Practical No: 5. Write a program for sorting given array in ascending/descending order using

A. <u>Heap sort</u>

//Heap Sort (Ascending order)

```
#include <iostream>
#include <conio.h>
#include <stdlib.h>
using namespace std;
int a[1100], n;
class Heap_Sort {
public:
void getdata();
void show();
void heap_sort(int[], int);
void heapify(int[], int);
void adjust(int[], int, int);
} m;
void Heap_Sort::getdata() {
cout<< "\nEnter the number of elements: ";
cin>> n;
cout<< "Enter " << n << " elements:\n";
for (int i = 1; i<= n; i++) {
cout<< "Element " <<i<< ": ";
cin>> a[i];
    }
}
void Heap_Sort::show() {
for (int i = 1; i<= n; i++)
cout<< a[i] << " ";
cout<<endl;
}
void Heap_Sort::heap_sort(int a[], int n) {
heapify(a, n);
for (int i = n; i>= 2; i--) {
int t = a[i];
a[i] = a[1];
a[1] = t;
adjust(a, 1, i - 1);
    }
}
```

```
void Heap_Sort::heapify(int a[], int n) {
for (int i = n / 2; i>= 1; i--)
adjust(a, i, n);
}
void Heap_Sort::adjust(int a[], int i, int n) {
int item = a[i];
int j = 2 * i;
while (j <= n) {
if (j < n && a[j] < a[j + 1])
j++;
if (item >= a[j])
break;
a[j / 2] = a[j];
        j = 2 * j;
    }
a[j / 2] = item;
}
void main() {
//clrscr();
m.getdata();
cout<< "\nOriginal array: ";
m.show();
m.heap_sort(a, n);
cout<< "\nSorted array in Ascending Order: ";
m.show();
getch();
}
```

/* OUTPUT*/

```
Enter the number of elements: 5
Enter 5 elements:
Element 1: 90
Element 2: 30
Element 3: 60
Element 4: 10
Element 5: 20

Original array: 90 30 60 10 20

Sorted array in Ascending Order: 10 20 30 60 90
```

//Heap Sort (Descending Order)

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
using namespace std;
int a[100], n;
class Heap_Sort {
public:
void getdata();
void show();
void heap_sort(int[], int);
void heapify(int[], int);
void adjust(int[], int, int);
} m;
void Heap_Sort::getdata() {
cout<< "\nEnter the number of elements: ";
cin>> n;
cout<< "Enter " << n << " elements:\n";
for(int i = 1; i<= n; i++) {
cout<< "Element " <<i<< ": ";
cin>> a[i];
   }
}
void Heap_Sort::show() {
for(int i = 1; i<= n; i++)
cout<< a[i] << " ";
cout<<endl;
}
void Heap_Sort::heap_sort(int a[], int n) {
heapify(a, n);
for(int i = n; i>= 2; i--) {
int t = a[i];
a[i] = a[1];
a[1] = t;
adjust(a, 1, i - 1);
   }
}
void Heap_Sort::heapify(int a[], int n) {
for(int i = n / 2; i>= 1; i--)
adjust(a, i, n);
}
void Heap_Sort::adjust(int a[], int i, int n) {
```

```
int item = a[i];
int j = 2 * i;
while(j <= n) {
if(j < n && a[j] > a[j + 1])
j++;
if(item <= a[j])
break;
else {
a[j / 2] = a[j];
          j = 2 * j;
       }
   }
a[j / 2] = item;
}
void main() {
//clrscr();
m.getdata();
cout<< "\nOriginal array: ";
m.show();
m.heap_sort(a, n);
cout<< "\nSorted array (Descending Order): ";
m.show();
getch();
}
```

/*OUTPUT*/

```
Enter the number of elements: 5
Enter 5 elements:
Element 1: 4
Element 2: 9
Element 3: 5
Element 4: 2
Element 5: 1

Original array: 4 9 5 2 1

Sorted array (Descending Order): 9 5 4 2 1
```

B. Merge Sort

//Merge Sort (Ascending Order)

```
#include <iostream>
#include <conio.h>
#include <stdlib.h>
using namespace std;
int a[1100];
int n;
class Merge_Sort {
public:
void getdata();
void show();
void sort();
void merge_sort(int low, int high);
void merge(int low, int mid, int high);
};
void Merge_Sort::getdata() {
cout<< "Enter the number of elements: ";
cin>> n;
cout<< "Enter " << n << " elements:\n";
for (int i = 0; i< n; ++i)
cin>> a[i];
}
void Merge_Sort::show() {
for (inti = 0; i< n; ++i)
cout<< a[i] << " ";
cout<<endl;
}
void Merge_Sort::merge_sort(int low, int high) {
if (low < high) {
int mid = (low + high) / 2;
merge_sort(low, mid);
merge_sort(mid + 1, high);
merge(low, mid, high);
    }
}
void Merge_Sort::merge(int low, int mid, int high) {
int i = low, j = mid + 1, k = 0;
int temp[1100];

while (i<= mid && j <= high) {
```

```
if (a[i] <= a[j]) {
temp[k++] = a[i++];
        } else {
temp[k++] = a[j++];
        }
    }
while (i<= mid)
temp[k++] = a[i++];
while (j <= high)
temp[k++] = a[j++];
for (int m = 0; m < k; ++m)
a[low + m] = temp[m];
}
void Merge_Sort::sort() {
if (n > 0)
merge_sort(0, n - 1);
}
int main() {
//clrscr();
Merge_Sort m;
m.getdata();
cout<< "Original array: ";
m.show();
m.sort();
cout<< "Sorted array in ascending order: ";
m.show();
getch();
return 0;
    //getch();
}
```

/*OUTPUT*/

