S.No: 1	Exp. Name: ***Design a C program which sorts the strings using array of pointers***	Date:2023-04-24

Aim:

Design a C program that sorts the strings using array of pointers.

Sample input output

```
Sample input-output -1:
Enter the number of strings: 2
Enter string 1: Tantra
Enter string 2: Code
Before Sorting
Tantra
Code
After Sorting
Code
Tantra
Sample input-output -2:
Enter the number of strings: 3
Enter string 1: India
Enter string 2: USA
Enter string 3: Japan
Before Sorting
India
USA
Japan
After Sorting
India
Japan
USA
```

Source Code:

stringssort.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
{
   char * temp;
   int i,j,diff,n;
   char * strarray[10];
   printf("Enter the number of strings: ");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      printf("Enter string %d: ",i+1);
      strarray[i]=(char *)malloc(sizeof(char)*20);
      scanf("%s",strarray[i]);

   }
   printf("Before Sorting\n");
   for(i=0;i<n;i++)
   {
```

```
      printf("%s\n",strarray[i]);
   }
   for(i=0;i<n;i++)
   {
      for(j=0;j<n-1;j++)
      {
         diff=strcmp(strarray[j],strarray[j+1]);
         if(diff>0)
         {
            temp=strarray[j];
            strarray[j]=strarray[j+1];
            strarray[j+1]=temp;
         }
      }

   }
   printf("After Sorting\n");
   for(i=0;i<n;i++)
   {
      printf("%s\n",strarray[i]);
   }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the number of strings: 2
Enter string 1: Tantra
Enter string 2: Code
Before Sorting
Tantra
Code
After Sorting
Code
Tantra

Test Case - 2
User Output
Enter the number of strings: 3
Enter string 1: Dhoni
Enter string 2: Kohli
Enter string 3: Rohit
Before Sorting
Dhoni
Kohli
Rohit
After Sorting
Dhoni
Kohli
Rohit

S.No: 2	Exp. Name: ***Write a C program to Search a Key element using Linear search Technique***	Date:2023-04-24

ID: 224G1A05C6

Aim:

Write a program to search a **key element** with in the given array of elements using `linear search` process.

At the time of execution, the program should print the message on the console as:

```
Enter value of n :
```

For example, if the user gives the **input** as:

```
Enter value of n : 3
```

Next, the program should print the messages one by one on the console as:

```
Enter element for a[0] :
Enter element for a[1] :
Enter element for a[2] :
```

if the user gives the **input** as:

```
Enter element for a[0] : 89
Enter element for a[1] : 33
Enter element for a[2] : 56
```

Next, the program should print the message on the console as:

```
Enter key element :
```

if the user gives the **input** as:

```
Enter key element : 56
```

then the program should **print** the result as:

```
The key element 56 is found at the position 2
```

Similarly if the key element is given as **25** for the above one dimensional array elements then the program should print the output as "**The key element 25 is not found in the array**".

Source Code:

LinearSearch.c

```
#include<stdio.h>
int main()
{
   int a[10],i,j,n,flag=0;
   printf("Enter value of n : ");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      printf("Enter element for a[%d] : ",i);
      scanf("%d",&a[i]);
```

2022-2026-CSE-B

Srinivasa Ramanujan Institute of Technology

```
    }

    printf("Enter key element : ");
    scanf("%d",&j);
    for(i=0;i<n;i++)

    {
       if(j==a[i])
    {
       flag++;
       break;
    }
    }
    if(flag==1)
    {
       printf("The key element %d is found at the position %d",j,i);
    }
       else
       {
          printf("The key element %d is not found in the array",j);
       }
       printf("\n");
    }
```


Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter value of n : 4
Enter element for a[0] : 1
Enter element for a[1] : 22
Enter element for a[2] : 33
Enter element for a[3] : 44
Enter key element : 22
The key element 22 is found at the position 1

Test Case - 2
User Output
Enter value of n : 7
Enter element for a[0] : 101
Enter element for a[1] : 102
Enter element for a[2] : 103
Enter element for a[3] : 104
Enter element for a[4] : 105
Enter element for a[5] : 106
Enter element for a[6] : 107
Enter key element : 110
The key element 110 is not found in the array

S.No: 3	Exp. Name: ***Write a C program to Search a Key element using Binary search Technique***	Date:2023-04-24

Aim:

Write a program to **search** a key element in the given array of elements using `binary search`.

