PLACEMENTS KEY'S

Coding Challenge #3 (Question)

Day - 2

Drishti is working in a Techno company. Drishti was developing an application, where the application will sort the salaries of the workers in increasing order. So, help him in building the application in required time.

Input format:

1. The first line of input contains integer "n" which resembles the size of the array.
2. The second line contains "n" number of array elements.

Output format:

The output contains sorted array in increasing manner.

Sample input 0:

5
900 500 100 200 800

Sample output 0:

100 200 500 800 900

Explanation:

Sort the given array in increasing order. Given array=[900, 500, 100, 200, 800].
The array in increasing manner i.e., array=[100, 200, 500, 800, 900]

Sample input 1:

10
1000 900 800 500 600 200 1100 100 200 1200

Sample output 1:

100 200 200 500 600 800 900 1000 1100 1200

PLACEMENTS KEY'S

Coding Challenge #3 (Question contd.)

Day - 2

Complete the Arraysort() function in the code given below:

```
#include<stdio.h>
int Arraysort(int size,int arr[]);
int Arraysort(int size,int arr[])
{


        /*complete the code*/


}

int main()
{

          int arr[1000],size;
          printf("Enter the size of array: ");
          scanf("%d",&size);
          printf("Enter array elements: ");
          for(int i=0;i<size;i++)
            scanf("%d",&arr[i]);
          Arraysort(size,arr);
          for(int i=0;i<size;i++)
             printf("%d ",arr[i]);
}
```

PLACEMENTS KEY'S

Coding Challenge #3 (C Solution)

Day - 2

```c
#include<stdio.h>
int Arraysort(int size,int arr[]);
int Arraysort(int size,int arr[])
{
      for(int i=0;i<size;i++)     //swapping logic is here
         {
                for(int j=0;j<size-1;j++)
                 {
                        if(arr[j]>arr[j+1])            //if jth element is greater than (j+1)
                        element
                        {                              //then swapping takes place
                           int temp=arr[j];    //temp variable is used to store jth
                        element
                           arr[j]=arr[j+1];
                           arr[j+1]=temp;
                         }
                }
         }

}

int main()
{

        int arr[1000],size;
        printf("Enter the size of array: ");
        scanf("%d",&size);
        printf("Enter array elements: ");
        for(int i=0;i<size;i++)
          scanf("%d",&arr[i]);
        Arraysort(size,arr);
        for(int i=0;i<size;i++)
           printf("%d ",arr[i]);
}
```

PLACEMENTS KEY'S

Coding Challenge #3 (C Solution contd.)

Day - 2

Note:: Earlier, we implemented Sorting using Bubble-Sort technique , as it is most used and easily understood by students. For optimized code and to reduce the time complexity we can prefer the Quick-Sort technique. Quick-sort can be implemented as follows:

```
#include <stdio.h>
int partition(int a[], int beg, int end);
int  quickSort(int a[], int beg, int end);
int main()
{
   int arr[100],size;
   printf("enter the size of the array: ");
   scanf("%d",&size);
   printf("Enter the array elements: ");
   for(int i=0;i<size;i++)
    scanf("%d",&arr[i]);
   quickSort(arr, 0, size-1);
   printf("\n The sorted array is: \n");
   for(int i=0;i<size;i++)
    printf(" %d\t", arr[i]);
}
int partition(int a[], int beg, int end)
{
   int left, right, temp, loc, flag;
   loc = left = beg;
   right = end;
   flag = 0;
   while(flag != 1)
   {
      while((a[loc] <= a[right]) && (loc!=right))
      right--;
      if(loc==right)
      flag =1;
```

PLACEMENTS KEY'S

Coding Challenge #3 (C Solution contd.)

Day - 2

```
else if(a[loc]>a[right])
    {
      temp = a[loc];
      a[loc] = a[right];
      a[right] = temp;
      loc = right;
    }
    if(flag!=1)
    {
      while((a[loc] >= a[left]) && (loc!=left))
      left++;
      if(loc==left)
      flag =1;
      else if(a[loc] < a[left])
      {
        temp = a[loc];
        a[loc] = a[left];
        a[left] = temp;
        loc = left;
      }
    }
  }
  return loc;
}
```

PLACEMENTS KEY'S

Coding Challenge #3 (C Solution contd.)