```
Enter the number of elements: 6
Enter 6 elements:
90
30
80
10
20
50
Original array: 90 30 80 10 20 5
Sorted array in ascending order: 10 20 30 50 80 90
```

//Merge Sort (Descending Order)

```
#include <iostream>
#include <conio.h>
using namespace std;
int a[1100];
int n;
class Merge_Sort {
public:
void getdata();
void show();
void sort();
void merge_sort(int low, int high);
void merge(int low, int mid, int high);
};
void Merge_Sort::getdata() {
cout<< "Enter the number of elements: ";
cin>> n;
cout<< "Enter " << n << " elements:\n";
for (int i = 0; i< n; ++i)
        cin>> a[i];
}
void Merge_Sort::show() {
for (int i = 0; i< n; ++i)
        cout<< a[i] << " ";
cout<<endl;
}
void Merge_Sort::merge_sort(int low, int high) {
if (low < high) {
        int mid = (low + high) / 2;
        merge_sort(low, mid);
        merge_sort(mid + 1, high);
        merge(low, mid, high);
   }
}
void Merge_Sort::merge(int low, int mid, int high) {
int i = low, j = mid + 1, k = 0;
int temp[1100];
while (i<= mid && j <= high) {
        if (a[i] >= a[j]) {
        temp[k++] = a[i++];
        } else {
        temp[k++] = a[j++];
```

```
        }
    }
while (i<= mid)
        temp[k++] = a[i++];
while (j <= high)
        temp[k++] = a[j++];
for (int m = 0; m < k; ++m)
        a[low + m] = temp[m];
}

void Merge_Sort::sort() {
if (n > 0)
        merge_sort(0, n - 1);
}
int main() {
//clrscr();
Merge_Sort m;
m.getdata();
cout<< "Original array: ";
m.show();
m.sort();
cout<< "Sorted array in descending order: ";
m.show();
getch();
return 0;
}
```

/*OUTPUT*/

```
Enter the number of elements: 5
Enter 5 elements:
89
32
19
55
99
Original array: 89 32 19 55 99
Sorted array in descending order: 99 89 55 32 19
```

C. Quick Sort

//Quick Sort(Ascending Order)

```
#include <iostream>
#include <conio.h>
using namespace std;
int a[1100];
int n;
class Quick_Sort {
public:
void getdata();
void show();
void quick_sort(int low, int high);
int partition(int low, int high);
};
void Quick_Sort::getdata() {
cout<< "Enter the number of elements: ";
cin>> n;
cout<< "Enter " << n << " elements:\n";
for (inti = 0; i< n; ++i)
        cin>> a[i];
}
void Quick_Sort::show() {
for (inti = 0; i< n; ++i)
        cout<< a[i] << " ";
cout<<endl;
}
int Quick_Sort::partition(int low, int high) {
int pivot = a[low];
int i = low + 1;
int j = high;
while (1) {
        while (i<= high && a[i] <= pivot)
        i++;
        while (a[j] > pivot)
        j--;
        if (i< j) {
        int temp = a[i];
        a[i] = a[j];
        a[j] = temp;
        } else {
        break;
```

```
        }
    }
int temp = a[low];
a[low] = a[j];
a[j] = temp;
return j;
}
void Quick_Sort::quick_sort(int low, int high) {
if (low < high) {
        int p = partition(low, high);
        quick_sort(low, p - 1);
        quick_sort(p + 1, high);
    }
}
int main() {
clrscr();
Quick_Sort q;
q.getdata();
cout<< "Original array: ";
q.show();
q.quick_sort(0, n - 1);
cout<< "Sorted array in ascending order: ";
q.show();
getch();
return 0;
}
```

/*OUTPUT*/

```
Enter the number of elements: 5
Enter 5 elements:
97
45
66
32
54
Original array: 97 45 66 32 54
Sorted array in ascending order: 32 45 54 66 97
```

//Quick Sort(Descending Order)

```
#include <iostream>
#include <conio.h>
using namespace std;
```

```
int a[1100];
int n;
class Quick_Sort {
public:
void getdata();
void show();
void quick_sort(int low, int high);
int partition(int low, int high);
};
void Quick_Sort::getdata() {
cout<< "Enter the number of elements: ";
cin>> n;
cout<< "Enter " << n << " elements:\n";
for (inti = 0; i< n; ++i)
cin>> a[i];
}
void Quick_Sort::show() {
for (inti = 0; i< n; ++i)
cout<< a[i] << " ";
cout<<endl;
}
int Quick_Sort::partition(int low, int high) {
int pivot = a[low];
inti = low + 1;
int j = high;
while (1) {
while (i<= high && a[i] >= pivot)
i++;
while (a[j] < pivot)
j--;
if (i< j) {
int temp = a[i];
a[i] = a[j];
a[j] = temp;
        } else {
break;
        }
    }
int temp = a[low];
a[low] = a[j];
a[j] = temp;
return j;
}
```

```
void Quick_Sort::quick_sort(int low, int high) {
if (low < high) {
int p = partition(low, high);
quick_sort(low, p - 1);
quick_sort(p + 1, high);
    }
}
int main() {
clrscr();
Quick_Sort q;
q.getdata();
cout<< "Original array: ";
q.show();
q.quick_sort(0, n - 1);
cout<< "Sorted array in descending order: ";
q.show();
getch();
return 0;
}
```

/*OUTPUT*/

```
Enter the number of elements: 5
Enter 5 elements:
33
60
49
32
11
Original array: 33 60 49 32 11
Sorted array in descending order: 60 49 33 32 11
```