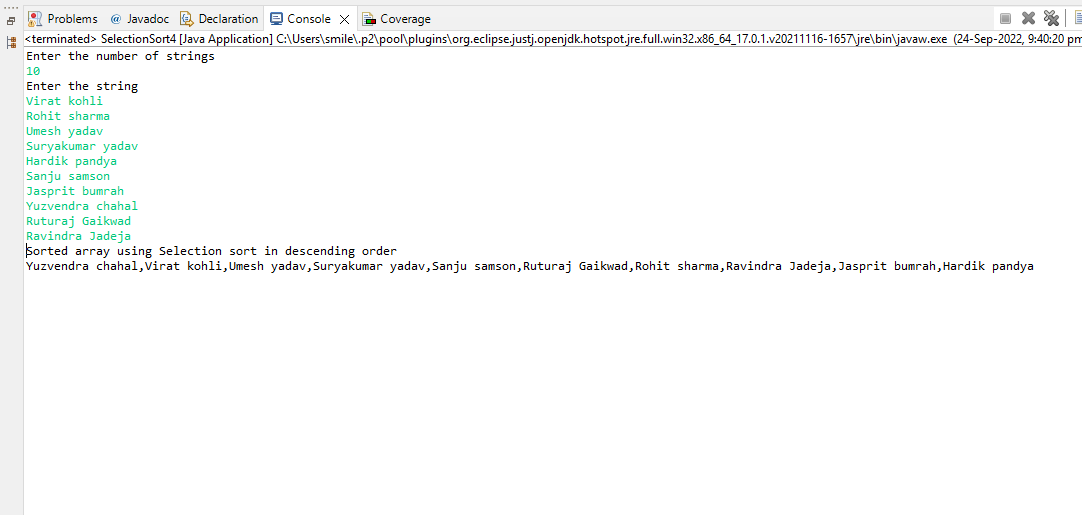
**return** arr;

}

}

**OUTPUT:**

****

****

Use Linear search algorithm to solve below questions

1. Search index of 5 from below array.  
    [1 ,4, 3, 5, 2]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** LinearSearch {

**public** **static** **void** main(String args[]) {

System.***out***.println("Enter the number of elements");

Scanner scan=**new** Scanner(System.***in***);

**int** number=scan.nextInt();

**int**[] arr=**new** **int**[number];

System.***out***.println("Enter the elements");

**for**(**int** i=0;i<number;i++) {

arr[i]=scan.nextInt();

}

System.***out***.println("Enter the element to search");

**int** k=scan.nextInt();

**int** index=*linearSearchAlgo*(arr,k);

**if**(index==-1) {

System.***out***.println("Element Not found in the list");

}

**else** {

System.***out***.println("Index of "+k+" is "+index);

}

}

**public** **static** **int** linearSearchAlgo(**int**[] arr,**int** element) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**if**(arr[i]==element) {

**return** i;

}

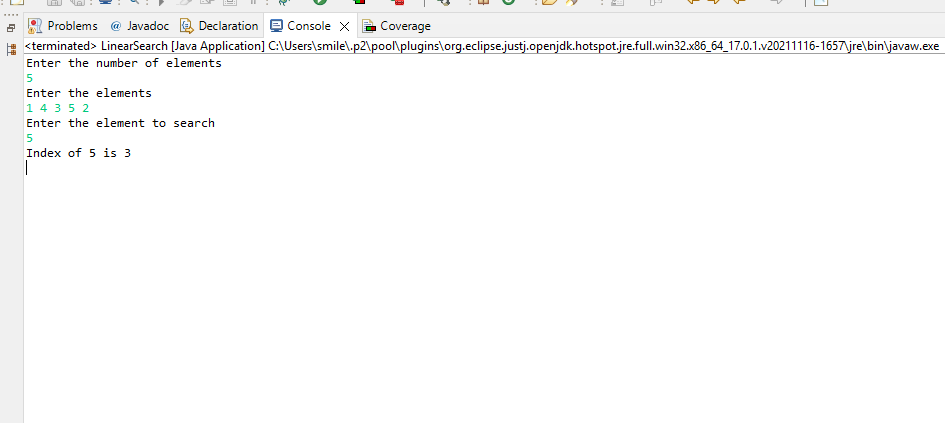
}

**return** -1;

}

}

**OUTPUT:**

****

1. Search index of 8 from below array.  
    [1 ,4, 3, 5, 2]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** LinearSearch {

**public** **static** **void** main(String args[]) {

System.***out***.println("Enter the number of elements");

Scanner scan=**new** Scanner(System.***in***);

**int** number=scan.nextInt();

**int**[] arr=**new** **int**[number];

System.***out***.println("Enter the elements");

**for**(**int** i=0;i<number;i++) {

arr[i]=scan.nextInt();

}

System.***out***.println("Enter the element to search");

**int** k=scan.nextInt();

**int** index=*linearSearchAlgo*(arr,k);

**if**(index==-1) {

System.***out***.println("Element Not found in the list");

}

**else** {

System.***out***.println("Index of "+k+" is "+index);

}

}

**public** **static** **int** linearSearchAlgo(**int**[] arr,**int** element) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**if**(arr[i]==element) {

**return** i;

}

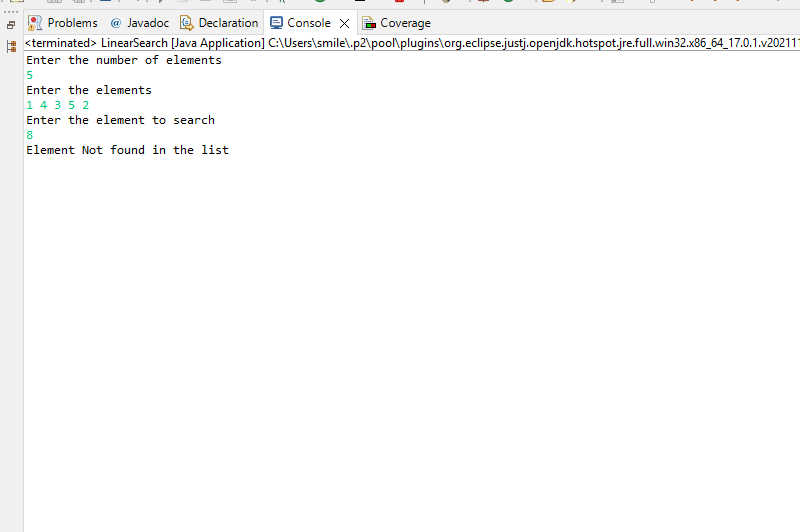
}

**return** -1;

}

}

**OUTPUT:**

****

1. Search index of “Sprite” from below array.

[Pepsi, Coca-cola, Limka, Fanta, Sprite]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** LinearSearch {

