

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 void insertq(int q[],int node, int *f, int *r)
5 {
6     if((*f===-1) && (*r===-1))
7     {
8         (*f)++, (*r)++, q[*f]=node;
9     }
10    else{
11        (*r)++, q[*r]=node;
12        }
13    }
14
15 int deleteq(int q[],int *f,int *r)
16 {
17     int temp;
18     temp=q[*f];
19     if(*f == *r) *f=*r=-1;
20     else (*f)++;
21     return temp;
22 }
23
24 void bfs(int n, int adj[][10],int src, int visited[])
25 {
26     int q[20], f=-1,r=-1,v,i;
27     insertq(q,src,&f,&r);
28     while((f <=r ) && (f != -1))
29     {
30         v=deleteq(q,&f,&r);
31         if(visited[v] !=1)
32         {
33             visited[v]=1;
34             printf("%d",v);
```

```
main.c

34         printf("%d",v);
35     }
36     for(i=1;i<=n;i++)
37     {
38         if((adj[v][i]==1) && (visited[i] !=1))
39             insertq(q,i,&f,&r);
40     }
41
42
43     int main()
44     {
45         int n,i,j,adj[10][10],src,visited[10];
46
47         printf("enter number of vertices\n");
48         scanf("%d",&n);
49         printf("Enter adjacency matrix\n");
50         for(i=1;i<=n;i++)
51         {
52             visited[i]=0;
53             for(j=1;j<=n;j++)
54                 scanf("%d",&adj[i][j]);
55         }
56         printf("enter starting vertex\n");
57         scanf("%d",&src);
58         printf("The nodes reachable from src are\n");
59         bfs(n,adj,src,visited);
60     }
```

```
+ clang-7 -pthread -lm -o main main.c
+ ./main
enter number of vertices
4
Enter adjacency matrix
0 1 0 1
0 0 0 1
1 1 0 1
0 0 0 0
enter starting vertex
1
The nodes reachable from src are
+ [ ]
```

main.c

```
1 #include<stdio.h>
2 #include<time.h>
3 void sort(int a[],int n);
4 int main()
5 {
6     clock_t t;
7     int n;
8     printf("\nEnter the Number Of Elements Of The
9     Array\n");
10    scanf("%d",&n);
11    int a[n];
12    printf("Enter the elements of the array\n");
13    for(int i=0;i<n;i++)
14    {
15        scanf("%d",&a[i]);
16    }
17    t=clock();
18    sort(a,n);
19    t=clock()-t;
20    double time_taken=((double)t)/CLOCKS_PER_SEC;
21    printf("Time Taken =%f\n",time_taken);
22
23    printf("Final Sorted Order Is\n");
24    for(int i=0;i<n;i++)
25    {
26        printf("%d\t",a[i]);
27    }
28
29 void sort(int a[] ,int n)
30 {
31     int v,j;
32     for(int i=1;i<=n-1;i++)
33     {
34         v=a[i];
35         j=i-1;
```



main.c

```
30
31 {
32     v=a[i];
33     j=i-1;
34     while(j>=0 && a[j]>v)
35     {
36         a[j+1]=a[j];
37         j=j-1;
38     }
39     a[j+1]=v;
40
41 }
42 }
```

```
↳ clang-7 -pthread -lm -o main main.c  
↳ ./main
```

Enter the Number Of Elements Of The Array

4

Enter the elements of the array

12 54 76 46

Time Taken =0.000002

Final Sorted Order Is

12 46 54 76 ↳ |

Time(in sec)

2000	0.002
4000	0.008
6000	0.019
8000	0.033
10000	0.053
12000	0.074
14000	0.100

INSERTION SORT