Day - 2

```
int quickSort(int a[], int beg, int end)
{
   int loc;
   if(beg<end)
   {
      loc = partition(a, beg, end);
      quickSort(a, beg, loc-1);
      quickSort(a, loc+1, end);
   }
}
```

PLACEMENTS KEY'S

Coding Challenge #3 (JAVA Solution)

Day - 2

```
import java.util.*;
public class SalarySort
{
   int partition(int A[], int low, int high)
   {
      int pivot = A[high];
      int pIndex = (low-1);
      for (int j=low; j<high; j++)
      {
         if (A[j] < pivot)
         {
            pIndex++;
            int temp = A[pIndex];
            A[pIndex] = A[j];
            A[j] = temp;
         }
      }
      int temp = A[pIndex+1];
      A[pIndex+1] = A[high];
      A[high] = temp;
      return pIndex+1;
   }
   void salarySort(int A[], int low, int high)
   {
      if (low < high)
      {
         int pIndex = partition(A, low, high);
         salarySort(A, low, pIndex-1);
         salarySort(A, pIndex+1, high);
      }
   }
```

PLACEMENTS KEY'S

Coding Challenge #3 (JAVA Solution contd.)

Day - 2

```
static void printSalaries(int A[])
   {
      int n = A.length;
      for (int i=0; i<n; ++i)
         System.out.print(A[i]+" ");
      System.out.println();
   }
   public static void main(String args[])
   {
      System.out.println("Enter the no. of salaries");
                  Scanner input = new Scanner(System.in);
                  int size = input.nextInt();
                  int salaries[] = new int[size];
                  System.out.println("Enter the salaries");
                  for(int i=0;i<size;i++){
                        salaries[i]=input.nextInt();
                  }
      SalarySort obj = new SalarySort();
      obj.salarySort(salaries, 0, size-1);

      System.out.println("Sorted Salaries");
      printSalaries(salaries);
   }
}
```

PLACEMENTS KEY'S

Coding Challenge #4 (Question)

Day - 2

Jaswanth wants to find the second largest number in the array store. Based on this he is developing the application. So, help him to build the logic which helps him to develop the application.

Input:
The first line contains an integer T, total number of elements. Then follow T integers.

Output:
Display the second largest among the given T integers.

Constraints
1 ≤ T ≤ 1000
1 ≤ integers ≤ 1000000

Sample Input 0:
7
23 45 7 34 25 25 89

Sample Output 0:
45

Sample input 1:
10
10 5 10 19 100 15 16 24 22 24

Sample output 1:
24

PLACEMENTS KEY'S

Coding Challenge #4 (Question contd.)

Day - 2

Complete the Second_largest() function in the code given below:

```
#include<stdio.h>
int Second_largest(int size, int arr[]);
int Second_largest(int size, int arr[])
{

        /*complete the code*/


          return sec_big;                  //returns second largest
}



int main()
{
          int arr[50], size, i, j = 0,sec_big;
          printf("enter the size: \n");    //enter the size of array
           scanf("%d", &size);
          printf("Enter the array elements: ");
          for(int i=0; i<size; i++)       //enter the array elements
            scanf("%d", &arr[i]);
          sec_big=Second_largest(size,arr); //function call is here
          printf("%d", sec_big);
          return 0;
}
```

PLACEMENTS KEY'S

Coding Challenge #4 (C Solution)

Day - 2

```c
#include<stdio.h>
int Second_largest(int size, int arr[]);
int Second_largest(int size, int arr[])
{
        int big=arr[0],sec_big=0,j=0,i;
        for(i=1;i<size;i++)
        {
            if(big<arr[i])
            {
                big=arr[i];                 //logic for second largest
                j = i;
             }
        }
        sec_big =arr[size-j-1];
        for(i=1;i<size;i++)
        {
            if(sec_big <arr[i] && j != i)
                sec_big =arr[i];
        }
        return sec_big;                     //returns second largest
}


int main()
{
        int arr[50], size, i, j = 0,sec_big;
        printf("enter the size: \n");    //enter the size of array
         scanf("%d", &size);
        printf("Enter the array elements: ");
        for(int i=0; i<size; i++)       //enter the array elements
          scanf("%d", &arr[i]);
        sec_big=Second_largest(size,arr); //function call is here
        printf("%d", sec_big);
        return 0;
}
```

PLACEMENTS KEY'S

Coding Challenge #4 (JAVA Solution)

Day - 2

```
import java.util.*;
public class SecondLargest
{
	public static void main(String[] args)
	{
		Scanner input = new Scanner(System.in);
		System.out.println("Enter the size of array");
		int size = input.nextInt();
		int[] array = new int[size];
		System.out.println("Enter the elements");
		for(int i=0;i<size;i++){
			array[i]=input.nextInt();
		}
		System.out.println("Second largest element in the given array is :"
+secondLargest(array,size));
	}
public static int secondLargest(int A[],int size){
		int largest =0;
		int second_largest = 0;
		for(int i=0;i<size;i++){
			if(A[i]>largest){
				int temp = largest;
				largest = A[i];
				second_largest = temp;
			}
			else if(A[i]>second_largest){
				second_largest = A[i];
			}
			else
				continue;
		}
		return second_largest;
	}
}
```