**public** **static** **void** main(String args[]) {

System.***out***.println("Enter the number of elements");

Scanner scan=**new** Scanner(System.***in***);

**int** number=scan.nextInt();

String[] arr=**new** String[number];

System.***out***.println("Enter the elements");

Scanner scan1=**new** Scanner(System.***in***);

**for**(**int** i=0;i<number;i++) {

arr[i]=scan1.nextLine();

}

Scanner scan2=**new** Scanner(System.***in***);

System.***out***.println("Enter the element to search");

String k=scan2.nextLine();

**int** index=*linearSearchAlgo*(arr,k);

**if**(index==-1) {

System.***out***.println("Element Not found in the list");

}

**else** {

System.***out***.println("Index of "+k+" is "+index);

}

}

**public** **static** **int** linearSearchAlgo(String[] arr,String element) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**if**(arr[i].equals(element)) {

**return** i;

}

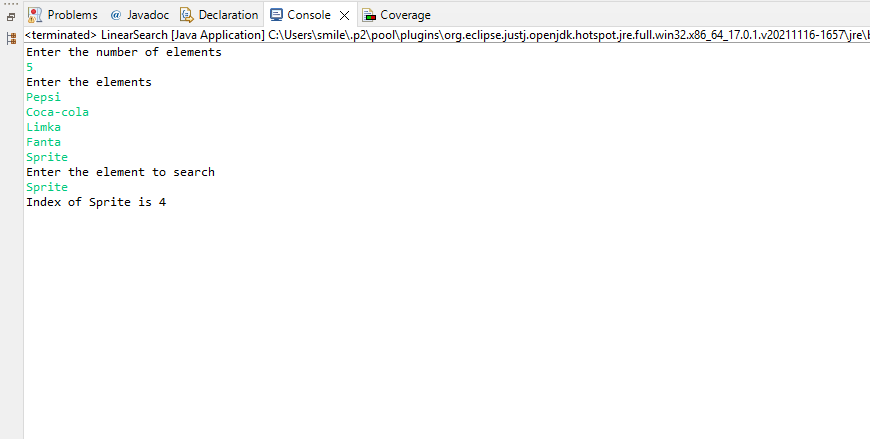
}

**return** -1;

}

}

**OUTPUT:**

****

4. Search index of “Slice” from below array.

[Pepsi, Coca-cola, Limka, Fanta, Sprite]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** LinearSearch {

**public** **static** **void** main(String args[]) {

System.***out***.println("Enter the number of elements");

Scanner scan=**new** Scanner(System.***in***);

**int** number=scan.nextInt();

String[] arr=**new** String[number];

System.***out***.println("Enter the elements");

Scanner scan1=**new** Scanner(System.***in***);

**for**(**int** i=0;i<number;i++) {

arr[i]=scan1.nextLine();

}

Scanner scan2=**new** Scanner(System.***in***);

System.***out***.println("Enter the element to search");

String k=scan2.nextLine();

**int** index=*linearSearchAlgo*(arr,k);

**if**(index==-1) {

System.***out***.println("Element Not found in the list");

}

**else** {

System.***out***.println("Index of "+k+" is "+index);

}

}

**public** **static** **int** linearSearchAlgo(String[] arr,String element) {

**int** len=arr.length;

**for**(**int** i=0;i<len;i++) {

**if**(arr[i].equals(element)) {

**return** i;

}

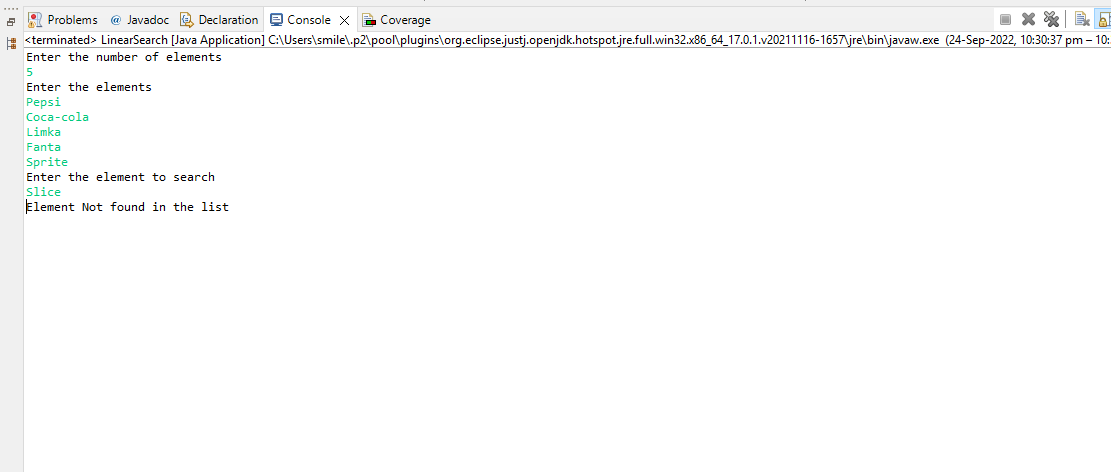
}

**return** -1;

}

}

**OUTPUT:**

****

Use binary search algorithm to solve below questions

1. Search index of 5 from below array

[1 , 4 , 3 , 5 , 2]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** BinarySearch {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of elements");

**int** n=in.nextInt();

System.***out***.println("Enter the elements");

**int**[] arr=**new** **int**[n];

**for**(**int** i=0;i<n;i++) {

arr[i]=in.nextInt();

}

System.***out***.println("Enter the element to search");

**int** k=in.nextInt();

**int** index=*binarySearchAlgo*(arr,k);

**if**(index==-1) {

System.***out***.println("Element not found in the list");

