```
#include <iostream>
using namespace std;

int bubble_sort()
{
   int size;
      cout <<"***** BUBBLE SORT *****"<<endl<<endl;
      cout <<"Enter The Size : ";
      cin >>size;
      int arr[size];
      cout <<"Enter The Element's To Be Sorted : "<<endl;
      for(int index = 0; index < size; index++)
      {
         cin >>arr[index];
      }

      for(int index1 = 0; index1 < size - 1; index1++)
      {
         for(int index2 = 0; index2 < size - 1; index2++)
         {
            if(arr[index2] > arr[index2 + 1])
            {
               int temp;
               temp = arr[index2];
```

```
                arr[index2] = arr[index2 + 1];
                arr[index2 + 1] = temp;
            }
        }
    }
    cout <<"Sorted Element's --> ";
    for(int index = 0; index < size; index++)
    {
        cout <<arr[index]<<" ";
    }
}

int insertion_sort()
{
   int size;
   cout <<"***** INSERTION SORT *****"<<endl<<endl;
   cout <<"Enter The Size : ";
   cin >>size;
   int arr[size];
   cout <<"Enter The Element's To Be Sorted : "<<endl;
   for(int index = 0; index < size; index++)
   {
      cin >>arr[index];
   }
```

```
    //Insertion Process
    for(int index = 1; index < size; index++)
    {
        int temp, prev;
        temp = arr[index];
        prev = index - 1;
        while(prev >= 0 && arr[prev] > temp)
        {
            arr[prev + 1] = arr[prev];
            prev--;
        }
        arr[prev + 1] = temp;
    }
    cout <<"Sorted Element's --> ";
    for(int index = 0; index < size; index++)
    {
        cout <<arr[index]<<" ";
    }
}

int selection_sort()
{
    int size;
    cout <<"***** SELECTION SORT *****"<<endl<<endl;
```

```
cout <<"Enter The Size : ";
cin >>size;
int arr[size];
cout <<"Enter The Element's To Be Sorted : "<<endl;
for(int index = 0; index < size; index++)
{
   cin >>arr[index];
}
//Selection Process
for(int index = 0; index < size - 1; index++)
{
   int min = index;
   for(int next_index = index + 1; next_index < size; next_index++)
   {
      if(arr[next_index] < arr[min])
      {
         min = next_index;
      }
   }
   if(min != index)
   {
      swap(arr[index], arr[min]);
   }
}
```

```
    cout <<"Sorted Element's --> ";
    for(int index = 0; index < size; index++)
    {
        cout <<arr[index]<<" ";
    }
}

int partition(int arr[], int start, int end)
{
    int pivot = arr[end];
    int pIndex = start;
    for(int index = start; index < end; index++)
    {
        if(arr[index] <= pivot)
        {
            swap(arr[index], arr[pIndex]);
            pIndex++;
        }
    }
    swap(arr[end], arr[pIndex]);
    return pIndex;
}
void quickSort(int arr[], int start, int end)
{
```

```
    if(start < end)
    {
        int mid = partition(arr, start, end);
        quickSort(arr, start, (mid - 1));
        quickSort(arr, (mid + 1), end);
    }
}

int quick_sort()
{
    int size;
    cout <<"***** QUICK SORT *****"<<endl<<endl;
    cout <<"Enter The Size : ";
    cin >>size;
    int arr[size];
    cout <<"Enter The Element's To Be Sorted : "<<endl;
    for(int index = 0; index < size; index++)
    {
        cin >>arr[index];
    }
    quickSort(arr, 0, (size - 1));
    cout <<"Sorted Element's --> ";
    for(int index = 0; index < size; index++)
    {
```

```
        cout <<arr[index]<<" ";
    }
}

void merge(int *, int, int, int);
void mergeSort(int *arr, int low, int high)
{
    int mid;
    if(low < high)
    //Divide the array at mid
    {
        mid = (low + high) / 2;
        mergeSort(arr, low, mid);
        mergeSort(arr, mid+1, high);
        //Merge or conquer sorted arrays
        merge(arr, low, high, mid);
    }
}
//Merge sort
void merge(int *arr, int low, int high, int mid)
{
    int low1, mid1, low2, arr1[50];
    low1 = low;
    mid1 = mid + 1;
```

```
low2 = low;
while(low1 <= mid && mid1 <= high)
{
   if(arr[low1] < arr[mid1])
   {
      arr1[low2] = arr[low1];
      low2++;
      low1++;
   }
   else
   {
      arr1[low2] = arr[mid1];
      low2++;
      mid1++;
   }
}
while(low1 <= mid)
{
   arr1[low2] = arr[low1];
   low2++;
   low1++;
}
while(mid1 <= high)
{
```

```
        arr1[low2] = arr[mid1];
        low2++;
        mid1++;
    }
    for(low1 = low; low1 < low2; low1++)
    {
        arr[low1] = arr1[low1];
    }
}

int merge_sort()
{
    int size;
    cout <<"***** MERGE SORT *****"<<endl<<endl;
    cout <<"Enter The Size : ";
    cin >>size;
    int arr[size];
    cout <<"Enter The Element's To Be Sorted : "<<endl;
    for(int index = 0; index < size; index++)
    {
        cin >>arr[index];
    }
    mergeSort(arr, 0, size - 1);
    cout <<"Sorted Element's --> ";
```

```
    for(int index = 0; index < size; index++)
    {
        cout <<arr[index]<<" ";
    }
}
int main()
{
    int n;
    cout <<"PROGRAM TO SORT THE ELEMENTS"<<endl<<endl;
    cout <<"1. Selection Sort "<<endl;
    cout <<"2. Insertion Sort "<<endl;
    cout <<"3. Merge Sort "<<endl;
    cout <<"4. Quick Sort "<<endl;
    cout <<"5. Bubble Sort "<<endl<<endl;
    cout <<"Enter The Choice (1-5): ";
    cin >> n;
    cout <<endl;

    switch(n)
    {
    case 1:
        selection_sort();
        break;
```

```
    case 2:
        insertion_sort();
        break;

    case 3:
        merge_sort();
        break;

    case 4:
        quick_sort();
        break;

    case 5:
        bubble_sort();
        break;
    }
}
```