At the time of execution, the program should print the message on the console as:

```
Enter value of n :
```

For example, if the user gives the **input** as:

```
Enter value of n : 3
```

Next, the program should print the messages one by one on the console as:

```
Enter element for a[0] :
Enter element for a[1] :
Enter element for a[2] :
```

if the user gives the **input** as:

```
Enter element for a[0] : 89
Enter element for a[1] : 33
Enter element for a[2] : 56
```

Next, the program should print the message on the console as:

```
Enter key element :
```

if the user gives the **input** as:

```
Enter key element : 56
```

then the program should **print** the result as:

```
After sorting the elements in the array are
Value of a[0] = 33
Value of a[1] = 56
Value of a[2] = 89
The key element 56 is found at the position 1
```

Similarly if the key element is given as **25** for the above one dimensional array elements then the program should print the output as "**The Key element 25 is not found in the array**".

Source Code:

BinarySearch.c

```
#include<stdio.h>
void main()
{
   int a[5],i,j,temp,k,n,flag=0;
   printf("Enter value of n : ");
   scanf("%d",&n);
   for(i=0;i<n;i++)
```

```
    {
       printf("Enter element for a[%d] : ",i);
       scanf("%d",&a[i]);

    }

    for(i=0;i<n-1;i++)
    {
       for(j=i+1;j<n;j++)
       {
          if(a[j]<a[i])
          {
             temp=a[i];
             a[i]=a[j];
             a[j]=temp;

          }
       }
    }
    printf("Enter key element : ");
    scanf("%d",&k);
    printf("After sorting the elements in the array are\n");
    for(i=0;i<n;i++)
    {
       printf("Value of a[%d] = %d\n",i,a[i]);
    }
    for(i=0;i<n;i++)
    {
       if(k==a[i])
       {
          flag++;
          break;

       }
    }
    if(flag==1)
    printf("The key element %d is found at the position %d\n",k,i);
    else
    printf("The Key element %d is not found in the array\n",k);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter value of n : 3
Enter element for a[0] : 25
Enter element for a[1] : 15
Enter element for a[2] : 23
Enter key element : 45
After sorting the elements in the array are
Value of a[0] = 15
Value of a[1] = 23

ID: 224G1A05C6
2022-2026-CSE-B
Srinivasa Ramanujan Institute of Technology

Value of a[2] = 25
The Key element 45 is not found in the array

Test Case - 2
User Output
Enter value of n : 2
Enter element for a[0] : 80
Enter element for a[1] : 39
Enter key element : 50
After sorting the elements in the array are
Value of a[0] = 39
Value of a[1] = 80
The Key element 50 is not found in the array

S.No: 4	Exp. Name: ***Write a C program to implement Fibonacci Search technique***	Date:2023-05-01

ID: 224G1A05C6

Aim:

Write a C program to implement **Fibonacci search** technique

Source Code:

FibonacciSearch.c

```
#include<stdio.h>

void main()
{
   int a[20],i,j,n,flag=0;
   printf("Enter the size of an array: ");
   scanf("%d",  &n);
   printf("Enter the %d array elements\n" ,n);
   for(i=0;i<n;i++)
   {
      scanf("%d",&a[i]);

   }
   printf("Enter the element to be searched: ");
   scanf("%d", &j);
   for(i=0;i<n;i++)
   {
      if(j==a[i])
      {
         flag++;
         break;

      }
   }
   if(flag==1)
   printf("Element found at index: %d.\n", i);
   else
   printf("Element not found.\n");
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the size of an array: 5
Enter the 5 array elements 3 4 5 6 7
Enter the element to be searched: 3
Element found at index: 0.

Test Case - 2
User Output
Enter the size of an array: 5

Enter the 5 array elements 3 4 5 6 7
Enter the element to be searched: 4
Element found at index: 1.

S.No: 5	Exp. Name: ***Write a C program to Sort the elements using Insertion Sort Technique***	Date:2023-05-01

Aim:

Write a program to **sort** the given elements using `insertion sort technique`.

