GITHUB REPOSITORY: https://github.com/Riyakumari1314/DAA-2nd-year

DESIGN AND ANALYSIS OF ALGORITHMS

EXPERIMENT

NAME:RIYA KUMARI

SAPID:590016221

BATCH:34

SUBMITTED TO:ARYAN GUPTA

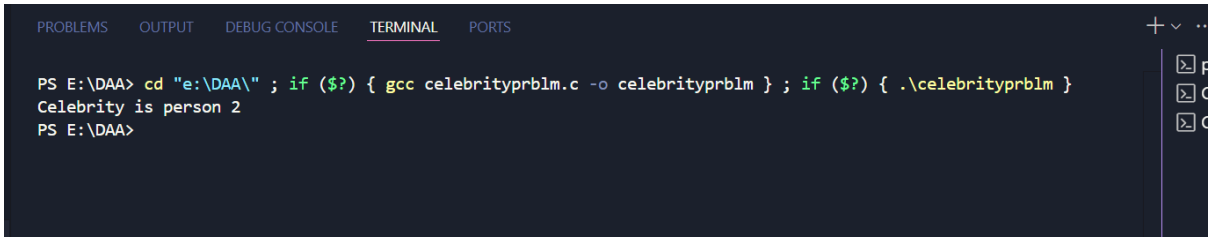
GITHUBREPOSITORY :https://github.com/Riyakumari1314/DAA-2nd-year

CODE:

```
#include <stdio.h>
M[a][b] = 1 if a knows b, else 0
int M[4][4] = {
    {0, 1, 1, 1},
    {0, 0, 1, 1},
    {0, 0, 0, 0}, // person 2 knows nobody → celebrity
    {0, 0, 1, 0}
};

// knows() API
int knows(int a, int b) {
    return M[a][b];
}

// Function to find celebrity
int findCelebrity(int n) {
    int i = 0, j = n - 1;

    // Step 1: Find candidate
    while (i < j) {
        if (knows(i, j)) {
            // If i knows j, then i cannot be celebrity
            i++;
        } else {
            // If i does not know j, then j cannot be celebrity
            j--;
        }
    }
```

```
    int cand = i;

    // Step 2: Verify candidate
    for (int k = 0; k < n; k++) {
        if (k == cand) continue;

        // Celebrity must not know anyone, and everyone must know celebrity
        if (knows(cand, k) || !knows(k, cand)) {
            return -1; // No celebrity found
        }
    }

    return cand; // Celebrity found
}

int main() {
    int n = 4;
    int celeb = findCelebrity(n);

    if (celeb == -1)
        printf("No celebrity found\n");
    else
        printf("Celebrity is person %d\n", celeb);

    return 0;
}
```

OUTPUT:



Time Complexity Analysis

- **Step 1 (Candidate Selection):**
  - Loop runs n-1 times.
  - Each step does **1 knows() query**.
  - Complexity = **O(n)**.
- **Step 2 (Verification):**
  - Loop runs n-1 times.
  - Each step does **2 knows() queries**.
  - Complexity = **O(n)**.
- **Total Complexity:**

$O(n)+O(n)=O(n)O(n) + O(n) = O(n)O(n)+O(n)=O(n)$

- **Space Complexity:**
  Only a few variables → **O(1)**.