}

**else** {

System.***out***.println("Index of "+k+" is "+index);

}

}

**public** **static** **int** binarySearchAlgo(**int**[] arr,**int** search) {

//Sort the array before performing binary search

**for**(**int** i=0;i<arr.length;i++) {

**for**(**int** j=1;j<arr.length-i;j++) {

**if**(arr[j-1]>arr[j]) {

**int** temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

}

}

System.***out***.println("Sorted Array");

**for**(**int** i=0;i<arr.length;i++) {

System.***out***.print(arr[i]+" ");

}

System.***out***.println();

//Binary Search

**int** first=0;**int** last=arr.length-1;

**for**(**int** i=0;i<arr.length;i++) {

**int** mid=(first+last)/2;

**if**(arr[mid]==search) {

**return** mid;

}

**else** **if**(search>arr[mid]) {

first=mid+1;

}

**else** **if**(search<arr[mid]) {

last=mid-1;

}

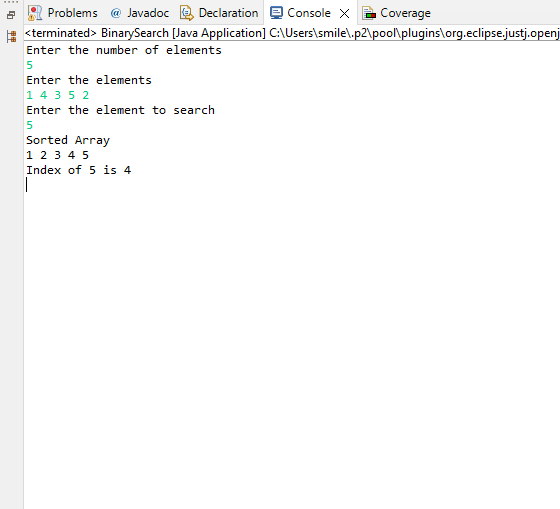
}

**return** -1;

}

}

**OUTPUT:**

****

1. Search index of 8 from below array.  
    [1 ,4, 3, 5, 2]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** BinarySearch {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of elements");

**int** n=in.nextInt();

System.***out***.println("Enter the elements");

**int**[] arr=**new** **int**[n];

**for**(**int** i=0;i<n;i++) {

arr[i]=in.nextInt();

}

System.***out***.println("Enter the element to search");

**int** k=in.nextInt();

**int** index=*binarySearchAlgo*(arr,k);

**if**(index==-1) {

System.***out***.println("Element not found in the list");

}

**else** {

System.***out***.println("Index of "+k+" is "+index);

}

}

**public** **static** **int** binarySearchAlgo(**int**[] arr,**int** search) {

//Sort the array before performing binary search

**for**(**int** i=0;i<arr.length;i++) {

**for**(**int** j=1;j<arr.length-i;j++) {

**if**(arr[j-1]>arr[j]) {

**int** temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

}

}

System.***out***.println("Sorted Array");

**for**(**int** i=0;i<arr.length;i++) {

System.***out***.print(arr[i]+" ");

}

System.***out***.println();

//Binary Search

**int** first=0;**int** last=arr.length-1;

**for**(**int** i=0;i<arr.length;i++) {

**int** mid=(first+last)/2;

**if**(arr[mid]==search) {

**return** mid;

}

**else** **if**(search>arr[mid]) {

first=mid+1;

}

**else** **if**(search<arr[mid]) {

last=mid-1;

}

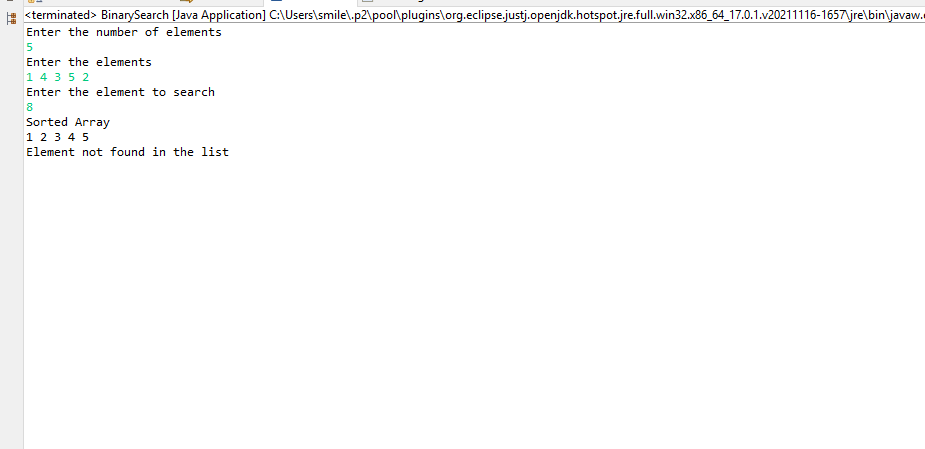
}

**return** -1;

}

}

**OUTPUT:**

****

1. Search index of “Sprite” from below array.

[Pepsi, Coca-cola, Limka, Fanta, Sprite]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** BinarySearch {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of Strings");

**int** n=in.nextInt();

System.***out***.println("Enter the Strings");

String[] arr=**new** String[n];

Scanner in1=**new** Scanner(System.***in***);

**for**(**int** i=0;i<n;i++) {

arr[i]=in1.nextLine();

}

Scanner in2=**new** Scanner(System.***in***);

System.***out***.println("Enter the element to search");

String k=in2.nextLine();

**int** index=*binarySearchAlgo*(arr,k);

**if**(index==-1) {

System.***out***.println("Element not found in the list");

}

**else** {

System.***out***.println("Index of "+k+" is "+index);

}

}

**public** **static** **int** binarySearchAlgo(String[] arr,String search) {

//Sort the array before performing binary search

**for**(**int** i=0;i<arr.length;i++) {

**for**(**int** j=1;j<arr.length-i;j++) {

**if**(arr[j-1].compareTo(arr[j])>0) {

String temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

}

}

System.***out***.println("Sorted Array");

**for**(**int** i=0;i<arr.length;i++) {

System.***out***.print(arr[i]+" ");

}

System.***out***.println();