At the time of execution, the program should print the message on the console as:

```
Enter value of n :
```

For example, if the user gives the **input** as:

```
Enter value of n : 3
```

Next, the program should print the messages one by one on the console as:

```
Enter element for a[0] :
Enter element for a[1] :
Enter element for a[2] :
```

if the user gives the **input** as:

```
Enter element for a[0] : 22
Enter element for a[1] : 33
Enter element for a[2] : 12
```

then the program should **print** the result as:

```
Before sorting the elements in the array are
Value of a[0] = 22
Value of a[1] = 33
Value of a[2] = 12
After sorting the elements in the array are
Value of a[0] = 12
Value of a[1] = 22
Value of a[2] = 33
```

Fill in the missing code so that it produces the desired result.

Source Code:

InsertionSortDemo3.c

```
#include<stdio.h>
void main()
{
   int a[20],i,n,j,temp;
   printf("Enter value of n : ");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      printf("Enter element for a[%d] : ",i);
      scanf("%d",&a[i]);
   }
   //Write the for loop to read array elements
   printf("Before sorting the elements in the array are\n");
```

```
    for(i=0;i<n;i++)
    {
       printf("Value of a[%d] = %d",i,a[i]);
       printf("\n");
    }
    //Write the for loop to display aray element before sorting
    for(i=0;i<n;i++)
    {
       for(j=i+1;j<n;j++)
       {if(a[i]>a[j])
       {
          temp=a[i];
          a[i]=a[j];
          a[j]=temp;
       }
    }
}
//Write the code to sort elements

printf("After sorting the elements in the array are\n");
for(i=0;i<n;i++)
{
   printf("Value of a[%d] = %d",i,a[i]);
   printf("\n");
}
//Write the for loop to display array elements after sorting
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter value of n : 6
Enter element for a[0] : 5
Enter element for a[1] : 9
Enter element for a[2] : 2
Enter element for a[3] : 5
Enter element for a[4] : 1
Enter element for a[5] : 3
Before sorting the elements in the array are
Value of a[0] = 5
Value of a[1] = 9
Value of a[2] = 2
Value of a[3] = 5
Value of a[4] = 1
Value of a[5] = 3
After sorting the elements in the array are
Value of a[0] = 1
Value of a[1] = 2
Value of a[2] = 3
Value of a[3] = 5
Value of a[4] = 5

Value of a[5] = 9

Test Case - 2
User Output
Enter value of n : 3
Enter element for a[0] : 5
Enter element for a[1] : 9
Enter element for a[2] : 4
Before sorting the elements in the array are
Value of a[0] = 5
Value of a[1] = 9
Value of a[2] = 4
After sorting the elements in the array are
Value of a[0] = 4
Value of a[1] = 5
Value of a[2] = 9

S.No: 6	Exp. Name: ***Write a C program to Sort the elements using Selection Sort - Smallest element method Technique***	Date:2023-05-07

Aim:

Write a program to **sort** the given array elements using **selection sort smallest element** method.

At the time of execution, the program should print the message on the console as:

```
Enter value of n :
```

For example, if the user gives the **input** as:

```
Enter value of n : 3
```

Next, the program should print the messages one by one on the console as:

```
Enter element for a[0] :
Enter element for a[1] :
Enter element for a[2] :
```

if the user gives the **input** as:

```
Enter element for a[0] : 22
Enter element for a[1] : 33
Enter element for a[2] : 12
```

then the program should **print** the result as:

```
Before sorting the elements in the array are
Value of a[0] = 22
Value of a[1] = 33
Value of a[2] = 12
After sorting the elements in the array are
Value of a[0] = 12
Value of a[1] = 22
Value of a[2] = 33
```

Fill in the missing code so that it produces the desired result.

Source Code:

SelectionSortDemo6.c

```
#include<stdio.h>
void main()
{
   int a[20],i,n,j,small,index;
   printf("Enter value of n : ");
   scanf("%d", &n);
   for(i=0;i<n;i++)
   {
      printf("Enter element for a[%d] : ",i);
      scanf("%d", &a[i]);
      }
      printf("Before sorting the elements in the array are\n");
      for(i=0;i<n;i++)
```

