10) THE CELEBRITY PROBLEM

METHOD 1 :
This method is brute Force method.In this method, we check for a particular column which has number of ones = n-1
If we find a particular column satisfying the above condition, then we iterate through the row having the same column number and count the number of zeroes.If the number of zeroes = n,then we return that row number , as the answer.
Else we check for another column.

METHOD 2:
Link for Explanation : The Celebrity Problem | Stacks | Love Babbar DSA Sheet | Amazon🔥 | .
This method is by using the elimination approach.
We go on checking whether the two people in order know each other or not.
That is we first assume the 0th person as the celebrity and then go on comparing it with 1,2,3,... and simultaneously go on updating the value of c .
If c knows the i'th person then c will not be the celebrity/.
So we update the value of c to i and at the end the value stored in the c has the only chance to be celebrity.

CODE FOR METHOD 1:

```
class Solution
{
  public:
  //Function to find if there is a celebrity in the party or not.
  int celebrity(vector<vector<int> >& M, int n)
  {
    for(int i=0;i<n;i++){
      int count=0;
      for(int j=0;j<n;j++){
        if(M[j][i]==1){
          count++;
        }
      }
      if(count==n-1){
        int zeroes=0;
        for(int j=0;j<n;j++){
          if(M[i][j]==0){
            zeroes++;
          }
        }
        if(zeroes==n){
          return i;
        }
      }
    }
    return -1;
  }
```

```
};
```

CODE OF METHOD 2:

```
class Solution
{
   public:
   //Function to find if there is a celebrity in the party or not.
   int celebrity(vector<vector<int> >& M, int n)
   {
      int c=0;
      for(int i=0;i<n;i++){
         if(M[c][i]==1){
            c=i;
         }
      }
      int count=0;
      for(int i=0;i<n;i++){
         if(M[c][i]==0){
            count++;
         }
      }
      if(count!=n){
         return -1;
      }
      count=0;
      for(int i=0;i<n;i++){
         if(M[i][c]==1){
            count++;
         }
      }
      if(count!=n-1){
         return -1;
      }
      return c;
   }
};
```