//Binary Search

**int** first=0;**int** last=arr.length-1;

**for**(**int** i=0;i<arr.length;i++) {

**int** mid=(first+last)/2;

**if**(arr[mid].equals(search)) {

**return** mid;

}

**else** **if**(search.compareTo(arr[mid])>0) {

first=mid+1;

}

**else** **if**(search.compareTo(arr[mid])<0) {

last=mid-1;

}

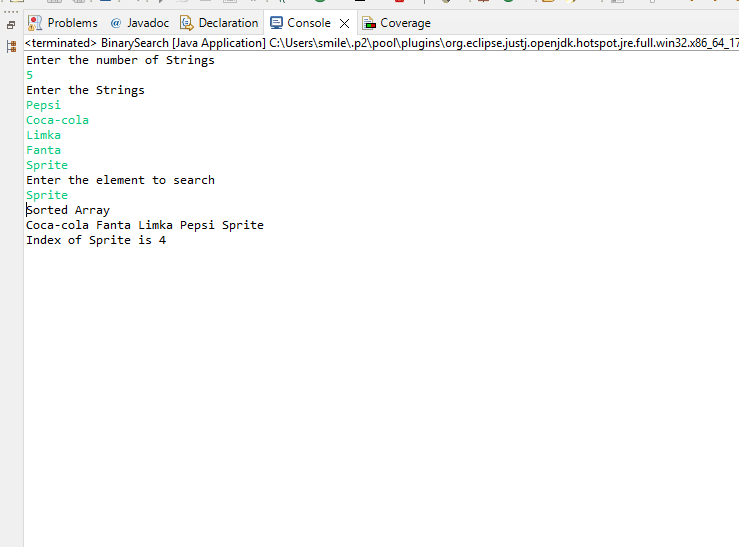
}

**return** -1;

}

}

**OUTPUT:**

****

4. Search index of “Slice” from below array.

[Pepsi, Coca-cola, Limka, Fanta, Sprite]

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** BinarySearch {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of Strings");

**int** n=in.nextInt();

System.***out***.println("Enter the Strings");

String[] arr=**new** String[n];

Scanner in1=**new** Scanner(System.***in***);

**for**(**int** i=0;i<n;i++) {

arr[i]=in1.nextLine();

}

Scanner in2=**new** Scanner(System.***in***);

System.***out***.println("Enter the element to search");

String k=in2.nextLine();

**int** index=*binarySearchAlgo*(arr,k);

**if**(index==-1) {

System.***out***.println("Element not found in the list");

}

**else** {

System.***out***.println("Index of "+k+" is "+index);

}

}

**public** **static** **int** binarySearchAlgo(String[] arr,String search) {

//Sort the array before performing binary search

**for**(**int** i=0;i<arr.length;i++) {

**for**(**int** j=1;j<arr.length-i;j++) {

**if**(arr[j-1].compareTo(arr[j])>0) {

String temp=arr[j-1];

arr[j-1]=arr[j];

arr[j]=temp;

}

}

}

System.***out***.println("Sorted Array");

**for**(**int** i=0;i<arr.length;i++) {

System.***out***.print(arr[i]+" ");

}

System.***out***.println();

//Binary Search

**int** first=0;**int** last=arr.length-1;

**for**(**int** i=0;i<arr.length;i++) {

**int** mid=(first+last)/2;

**if**(arr[mid].equals(search)) {

**return** mid;

}

**else** **if**(search.compareTo(arr[mid])>0) {

first=mid+1;

}

**else** **if**(search.compareTo(arr[mid])<0) {

last=mid-1;

}

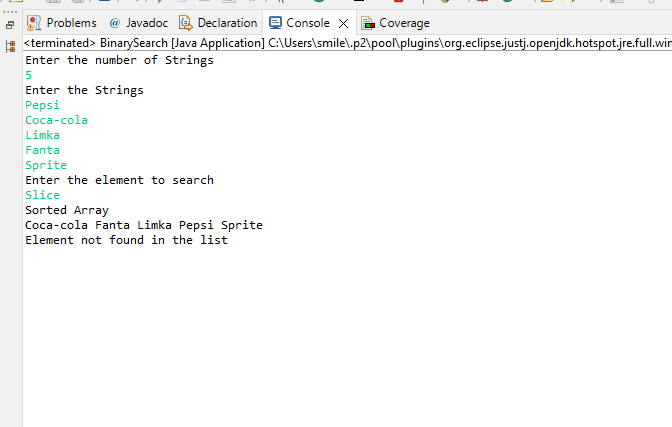
}

**return** -1;

}

}

**OUTPUT:**

****

Use 2D array data structure to solve below questions

1. Traverse and print array row-wise

1 4 6 8

2 4 6 7

7 8 5 1

Output: 1,4,6,8,2,4,6,7,7,8,5,1

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** Traverse {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of rows");

**int** rows=in.nextInt();

System.***out***.println("Enter the number of columns");

**int** columns=in.nextInt();

**int** arr[][]=**new** **int**[rows][columns];

System.***out***.println("Enter the elements");

**for**(**int** i=0;i<rows;i++) {

**for**(**int** j=0;j<columns;j++) {

arr[i][j]=in.nextInt();

}

}

System.***out***.println("Array in row wise");

**for**(**int** i=0;i<rows;i++) {

**for**(**int** j=0;j<columns;j++) {

**if**(i==rows-1 && j==columns-1) {

System.***out***.print(arr[i][j]);

}

**else** {

System.***out***.print(arr[i][j]+",");

}

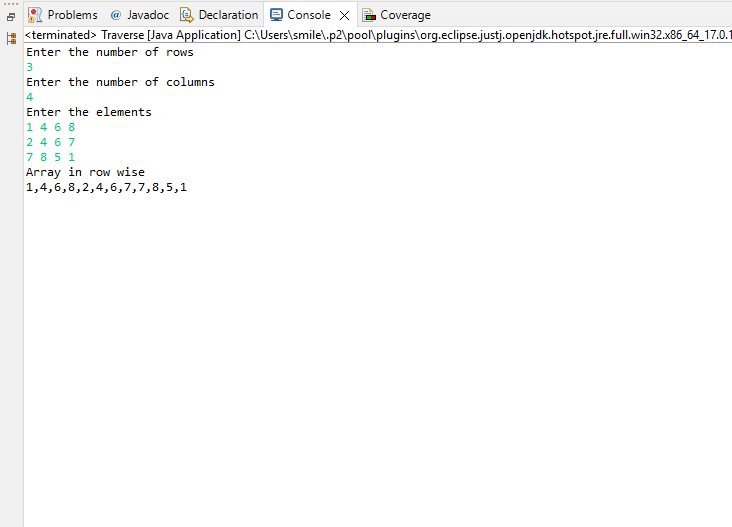
}

}

}

}

**OUTPUT:**

****

1. Traverse and print below array column wise

1 4 6 8

2 4 6 7

7 8 5 1

Output: 1,2,7,4,4,8,6,6,5,8,7,1

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.Scanner;

