**Sorting**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

BUBBLE SORT USING C PROGRAM

**Source code of simple bubble sort implementation using array ascending order in c programming language**

#include<stdio.h>

int main(){

  int s,temp,i,j,a[20];

  printf("Enter total numbers of elements: ");

  scanf("%d",&s);

  printf("Enter %d elements: ",s);

  for(i=0;i<s;i++)

      scanf("%d",&a[i]);

  //Bubble sorting algorithm

  for(i=s-2;i>=0;i--){

      for(j=0;j<=i;j++){

           if(a[j]>a[j+1]){

               temp=a[j];

              a[j]=a[j+1];

              a[j+1]=temp;

           }

      }

  }

  printf("After sorting: ");

  for(i=0;i<s;i++)

      printf(" %d",a[i]);

  return 0;

}

Output:

Enter total numbers of elements: 5

Enter 5 elements: 6 2 0 11 9

After sorting:  0 2 6 9 11

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

C PROGRAM FOR INSERTION SORT

**Source code of simple insertion sort implementation using array in ascending order in c programming language**

#include<stdio.h>

int main(){

  int i,j,s,temp,a[20];

  printf("Enter total elements: ");

  scanf("%d",&s);

  printf("Enter %d elements: ",s);

  for(i=0;i<s;i++)

      scanf("%d",&a[i]);

  for(i=1;i<s;i++){

      temp=a[i];

      j=i-1;

      while((temp<a[j])&&(j>=0)){

      a[j+1]=a[j];

          j=j-1;

      }

      a[j+1]=temp;

  }

  printf("After sorting: ");

  for(i=0;i<s;i++)

      printf(" %d",a[i]);

  return 0;

}

Output:

Enter total elements: 5

Enter 5 elements: 3 7 9 0 2

After sorting:  0 2 3 7 9

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SELECTION SORT USING C PROGRAM

**Source code of simple Selection sort implementation using array ascending order in c programming language**

#include<stdio.h>

int main(){

  int s,i,j,temp,a[20];

  printf("Enter total elements: ");

  scanf("%d",&s);

  printf("Enter %d elements: ",s);

  for(i=0;i<s;i++)

      scanf("%d",&a[i]);

  for(i=0;i<s;i++){

      for(j=i+1;j<s;j++){

           if(a[i]>a[j]){

               temp=a[i];

              a[i]=a[j];

              a[j]=temp;

           }

      }

  }

  printf("After sorting is: ");

  for(i=0;i<s;i++)

      printf(" %d",a[i]);

  return 0;

}

Output:

Enter total elements: 5

Enter 5 elements: 4 5 0 21 7

The array after sorting is:  0 4 5 7 21

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

QUICK SORT USING C PROGRAM

**Source code of simple quick sort implementation using array ascending order in c programming language**

#include<stdio.h>

void quicksort(int [10],int,int);

int main(){

  int x[20],size,i;

  printf("Enter size of the array: ");

  scanf("%d",&size);

  printf("Enter %d elements: ",size);

  for(i=0;i<size;i++)

    scanf("%d",&x[i]);

  quicksort(x,0,size-1);

  printf("Sorted elements: ");

  for(i=0;i<size;i++)

    printf(" %d",x[i]);

  return 0;

}

void quicksort(int x[10],int first,int last){

    int pivot,j,temp,i;

     if(first<last){

         pivot=first;

         i=first;

         j=last;

         while(i<j){

             while(x[i]<=x[pivot]&&i<last)

                 i++;

             while(x[j]>x[pivot])

                 j--;

             if(i<j){

                 temp=x[i];

                  x[i]=x[j];

                  x[j]=temp;

             }

         }

         temp=x[pivot];

         x[pivot]=x[j];

         x[j]=temp;

         quicksort(x,first,j-1);

         quicksort(x,j+1,last);

    }

}

Output:

Enter size of the array: 5

Enter 5 elements: 3 8 0 1 2

Sorted elements: 0 1 2 3 8

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Merge sort program in c

**Source code of simple merge sort implementation using array in ascending order in c programming language**

#include<stdio.h>

#define MAX 50

void mergeSort(int arr[],int low,int mid,int high);

void partition(int arr[],int low,int high);

int main(){

    int merge[MAX],i,n;

    printf("Enter the total number of elements: ");

    scanf("%d",&n);

    printf("Enter the elements which to be sort: ");

    for(i=0;i<n;i++){

         scanf("%d",&merge[i]);

    }

    partition(merge,0,n-1);

    printf("After merge sorting elements are: ");

    for(i=0;i<n;i++){

         printf("%d ",merge[i]);

    }

   return 0;

}

void partition(int arr[],int low,int high){

    int mid;

    if(low<high){

         mid=(low+high)/2;

         partition(arr,low,mid);

         partition(arr,mid+1,high);

         mergeSort(arr,low,mid,high);

    }

}

void mergeSort(int arr[],int low,int mid,int high){

    int i,m,k,l,temp[MAX];

    l=low;

    i=low;

    m=mid+1;

    while((l<=mid)&&(m<=high)){

         if(arr[l]<=arr[m]){

             temp[i]=arr[l];

             l++;

         }

         else{

             temp[i]=arr[m];

             m++;

         }

         i++;

    }

    if(l>mid){

         for(k=m;k<=high;k++){

             temp[i]=arr[k];

             i++;

         }

    }

    else{

         for(k=l;k<=mid;k++){

             temp[i]=arr[k];

             i++;

         }

    }

    for(k=low;k<=high;k++){

         arr[k]=temp[k];

    }

}

**Sample output:**

Enter the total number of elements: 5

Enter the elements which to be sort: 2 5 0 9 1

After merge sorting elements are: 0 1 2 5 9

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/