EXPERIMENT NO.8.2

BFS:

```
#include<iostream>
#define max 100
using namespace std;

class BFS
{
   int adj[max][max],n,rear=0,front=1,arr[max],check[max];
   public:
   void getdata()
   {
      cout<<"How many vertices are there? : ";
      cin>>n;
      cout<<"\n\n";
      for(int i=1;i<=n;i++)
      {
         cout<<"\n";
         for(int j=1;j<=n;j++)
         {
            cout<<"Enter 1 if edge is present otherwise 0 between "<<i<<"and "<<j<<": ";
            cin>>adj[i][j];
         }
         check[i]=0;
      }
   }
   void enqueue(int element)
   {
      arr[++rear]=element;
      cout<<"\n\n"<<  arr[rear]<<" Inserted in the queue";
   }
   void dequeue()
      {
         if(front>rear)
         {
            cout<<"\nQueue Underflow";
         }
         else
         {
            cout<<"\nDeleted element is :"<<arr[front];
            front++;
         }
      }
      void graph()
```

```
    {
        for(int i=1;i<=n;i++)
        {
            if(check[i]==0)
            {
                enqueue(i);
                check[i]=1;
            }
            for(int j=1;j<=n;j++)
            {
                if(adj[i][j]==1 && check[j]==0)
                {
                    enqueue(j);
                    check[j]=1;
                }
            }
            dequeue();
        }
    }
};
int main()
{
   BFS b1;
   b1.getdata();
   b1.graph();
}
```

OUTPUT:

How many vertices are there? : 6

Enter 1 if edge is present otherwise 0 between 1and 1: 0
Enter 1 if edge is present otherwise 0 between 1and 2: 1
Enter 1 if edge is present otherwise 0 between 1and 3: 1
Enter 1 if edge is present otherwise 0 between 1and 4: 0
Enter 1 if edge is present otherwise 0 between 1and 5: 0
Enter 1 if edge is present otherwise 0 between 1and 6: 0

Enter 1 if edge is present otherwise 0 between 2and 1: 1
Enter 1 if edge is present otherwise 0 between 2and 2: 0
Enter 1 if edge is present otherwise 0 between 2and 3: 0
Enter 1 if edge is present otherwise 0 between 2and 4: 1
Enter 1 if edge is present otherwise 0 between 2and 5: 1
Enter 1 if edge is present otherwise 0 between 2and 6: 0

Enter 1 if edge is present otherwise 0 between 3and 1: 1
Enter 1 if edge is present otherwise 0 between 3and 2: 0
Enter 1 if edge is present otherwise 0 between 3and 3: 0
Enter 1 if edge is present otherwise 0 between 3and 4: 1
Enter 1 if edge is present otherwise 0 between 3and 5: 0
Enter 1 if edge is present otherwise 0 between 3and 6: 0

Enter 1 if edge is present otherwise 0 between 4and 1: 0
Enter 1 if edge is present otherwise 0 between 4and 2: 1
Enter 1 if edge is present otherwise 0 between 4and 3: 1
Enter 1 if edge is present otherwise 0 between 4and 4: 0
Enter 1 if edge is present otherwise 0 between 4and 5: 0
Enter 1 if edge is present otherwise 0 between 4and 6: 0

Enter 1 if edge is present otherwise 0 between 5and 1: 0
Enter 1 if edge is present otherwise 0 between 5and 2: 1
Enter 1 if edge is present otherwise 0 between 5and 3: 0
Enter 1 if edge is present otherwise 0 between 5and 4: 0
Enter 1 if edge is present otherwise 0 between 5and 5: 0
Enter 1 if edge is present otherwise 0 between 5and 6: 1

Enter 1 if edge is present otherwise 0 between 6and 1: 0
Enter 1 if edge is present otherwise 0 between 6and 2: 0
Enter 1 if edge is present otherwise 0 between 6and 3: 0
Enter 1 if edge is present otherwise 0 between 6and 4: 0
Enter 1 if edge is present otherwise 0 between 6and 5: 1
Enter 1 if edge is present otherwise 0 between 6and 6: 0

1 Inserted in the queue

2 Inserted in the queue

3 Inserted in the queue
Deleted element is :1

4 Inserted in the queue

5 Inserted in the queue
Deleted element is :2
Deleted element is :3
Deleted element is :4

6 Inserted in the queue
Deleted element is :5