**public** **class** Traverse2 {

**public** **static** **void** main(String args[]) {

Scanner in=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of rows");

**int** rows=in.nextInt();

System.***out***.println("Enter the number of columns");

**int** columns=in.nextInt();

**int** arr[][]=**new** **int**[rows][columns];

System.***out***.println("Enter the elements");

**for**(**int** i=0;i<rows;i++) {

**for**(**int** j=0;j<columns;j++) {

arr[i][j]=in.nextInt();

}

}

System.***out***.println("Array in column wise");

**for**(**int** i=0;i<columns;i++) {

**for**(**int** j=0;j<rows;j++) {

**if**(i==columns-1 && j==rows-1) {

System.***out***.print(arr[j][i]);

}

**else** {

System.***out***.print(arr[j][i]+",");

}

}

}

}

}

**OUTPUT:**

****

**Use stack data structures to solve below questions**

1. Covert below array into stack, then traverse the newly created stack

int[] a={1, 9 , 3, 5, 4, 6, 2, 8};

1. Find the position of 5 in newly created stack from question number 1
2. Find the middle element of the same stack
3. What are the numbers which are lesser than middle element and will be popped before middle element?

Use another stack to store those elements before printing

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.\*;

**public** **class** Stack {

**private** **int** max;

**private** **int**[] arr;

**private** **int** top;

Stack(){

**this**.max=100;

**this**.top=-1;

**this**.arr=**new** **int**[max];

}

**public** **int** push(**int** element) {

top+=1;

arr[top]=element;

**return** element;

}

**public** **void** pop() {

**if**(top==-1) {

System.***out***.println("Stack is empty");

}

**else** {

System.***out***.println("Element Popped "+ arr[top]);

arr[top]=-1;

top=top-1;

}

}

**public** **void** displayStack() {

**if**(top>=0) {

System.***out***.print("Stack is: ");

**for**(**int** i=0;i<=top;i++) {

System.***out***.print(arr[i]+" ");

}

System.***out***.println();

}

**else** {

System.***out***.println("Stack is empty");

}

}

**public** **void** position(**int** element) {

**int** k=0;

**for**(**int** i=0;i<top;i++) {

**if**(element==arr[i]) {

k=1;

System.***out***.println("Index of "+element+" is "+i);

**break**;

}

}

**if**(k==0) {

System.***out***.println("Element not found");

}

}

**public** **void** middleElement() {

**int** middle=(top+1)/2;

System.***out***.println("Middle element is "+middle);

}

**public** **int**[] lessMiddlePop() {

**int** middle=(top+1)/2;**int** len=0;

**for**(**int** i=middle+1;i<=top;i++) {

**if**(arr[i]<arr[middle]) {

len++;

}

}

**int**[] newList=**new** **int**[len];

**if**(len>0) {

**int** p=0;

**for**(**int** i=middle+1;i<=top;i++) {

**if**(arr[i]<arr[middle]) {

newList[p]=arr[i];

p++;

}

}

}

**return** newList;

}

**public** **static** **void** main(String args[]) {

Stack stack=**new** Stack();

**boolean** cond=**true**;

Scanner in=**new** Scanner(System.***in***);

**do** {

System.***out***.println("1. Push a element");

System.***out***.println("2. Pop a element");

System.***out***.println("3. Display Stack elements");

System.***out***.println("4. Position of the element");

System.***out***.println("5. Middle Element");

System.***out***.println("6. Elements less than middle element and that are popped");

System.***out***.println("7. Exit");

System.***out***.println("Enter your option");

**int** option=in.nextInt();

**switch**(option) {

**case** 1:

System.***out***.println("Enter a element to push");

**int** element=in.nextInt();

**if**(stack.top==stack.max-1) {

System.***out***.println("Stack is full");

}

stack.push(element);

**break**;

**case** 2:

stack.pop();

**break**;

**case** 3:

stack.displayStack();

**break**;

**case** 4:

System.***out***.println("Enter the element to find the position");

**int** newelement=in.nextInt();

stack.position(newelement);

**break**;

**case** 5:

stack.middleElement();

**break**;

**case** 6:

**int**[] newArray=stack.lessMiddlePop();

**if**(newArray.length==0) {

System.***out***.println("No elements found");

}

**else** {

System.***out***.println("Elements less than middle and that will be popped first are");

**for**(**int** i=0;i<newArray.length;i++) {

System.***out***.print(newArray[i]+" ");

}

System.***out***.println();

}

**break**;

**case** 7:

System.***out***.println("Exiting");

cond=**false**;

**break**;

**default**:

System.***out***.println("Enter a valid option");

