S1

단지 번호 붙이기 (Baekjoon 2667)

categories: 그래프 이론, 그래프 탐색, 너비 우선 탐색, 깊이 우선 탐색

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	139844	60757	38406	41.280%

문제

<그림 1>과 같이 정사각형 모양의 지도가 있다. 1은 집이 있는 곳을, 0은 집이 없는 곳을 나타낸다. 철수는 이 지도를 가지고 연결된 집의 모임인 단지를 정의하고, 단지에 번호를 붙이려 한다. 여기서 연결되었다는 것은 어떤 집이 좌우, 혹은 아래위로 다른 집이 있는 경우를 말한다. 대각선상에 집이 있는 경우는 연결된 것이 아니다.
<그림 2>는 <그림 1>을 단지별로 번호를 붙인 것이다. 지도를 입력하여 단지수를 출력하고, 각 단지에 속하는 집의 수를 오름차순으로 정렬하여 출력하는 프로그램을 작성하시오.

0	1	1	0	1	0	0
0	1	1	0	1	0	1
1	1	1	0	1	0	1
0	0	0	0	1	1	1
0	1	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	0	0	0

<그림 1>

0	1	1	0	2	0	0
0	1	1	0	2	0	2
1	1	1	0	2	0	2
0	0	0	0	2	2	2
0	3	0	0	0	0	0
0	3	3	3	3	3	0
0	3	3	3	0	0	0

<그림 2>

입력

첫 번째 줄에는 지도의 크기 N(정사각형이므로 가로와 세로의 크기는 같으며 5≤N≤25)이 입력되고, 그 다음 N줄에는 각각 N개의 자료(0혹은 1)가 입력된다.

출력

첫 번째 줄에는 총 단지수를 출력하시오. 그리고 각 단지내 집의 수를 오름차순으로 정렬하여 한 줄에 하나씩 출력하시오.

GitHub
Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering

S1

단지 번호 붙이기 (Baekjoon 2667)

categories: 그래프 이론, 그래프 탐색, 너비 우선 탐색, 깊이 우선 탐색

예제입력

```
7
0110100
0110101
1110101
0000111
0100000
0111110
0111000
```

예제출력

```
3
7
8
9
```

Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering

S1

단지 번호 붙이기 (Baekjoon 2667)

categories: 그래프 이론, 그래프 탐색, 너비 우선 탐색, 깊이 우선 탐색

Answer(1/2)

```python
import time
need_to_find_count = 0
found = [] # element type: tuple
dfs_search_count = 0
count_record = []
dir = [(0,1),(0,-1),(1,0),(-1,0)] # up down left right 4 way

def reset_variables():
    global need_to_find_count, dfs_search_count, found
    need_to_find_count = 0

def search_next_start_node(graph):
    for i in range(N):
        for j in range(N):
            if graph[i][j] == 1 and (i,j) not in found:
                return (i,j)
    return (-1,-1)

def dfs(graph, start_node:tuple):
    global dfs_search_count,count_record
    count = 0
    visited, stack = [[False]*N for _ in range(N)],[start_node]
    while stack:
        cur = stack.pop() # cur type: tuple
        if not visited[cur[0]][cur[1]]:
            visited[cur[0]][cur[1]] = True
            found.append((cur[0],cur[1]))
            count += 1

            # check 4 way
            for i in range(len(dir)):
                next_x = cur[0] + dir[i][0]
                next_y = cur[1] + dir[i][1]
                if 0 <= next_x < N and 0 <= next_y < N:
                    if graph[next_x][next_y] != 0 and not visited[next_x][next_y]:
                        stack.append((next_x, next_y))

    dfs_search_count += 1
    count_record.append(count)
```

Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering

S1

단지 번호 붙이기 (Baekjoon 2667)

categories: 그래프 이론, 그래프 탐색, 너비 우선 탐색, 깊이 우선 탐색

Answer(2/2)

```
41  def solve():
42      global  need_to_find_count, N, count_record
43      graph = []
44      # init graph
45      for i in range(N):
46          list_to_append = list(map(int,input()))
47          need_to_find_count += list_to_append.count(1)
48          graph.append(list_to_append)
49
50      while len(found) != need_to_find_count:
51          next_start_node = search_next_start_node(graph)
52
53          if next_start_node != (-1,-1):
54              dfs(graph, next_start_node)
55      count_record.sort()
56      print(dfs_search_count)
57      for i in range(len(count_record)):
58          print(count_record[i])
59  N = int(input())
60  solve()
```

Woody K
Contact: woody35545@gmail.com
Github: https://github.com/woody35545
CNU, Computer Science and Engineering