OUTPUT

1.

```
av@vasundhara:~/Desktop/DAA$ g++ Red_black_tree.cpp
av@vasundhara:~/Desktop/DAA$ ./a.out
sh: 1: cls: not found

Enter your choice.
1.Insertion.
2.Deletion.
3.Search a number.
4.Exit.
1

Enter the number to be inserted in tree.
23
sh: 1: cls: not found

Enter your choice.
1.Insertion.
2.Deletion.
3.Search a number.
4.Exit.
1

Enter the number to be inserted in tree.
34
sh: 1: cls: not found

Enter your choice.
1.Insertion.
2.Deletion.
3.Search a number.
4.Exit.
1
```

```
Enter the number to be inserted in tree.
45
sh: 1: cls: not found
0
Enter your choice.
1.Insertion.
2.Deletion.
3.Search a number.
4.Exit.
1

Enter the number to be inserted in tree.
24
sh: 1: cls: not found

Enter your choice.
1.Insertion.
2.Deletion.
3.Search a number.
4.Exit.
1

Enter the number to be inserted in tree.
27
sh: 1: cls: not found
0
Enter your choice.
1.Insertion.
2.Deletion.
3.Search a number.
4.Exit.
1
```

```
Enter the number to be inserted in tree.
39
sh: 1: cls: not found

Enter your choice.
1.Insertion.
2.Deletion.
3.Search a number.
4.Exit.
1

Enter the number to be inserted in tree.
50
sh: 1: cls: not found

Enter your choice.
1.Insertion.
2.Deletion.
3.Search a number.
4.Exit.
3

Enter number to be searched.
24


24
sh: 1: pause: not found
sh: 1: cls: not found
color :black
Enter your choice.
1.Insertion.
2.Deletion.
3.Search a number.
4.Exit.
2
```

```
Enter number to be searched.
24


24
sh: 1: pause: not found
sh: 1: cls: not found
color :black
Enter your choice.
1.Insertion.
2.Deletion.
3.Search a number.
4.Exit.
2

Enter number to be deleted.
50
sh: 1: pause: not found
sh: 1: cls: not found

Enter your choice.
1.Insertion.
2.Deletion.
3.Search a number.
4.Exit.
4
av@vasundhara:~/Desktop/DAA$
```

2.

```
av@vasundhara:~/Desktop/DAA$ g++ minimum_spanning_tree
av@vasundhara:~/Desktop/DAA$ ./a.out

***************KRUSKAL'S ALGORITHM****************
Enter the no. of vertices in the graph:6

Enter the weights of the following:
edge 1 , 2 :1
edge 1 , 3 :0
edge 1 , 4 :0
edge 1 , 5 :1
edge 1 , 6 :1
edge 2 , 3 :1
edge 2 , 4 :0
edge 2 , 5 :1
edge 2 , 6 :0
edge 3 , 4 :1
edge 3 , 5 :0
edge 3 , 6 :0
edge 4 , 5 :1
edge 4 , 6 :0
edge 5 , 6 :1


The edges in the given graph are::
 < 1 , 2 > 1
 < 1 , 5 > 1
 < 1 , 6 > 1
 < 2 , 3 > 1
 < 2 , 5 > 1
 < 3 , 4 > 1
 < 4 , 5 > 1
 < 5 , 6 > 1


After sorting the edges in the given graph are::
1 , 2 > ::1
```

```
After sorting the edges in the given graph are::
1 , 2 > ::1
1 , 5 > ::1
1 , 6 > ::1
2 , 3 > ::1
2 , 5 > ::1
3 , 4 > ::1
4 , 5 > ::1
5 , 6 > ::1

********* THE MINIMUM SPANNING TREE IS**************The edge included in MST is :: < 1 , 2 >
The edge included in MST is :: < 1 , 5 >
The edge included in MST is :: < 1 , 6 >
The edge included in MST is :: < 2 , 3 >
Edge  < 2 , 5 > is not included as it forms a cycle

The edge included in MST is :: < 3 , 4 >
Edge  < 4 , 5 > is not included as it forms a cycle

Edge  < 5 , 6 > is not included as it forms a cycle

av@vasundhara:~/Desktop/DAA$
```

3.

```
insertion sort:--
Comparisons : 9
1 2 4 5 6 10
----------------------------
merge sort :--
comparsion  :23
Array after Sorting
3
5
7
9
10
11
12
13
16
24
----------------------------
bubble sort :--
comparision:-17
Sorted array:
64 25 34 12 22 11 90
----------------------------
selection sort :--
no.of comparsion:-9
Sorted array:
32671 64 25 12 11
----------------------------
Quick sort :--
no. of comparsion:--1
sorted array1
5
7
8
9
10
```