**break**;

}

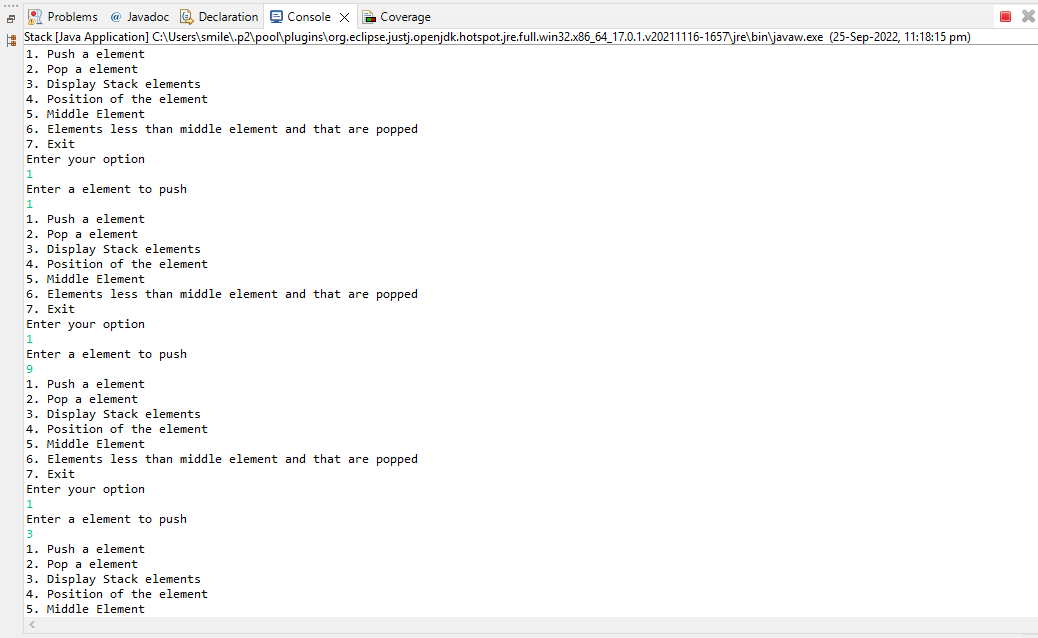
}**while**(cond);

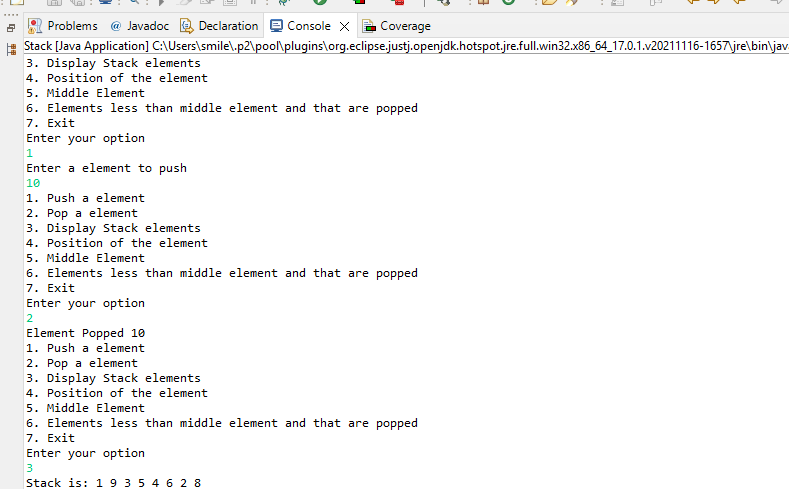
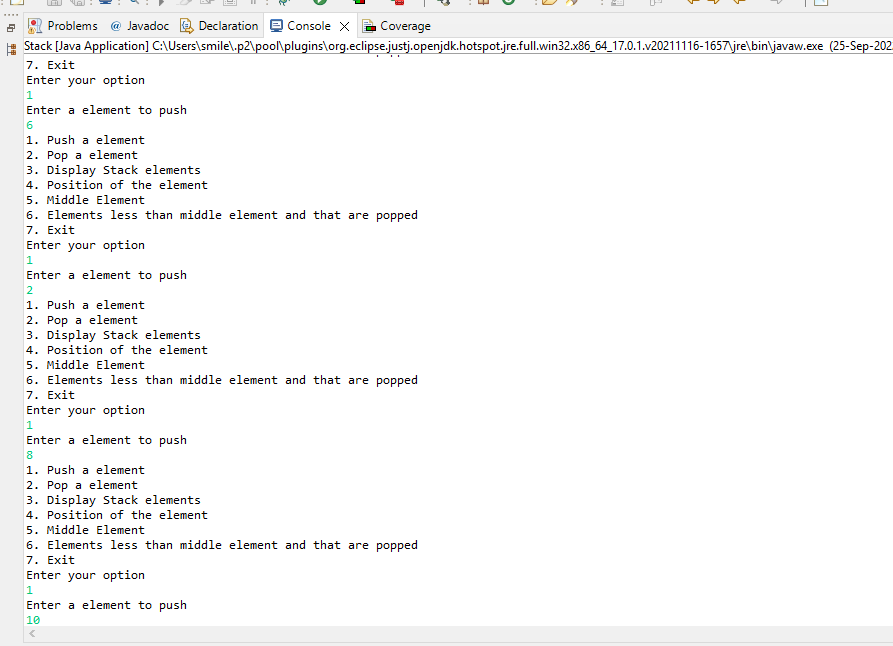
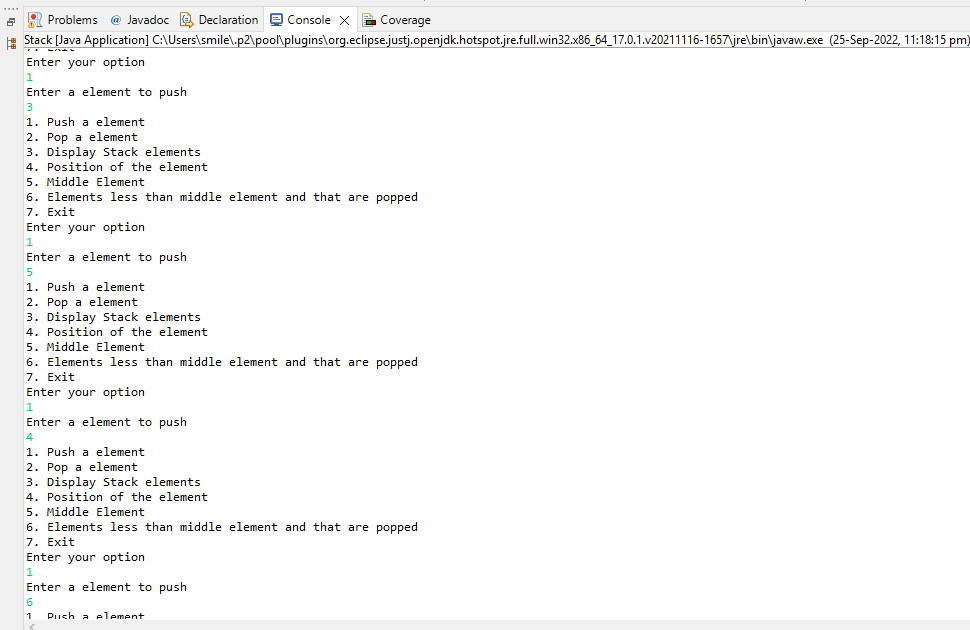
}

}

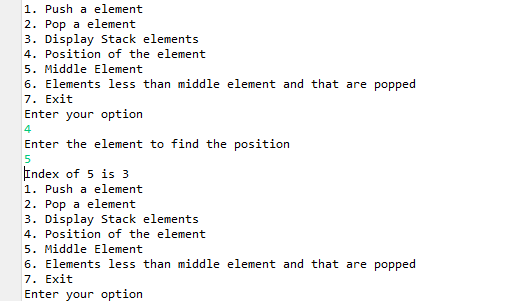
**OUTPUT:**

1.