```
      {
         printf("Value of a[%d] = %d",i,a[i]);
         printf("\n");
      }
      for(i=0;i<n;i++)
      {
         for(j=i+1;j<n;j++)
         {
            index=i;
            if(a[j]<a[index])
            {
               index=j;
            }

            small=a[i];
            a[i]=a[index];
            a[index]=small;

         }
      }

      printf("After sorting the elements in the array are\n");
for(i=0;i<n;i++)
{

      printf("Value of a[%d] = %d",i,a[i]);

      printf("\n");

      }
      }
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter value of n : 4
Enter element for a[0] : 78
Enter element for a[1] : 43
Enter element for a[2] : 99
Enter element for a[3] : 27
Before sorting the elements in the array are
Value of a[0] = 78
Value of a[1] = 43
Value of a[2] = 99
Value of a[3] = 27
After sorting the elements in the array are
Value of a[0] = 27
Value of a[1] = 43
Value of a[2] = 78
Value of a[3] = 99

S.No: 7	Exp. Name: ***Write a C program to sort given elements using shell sort technique.***	Date:2023-05-12

Aim:

Write a program to `sort` (`ascending order`) the given elements using `shell sort` technique.

At the time of execution, the program should print the message on the console as:

```
Enter array size :
```

For example, if the user gives the **input** as:

```
Enter array size : 5
```

Next, the program should print the following message on the console as:

```
Enter 5 elements :
```

if the user gives the **input** as:

```
Enter 5 elements : 34 67 12 45 22
```

then the program should **print** the result as:

```
Before sorting the elements are : 34 67 12 45 22
After sorting the elements are : 12 22 34 45 67
```

Note: Do use the **printf()** function with a **newline** character (`\n`).

Source Code:

ShellSort2.c

```
#include <stdio.h>
#include <conio.h>
int main()  {
      int size;
      int *arr, i;
      printf("Enter array size : ");
      scanf("%d",&size);
      arr = (int*) malloc(size *sizeof(int));
      printf("Enter %d elements : ",size);
      for(i=0;i<size;i++)  {
         scanf("%d",&arr[i]);
            }
      printf("Before sorting the elements are : ");
      printArray(arr,size);


      shellSort(arr,size);


      printf("After sorting the elements are : ");
      printArray(arr,size);
      return 0;
   }
```

```
int shellSort(int arr[],int n)
{
   int gap,i,j,temp;
   for(gap=n/2;gap>0;gap/=2)
   {
      for(i=gap;i<n;i++)
      {
         temp = arr[i];
         for(j=i;j>=gap && arr[j-gap]>temp;j-=gap) {
            arr[j]=arr[j-gap];
         }

            arr[j] = temp;
         }
      }
   }

   void printArray(int arr[],int n)
   {
      for(int i=0;i<n;i++)
      {
         printf("%d ",arr[i]);
   }
   printf("\n");

}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter array size : 5
Enter 5 elements : 12 32 43 56 78
Before sorting the elements are : 12 32 43 56 78
After sorting the elements are : 12 32 43 56 78

ID: 224G1A05C6
2022-2026-CSE-B
Srinivasa Ramanujan Institute of Technology

S.No: 8	Exp. Name: ***Write a C program to Sort the elements using Bubble Sort Technique***	Date:2023-05-07

ID: 224G1A05C6

Aim:

Write a program to **sort** the given elements using `bubble sort technique`.

At the time of execution, the program should print the message on the console as:

```
Enter value of n :
```

For example, if the user gives the **input** as:

```
Enter value of n : 3
```

Next, the program should print the messages one by one on the console as:

```
Enter element for a[0] :
Enter element for a[1] :
Enter element for a[2] :
```

if the user gives the **input** as:

```
Enter element for a[0] : 22
Enter element for a[1] : 33
Enter element for a[2] : 12
```

then the program should **print** the result as:

```
Before sorting the elements in the array are
Value of a[0] = 22
Value of a[1] = 33
Value of a[2] = 12
After sorting the elements in the array are
Value of a[0] = 12
Value of a[1] = 22
Value of a[2] = 33
```

Fill in the missing code so that it produces the desired result.

Source Code:

BubbleSortDemo3.c

```
#include<stdio.h>
void main()
{
   int a[20], i, n, j, temp;
   printf("Enter value of n : ");
   scanf("%d", &n);
   for(i=0;i<n;i++)
   {
      printf("Enter element for a[%d] : ",i);
      scanf("%d",&a[i]);

