**OBJECTIVE:**-Program to sort the array using various technique.

#include<stdio.h>

void swap(int \*a,int \*b){

int t=\*a;

\*a=\*b;

\*b=t;

}

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

int pivot=arr[high];

int i=(low-1),j;

for(j=low;j<=high-1;j++){

if(arr[j]<pivot){

i++;

swap(&arr[i],&arr[j]);

}

}

swap(&arr[i+1],&arr[high]);

return (i+1);

}

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

if(low<high){

int pi=partition(arr,low,high);

quickSort(arr,low,pi-1);

quickSort(arr,pi+1,high);

}

}

void insertionSort(int arr[],int n){

int i,key,j;

for(i=1;i<n;i++){

key=arr[i];

j=i-1;

while(j>=0 && arr[j]>key){

arr[j+1]=arr[j];

j=j-1;

}

arr[j+1]=key;

}

}

void merge(int arr[],int l,int m,int r){

int i,j,k;

int n1=m-l+1;

int n2=r-m;

int L[n1],R[n2];

for(i=0;i<n1;i++)

L[i]=arr[l+i];

for(j=0;j<n2;j++)

R[j]=arr[m+1+j];

i=0;

j=0;

k=l;

while(i<n1 && j<n2){

if(L[i]<=R[j]){

arr[k]=L[i];

i++;

}

else{

arr[k]=R[j];

j++;

}

k++;

}

while(i<n1){

arr[k]=L[i];

i++;

k++;

}

while(j < n2){

arr[k]=R[j];

j++;

k++;

}

}

void mergeSort(int arr[], int l, int r){

if (l<r){

int m=l+(r-l)/2;

mergeSort(arr,l,m);

mergeSort(arr,m+1,r);

merge(arr,l,m,r);

}

}

void printArray(int arr[],int size){

int i;

for(i=0;i<size;i++) printf("%d ",arr[i]);

}

int main(){

int n,i,arr[20],choice;

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

scanf("%d",&n);

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

for(i=0;i<n;i++) scanf("%d",&arr[i]);

printf("\n1. Quick Sort \n2. Insertion Sort \n3. Merge Sort\n");

printf("\nEnter your choice : ");

scanf("%d", &choice);

switch(choice){

case 1:

quickSort(arr,0,n-1);

printf("Sorted array: \n");

printArray(arr,n);

break;

case 2:

insertionSort(arr,n);

printf("Sorted array: \n");

printArray(arr,n);

break;

case 3:

mergeSort(arr,0,n-1);

printf("Sorted array: \n");

printArray(arr,n);

break;

}

return 0;

}

**OUTPUT:-**

Enter the size of array: 5

Enter the elements of array: 12 22 44 55 66

1. Quick Sort

2. Insertion Sort

3. Merge Sort

Enter your choice : 1

Sorted array:

12 22 44 55 66