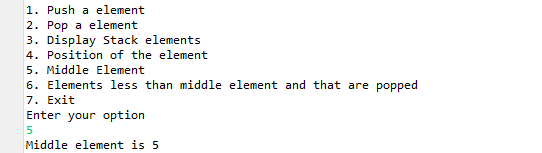




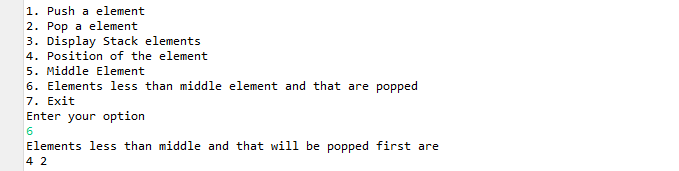
2.



3.



4.

****

**Use Queue data structures to solve below questions**

1. Covert below array into queue, then traverse the newly created queue

int[] a={1, 9 , 3, 5, 4, 6, 2, 8};

1. Find the position of 5 in newly created queue from question number 1
2. Find the middle element of the same queue
3. What are the numbers which are lesser than middle element and will be encountered before middle element?

Use another queue to store those elements before printing

**CODE:**

**package** com.mindtree.datastructures;

**import** java.util.Scanner;

**public** **class** Queue {

**private** **int** max;

**private** **int**[] arr;

**private** **int** front;

**private** **int** rear;

Queue(){

**this**.max=100;

**this**.arr=**new** **int**[max];

**this**.front=-1;

**this**.rear=-1;

}

**public** **void** enQueue(**int** element) {

**if**(rear==max) {

System.***out***.println("Queue is full");

}

**else** {

**if**(front==-1) {

front=0;

}

rear++;

arr[rear]=element;

System.***out***.println("Element "+element +" Pushed");

}

}

**public** **void** deQueue() {

**if**(front==-1 && rear==-1) {

System.***out***.println("Queue is empty");

}

**else** {

**if**(front>rear) {

front=-1;rear=-1;

}

**else** {

System.***out***.println("DeQueued");

arr[front]=-1;

front++;

}

}

}

**public** **void** displayQueue() {

**for**(**int** i=front;i<=rear;i++) {

System.***out***.print(arr[i]+" ");

}

}

**public** **void** position(**int** element) {

**int** p=-1;

**for**(**int** i=front;i<=rear;i++) {

p++;

**if**(element==arr[i]) {

System.***out***.println("Index of "+element+" is "+p);

**break**;

}

}

**if**(p==-1) {

System.***out***.println("Element not found");

}

}

**public** **int** middleElement() {

**int** mid=(front+rear)/2;

**if**(mid>=0) {

**return** arr[mid];

}

**else** {

**return** -1;

}

}

**public** **int**[] middleEncountered() {

**int** n=(front+rear)/2;**int** p=0;

**for**(**int** i=front;i<n;i++) {

**if**(arr[i]<arr[n]) {

p++;

}

}

**int** index=0;

**int**[] newArray=**new** **int**[p];

**for**(**int** i=front;i<n;i++) {

**if**(arr[i]<arr[n]) {

newArray[index]=arr[i];

index++;

}

}

**return** newArray;

}

**public** **static** **void** main(String args[]) {

Queue queue=**new** Queue();

**boolean** cond=**true**;

Scanner in=**new** Scanner(System.***in***);

**do** {

System.***out***.println("1. EnQueue");

System.***out***.println("2. DeQueue");

System.***out***.println("3. Display Queue elements");

System.***out***.println("4. Position of the element");

System.***out***.println("5. Middle Element");

System.***out***.println("6. Elements less than middle element and that are encountered");

System.***out***.println("7. Exit");

System.***out***.println("Enter your option");

**int** option=in.nextInt();

**switch**(option) {

**case** 1:

System.***out***.println("Enter a element to enqueue");

**int** element=in.nextInt();

queue.enQueue(element);

**break**;

**case** 2:

queue.deQueue();

**break**;

**case** 3:

System.***out***.print("Queue is ");

queue.displayQueue();

System.***out***.println();

**break**;

**case** 4:

System.***out***.println("Enter the element to find the position");

**int** newelement=in.nextInt();

queue.position(newelement);

**break**;

**case** 5:

**if**(queue.middleElement()>=0) {

System.***out***.println("Middle Element is "+queue.middleElement());

}

**break**;

**case** 6:

**int**[] newArray=queue.middleEncountered();

**if**(newArray.length==0) {

System.***out***.println("No elements found");

}

**else** {

System.***out***.println("Elements less than middle and that will be encountered first are");

**for**(**int** i=0;i<newArray.length;i++) {

System.***out***.print(newArray[i]+" ");

}

System.***out***.println();

}

**break**;

**case** 7:

System.***out***.println("Exiting");

cond=**false**;

**break**;

**default**:

System.***out***.println("Enter a valid option");