   }
   printf("Before sorting the elements in the array are\n");
   for(i=0;i<n;i++)
```

```
    {
       printf("Value of a[%d] = %d",i,a[i]);
       printf("\n");

    }
    for(i=0;i<n;i++)

    {
       for(j=i+1;j<n;j++)
       {
          if(a[j]<a[i])
          {
             temp=a[i];
             a[i]=a[j];
             a[j]=temp;
          }
       }
    }
    printf("After sorting the elements in the array are\n");
    for(i=0;i<n;i++)
    {
       printf("Value of a[%d] = %d",i,a[i]);
       printf("\n");
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter value of n : 3
Enter element for a[0] : 34
Enter element for a[1] : 25
Enter element for a[2] : 28
Before sorting the elements in the array are
Value of a[0] = 34
Value of a[1] = 25
Value of a[2] = 28
After sorting the elements in the array are
Value of a[0] = 25
Value of a[1] = 28
Value of a[2] = 34

Test Case - 2
User Output
Enter value of n : 5
Enter element for a[0] : 1
Enter element for a[1] : 6
Enter element for a[2] : 3
Enter element for a[3] : 8
Enter element for a[4] : 4

ID: 224G1A05C6
2022-2026-CSE-B
Srinivasa Ramanujan Institute of Technology

Before sorting the elements in the array are
Value of a[0] = 1
Value of a[1] = 6
Value of a[2] = 3
Value of a[3] = 8
Value of a[4] = 4
After sorting the elements in the array are
Value of a[0] = 1
Value of a[1] = 3
Value of a[2] = 4
Value of a[3] = 6
Value of a[4] = 8

S.No: 9	Exp. Name: ***Write a program to sort Ascending order the given elements using quick sort technique.***	Date:2023-05-08

Aim:

Write a program to `sort` (`Ascending order`) the given elements using `quick sort` technique.

Note: Pick the first element as pivot. You will not be awarded marks if you do not follow this instruction.

At the time of execution, the program should print the message on the console as:

```
Enter array size :
```

For example, if the user gives the **input** as:

```
Enter array size : 5
```

Next, the program should print the following message on the console as:

```
Enter 5 elements :
```

if the user gives the **input** as:

```
Enter 5 elements : 34 67 12 45 22
```

then the program should **print** the result as:

```
Before sorting the elements are : 34 67 12 45 22
After sorting the elements are : 12 22 34 45 67
```

Note: Do use the **printf()** function with a **newline** character (`\n`).

Source Code:

QuickSortMain.c

```
#include <stdio.h>


void main()
{
   int arr[15], i, n;
   printf("Enter array size : ");
   scanf("%d", &n);
   printf("Enter %d elements : ",n);
   for(i = 0; i < n; i++)
   {
      scanf("%d",&arr[i]);
   }
   printf("Before sorting the elements are : ");
   display(arr, n);
   quickSort(arr, 0, n - 1);
   printf("After sorting the elements are : ");
   display(arr, n);

}
```

```
void display(int arr[15], int n)
{
int i;
for(i=0;i<n;i++)
{

      printf("%d ",arr[i]);

   }
   printf("\n");
}
int partition(int arr[15], int lb, int ub)
{
   int pivot,down=lb,up=ub,temp;
   pivot=arr[lb];
   while(down<up)
   {
      while(arr[down]<=pivot&&down<up)
      {
         down++;
      }
      while(arr[up]>pivot)
      {
         up--;
      }
      if(down<up)
      {
         temp=arr[up];
         arr[up]=arr[down];
         arr[down]=temp;
      }
   }
   arr[lb]=arr[up];
   arr[up]=pivot;
   return up;

}
void quickSort(int arr[15],int low,int high)
{
   int j;
   if(low<high)
   {
      j=partition(arr,low,high);
      quickSort(arr,low,j-1);
      quickSort(arr,j+1,high);
   }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output

Enter array size : 5
Enter 5 elements : 34 67 12 45 22
Before sorting the elements are : 34 67 12 45 22
After sorting the elements are : 12 22 34 45 67

Test Case - 2
User Output
Enter array size : 8
Enter 8 elements : 77 55 22 44 99 33 11 66
Before sorting the elements are : 77 55 22 44 99 33 11 66
After sorting the elements are : 11 22 33 44 55 66 77 99

Test Case - 3
User Output
Enter array size : 5
Enter 5 elements : -32 -45 -67 -46 -14
Before sorting the elements are : -32 -45 -67 -46 -14
After sorting the elements are : -67 -46 -45 -32 -14

ID: 224G1A05C6

S.No: 10	Exp. Name: ***Write a C program to sort the given elements using Heap sort***	Date:2023-05-08

Aim:

Write a program to sort (ascending order) the given elements using heap sort technique.

Note: Do use the printf() function with a newline character (\n).

Source Code:

HeapSortMain.c

