The Celebrity Problem

You are in a party of **N** people, where only one person is known to everyone. Such a person **may be present** in the party, if yes, **(s)he doesn't know anyone** in the party. Your task is to find the stranger (celebrity) at the party.

```
Input:
N = 3
M[][] = {{0 1 0},
         {0 0 0},
         {0 1 0}}
Output: 1
Explanation: The matrix will look like
0 1 0
0 0 0
0 1 0
Here,  the celebrity is the person with
index 1 ie id 1
```

```
Input:
N = 2
M[][] = {{0 1},
         {1 0}}
Output: -1
Explanation: The matrix will look like
0 1
1 0
Here, there is no such person who is a
celebrity (a celebrity should know no
one).
```

SOLUTION - 1

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(N)

ALGORITHM

1. Create a stack and push the number of rows in it (Matrix is of size n*n).
2. Let a = stack.top() ,
   stack.pop() and b = stack.top() ,
   stack.pop().
3. If a knows b , then push b in stack , else push a.
4. Repeat step 2 and 3 until stack.size() == 1.
5. Check for each person , if stack.top() knows anyone , there is no celebrity ,
   Else if anyone in the party doesn't know stack.top() , there is no celebrity ,
   Else stack.top() is the celebrity.

CODE:

```
#include<bits/stdc++.h>
using namespace std;

#define MAX 501

int getId(int M[MAX][MAX],int n);

int main()
{
      int T;
      cin>>T;
      int M[MAX][MAX];

      while(T--)
      {
```

```cpp
        int N;
        cin>>N;

        memset(M,0,sizeof M);

        for(int i=0;i<N;i++)
        {
            for(int j=0;j<N;j++)
            {
                cin>>M[i][j];
            }
        }
        cout<<getId(M,N)<<endl;
    }
}

// } Driver Code Ends


// The task is to complete this function

// M[][]: input matrix
// n: size of matrix (n*n)

int getId(int M[MAX][MAX], int n)
{
    //Your code here

    stack<int> st;
    int i,a,b;

    for(i=0;i<n;i++)
        st.push(i);

    while(st.size()!=1)
    {
        a=st.top();
        st.pop();
        b=st.top();
        st.pop();
        if(M[a][b]==1)
            st.push(b);

        else
        st.push(a);

    }
    int c=st.top();

    for(i=0;i<n;i++)
    {

        if(i==c)
        Continue;

        if(M[i][c]==0)
        {
            return -1;
        }

        if(M[c][i]==1)
        {
            return -1;
        }

    }
    return c;
}
```

SOLUTION - 2

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(1)

ALGORITHM

1. Define two variables , a and b which points to the index of the row , and column respectively.Take a = 0, and b = n-1 (Size of matrix = n*n).
2. If a knows b , then do a++ else do b-- .
3. Repeat step 3 while(a<b).
4. Check for each person , if someone doesnt know a or if a knows someone , then , there is no celebrity,

   Else a is the celebrity,.

CODE :

```
#include<bits/stdc++.h>
using namespace std;
#define MAX 501

int getId(int M[MAX][MAX],int n);

int main()
{
    int T;
    cin>>T;
    int M[MAX][MAX];
    while(T--)
    {
        int N;
        cin>>N;
        memset(M,0,sizeof M);
        for(int i=0;i<N;i++)
        {
            for(int j=0;j<N;j++)
            {
                cin>>M[i][j];
```

```
          }
      }
      cout<<getId(M,N)<<endl;

  }
}

bool knows(int M[MAX][MAX],int a, int b)
{
  return M[a][b];
}
int getId(int M[MAX][MAX], int n)
{
  int a = 0;
  int b = n - 1;
  while (a < b)
  {
      if (knows(M,a, b))
          a++;
      else
          b--;
  }
  for (int i = 0; i < n; i++)
  {
      if ( (i != a) &&
              (knows(M,a, i) ||
              !knows(M,i, a)) )
          return -1;
  }
  return a;
}
```