**break**;

}

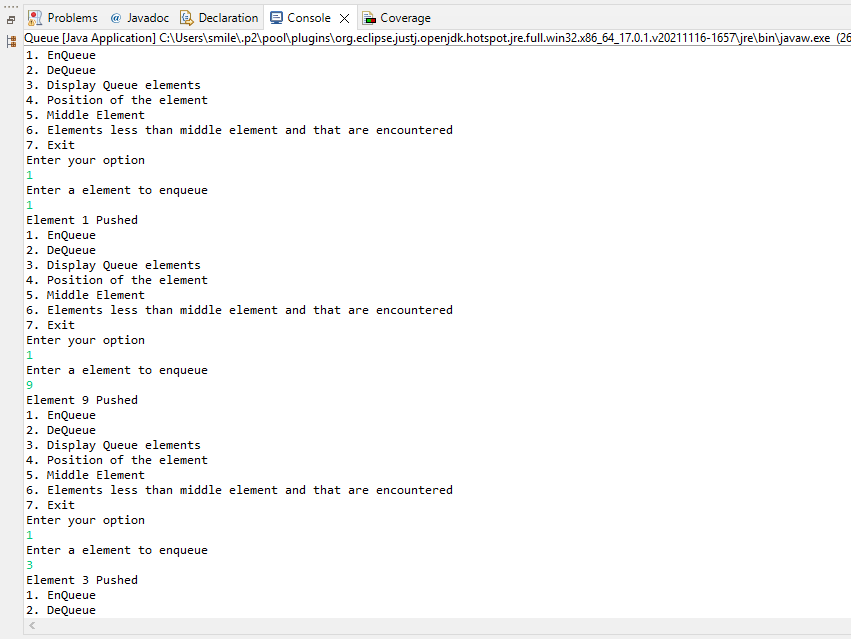
}**while**(cond);

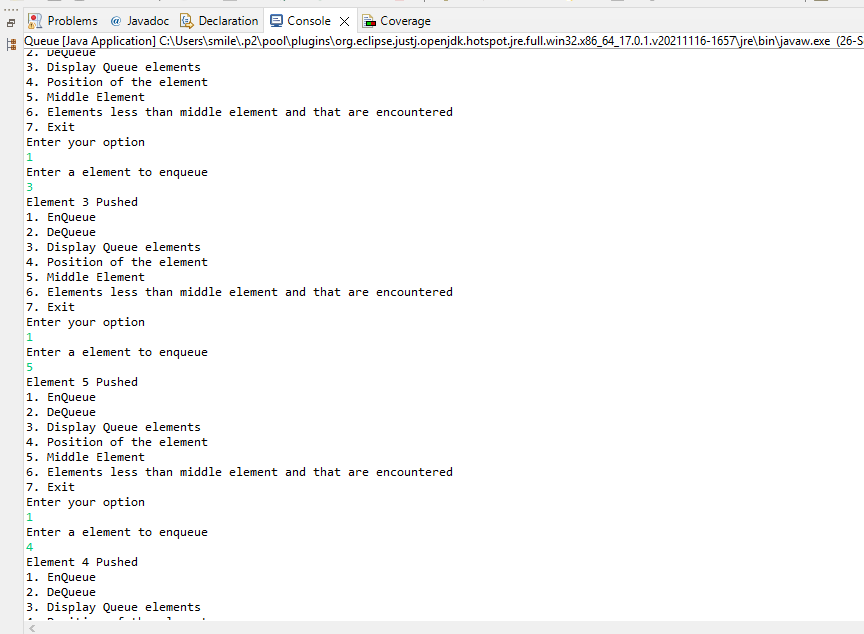
}

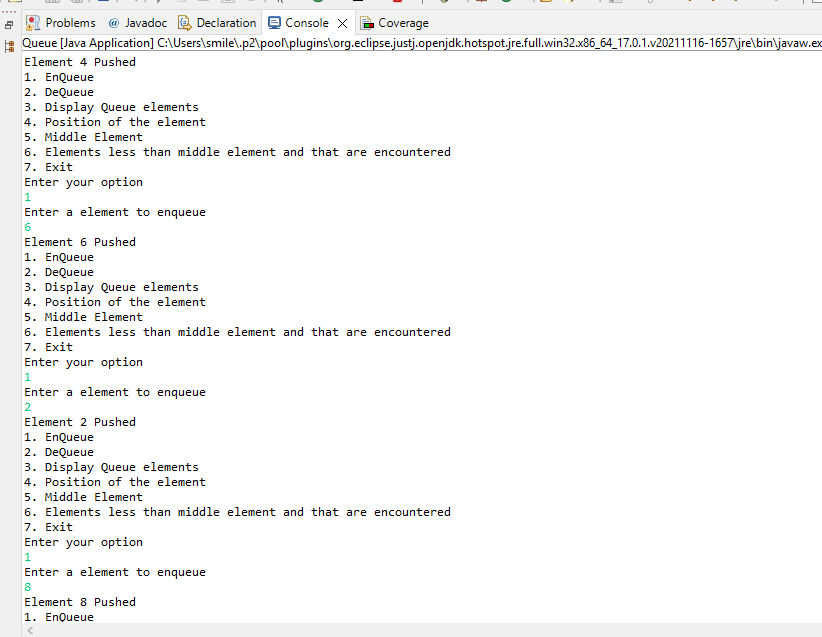
}

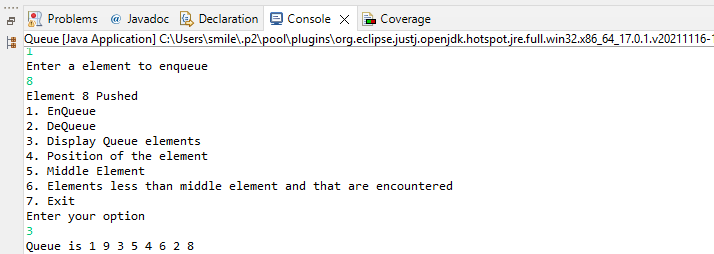
**OUTPUT:**

1.

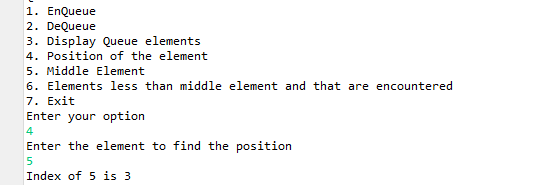




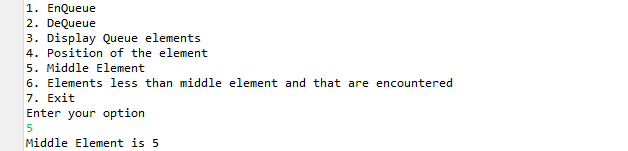




2.



3.



4.