```
#include<stdio.h>
void main()
{
   int arr[15],i,n;
   printf("Enter array size : ");
   scanf("%d",&n);
   printf("Enter %d elements : ",n);
   for(i=0;i<n;i++)
   {
      scanf("%d",&arr[i]);

   }
   printf("Before sorting the elements are : ");
   display(arr, n);
   heapsort(arr, n);
   printf("After sorting the elements are : ");
   display(arr, n);
}

   int display(int arr[15],int n)
   {
      int i;
      for(i=0;i<n;i++)
      {
      printf("%d ",arr[i]);
      }
      printf("\n");
   }
   int heapsort(int arr[15],int n)
   {
      int i;
      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);
      }
   }
   int heapify(int arr[15],int n,int i)
```


ID: 224G1A05C6

Srinivasa Ramanujan Institute of Technology

```
    {
       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 temp=arr[i];
          arr[i]=arr[largest];
          arr[largest]=temp;
          heapify(arr,n,largest);
       }


    }
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter array size : 5
Enter 5 elements : 23 54 22 44 12
Before sorting the elements are : 23 54 22 44 12
After sorting the elements are : 12 22 23 44 54

Test Case - 2
User Output
Enter array size : 6
Enter 6 elements : 12 65 23 98 35 98
Before sorting the elements are : 12 65 23 98 35 98
After sorting the elements are : 12 23 35 65 98 98

Test Case - 3
User Output
Enter array size : 4
Enter 4 elements : -23 -45 -12 -36
Before sorting the elements are : -23 -45 -12 -36
After sorting the elements are : -45 -36 -23 -12

Test Case - 4
User Output
Enter array size : 6
Enter 6 elements : 1 -3 8 -4 -2 5
Before sorting the elements are : 1 -3 8 -4 -2 5
After sorting the elements are : -4 -3 -2 1 5 8

S.No: 11	Exp. Name: ***Write a C program to Sort given elements using Merge sort***	Date:2023-05-15

Aim:

Write a program to `sort` (`Ascending order`) the given elements using `merge sort` technique.

At the time of execution, the program should print the message on the console as:

```
Enter array size :
```

For example, if the user gives the **input** as:

```
Enter array size : 5
```

Next, the program should print the following message on the console as:

```
Enter 5 elements :
```

if the user gives the **input** as:

```
Enter 5 elements : 34 67 12 45 22
```

then the program should **print** the result as:

```
Before sorting the elements are : 34 67 12 45 22
After sorting the elements are : 12 22 34 45 67
```

Note: Do use the **printf()** function with a **newline** character (`\n`).

Source Code:

MergeSortMain.c

```
#include<stdio.h>
void main()
{
   int arr[15],i,n;
   printf("Enter array size : ");
   scanf("%d",&n);
   printf("Enter %d elements : ",n);
   for(i=0;i<n;i++)
   {
      scanf("%d",&arr[i]);

   }
   printf("Before sorting the elements are : ");
   display(arr, n);
   splitAndMerge(arr, 0, n - 1);
   printf("After sorting the elements are : ");
   display(arr, n);

}
void display(int arr[15], int n)
{
   int i;
   for(i=0;i<n;i++)
   printf("%d ",arr[i]);
```

```
   printf("\n");

}
void merge(int arr[15],int low,int mid, int high)
{
   int i=low,h=low,j=mid+1,k,temp[15];
   while(h<=mid&&j<=high)
   {
      if(arr[h]<=arr[j])
      {
         temp[i]=arr[h];
         h++;

      }
      else
      {
         temp[i]=arr[j];
         j++;

      }
      i++;

   }
   if(h>mid)
   {
      for(k=j;k<=high;k++)
      {
         temp[i]=arr[k];
         i++;
      }
   }
   else
   {
      for(k=h;k<=mid;k++)
      {
         temp[i]=arr[k];
         i++;

      }


   }
   for(k=low;k<=high;k++)
   {
      arr[k]=temp[k];

   }


}
void splitAndMerge(int arr[15], int low, int high)
{
   if(low<high)
   {
      int mid=(low+high)/2;
      splitAndMerge(arr,low,mid);
```

```
        splitAndMerge(arr,mid+1,high);
        merge(arr,low,mid,high);

    }
    }
```


Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter array size : 5
Enter 5 elements : 34 67 12 45 22
Before sorting the elements are : 34 67 12 45 22
After sorting the elements are : 12 22 34 45 67

Test Case - 2
User Output
Enter array size : 8
Enter 8 elements : 77 55 22 44 99 33 11 66
Before sorting the elements are : 77 55 22 44 99 33 11 66
After sorting the elements are : 11 22 33 44 55 66 77 99

Test Case - 3
User Output
Enter array size : 5
Enter 5 elements : -32 -45 -67 -46 -14
Before sorting the elements are : -32 -45 -67 -46 -14
After sorting the elements are : -67 -46 -45 -32 -14

S.No: 12	Exp. Name: ***Write a C program to sort given elements using Radix sort***	Date:2023-06-15

Aim:

Write a program to `sort` (`ascending order`) the given elements using `radix sort` technique.

At the time of execution, the program should print the message on the console as:

```
Enter array size :
```

For example, if the user gives the **input** as:

```
Enter array size : 5
```

Next, the program should print the following message on the console as:

```
Enter 5 elements :
```

if the user gives the **input** as:

```
Enter 5 elements : 34 67 12 45 22
```

then the program should **print** the result as:

```
Before sorting the elements are : 34 67 12 45 22
After sorting the elements are : 12 22 34 45 67
```

Note: Do use the **printf()** function with a **newline** character (`\n`).

Source Code:

RadixSortMain2.c

```
#include<stdio.h>
#include<conio.h>

int main() {
   int size;
   int *arr,i;
   printf("Enter array size : ");
   scanf("%d",&size);
   arr = (int*) malloc(size * sizeof(int));
   printf("Enter %d elements : ",size);
   for(i=0; i<size;i++) {

      scanf("%d", &arr[i]);
   }
   printf("Before sorting the elements are : ");
   printArray(arr,size);
   RadixSort(arr,size);
   printf("After sorting the elements are : ");
   printArray(arr,size);
   }
int largest(int a[],int n)
{
   int i,k=a[0];
   for(i=1;i<n;i++) {
```

```
      if(a[i]>k) {
      k = a[i];
      }
   }
   return k;
}
int printArray(int a[],int n) {
   int i;
   for(i=0;i<n;i++) {
      printf("%d ",a[i]);
   }
   printf("\n");
}
int RadixSort(int a[], int n)
{
   int bucket[10][10],bucket_count[10],i,j,k,rem,NOP=0,divi=1,large,pass;
   large=largest(a,n);
   while(large>0)
   {
      NOP++;
      large/=10;
   }
   for(pass=0;pass<NOP;pass++)
   {
      for(i=0;i<10;i++)
      {
      bucket_count[i] = 0;
      }
   for(i=0;i<n;i++)
   {
      rem=(a[i]/divi)%10;
      bucket[rem][bucket_count[rem]]=a[i];
      bucket_count[rem]++;
   }
   i=0;
   for(k=0;k<10;k++)
   {
      for(j=0;j<bucket_count[k];j++)
      {
      a[i] = bucket[k][j];
      i++;
      }
   }
   divi*=10;
   }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter array size : 5
Enter 5 elements : 23
43
54
12
65
Before sorting the elements are : 23 43 54 12 65
After sorting the elements are : 12 23 43 54 65

Test Case - 2
User Output
Enter array size : 7
Enter 7 elements : 23
54
136
85
24
65
76
Before sorting the elements are : 23 54 136 85 24 65 76
After sorting the elements are : 23 24 54 65 76 85 136

ID: 224G1A05C6