

백준 1058 친구

202020988 조아영

# 001 문제 소개

# 002 아이디어

# 003 코드 설명

# # 문제 소개

## 문제

지민이는 세계에서 가장 유명한 사람이 누구인지 궁금해졌다. 가장 유명한 사람을 구하는 방법은 각 사람의 2-친구를 구하면 된다. 어떤 사람 A가 또다른 사람 B의 2-친구가 되기 위해선, 두 사람이 친구이거나, A와 친구이고, B와 친구인 C가 존재해야 된다. 여기서 가장 유명한 사람은 2-친구의 수가 가장 많은 사람이다. 가장 유명한 사람의 2-친구의 수를 출력하는 프로그램을 작성하시오.

A와 B가 친구면, B와 A도 친구이고, A와 A는 친구가 아니다.

## 입력

첫째 줄에 사람의 수 N이 주어진다. N은 50보다 작거나 같은 자연수이다. 둘째 줄부터 N개의 줄에 각 사람이 친구이면 Y, 아니면 N이 주어진다.

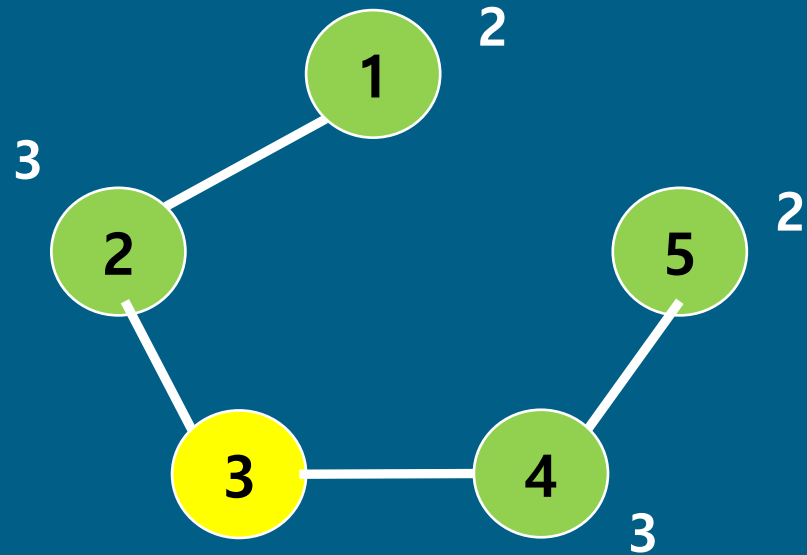
## 출력

첫째 줄에 가장 유명한 사람의 2-친구의 수를 출력한다.

# # 예시

입력 :

5  
NYNNN  
YNYNN  
NYNYN  
NNYNY  
NNNYN

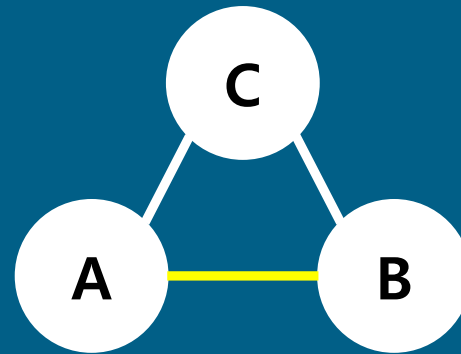


2-친구의 수 : 4(최대)

# # 아이디어



1. A와 B가 친구  
-> B는 A의 2-친구



2. A와 C가 친구, C와 B가 친구  
-> B는 A의 2-친구

그래프 탐색

- > 바로 연결되는 노드, 한 노드를 거쳐 연결되는 노드의 개수를 셈
- > 최대값 출력

# # 코드 설명

```
1 import sys
2 from collections import deque  BFS 사용 위한 queue
3
4 N = int(sys.stdin.readline())  사람 수 입력
5 graph = []
6
7 for _ in range(N):  친구 정보 입력
8     graph.append(sys.stdin.readline())
9
10 q = deque()  큐 생성
11 answer = []  2-친구의 수를 저장할 배열
```

# # 코드 설명

```
14 for i in range(N):
15     visited = [False] * N 방문 체크 위한 배열 초기화
16     q.append((i,0)) 시작위치 삽입
17     visited[i] = True
18     cnt = 0
19     while q:
20         x, y = q.popleft() pop
21
22         if y >= 2: 바로 연결되거나, 한 노드 건너 연결된 노드만 세기 위해
23             continue
24
25         for j in range(N): 친구 관계이면서 방문 안한 노드
26             if graph[x][j] == 'Y' and visited[j] == False:
27                 cnt += 1 2-친구의 수 셈
28                 visited[j] = True
29                 q.append((j,y+1)) append
30     answer.append(cnt) answer 배열에 cnt 저장
31
32 print(max(answer)) 최대값 출력
```

감사합니다