

문자열 압축

알파벳 대문자로 이루어진 문자열을 입력받아 같은 문자가 연속으로 반복되는 경우 반복되는 문자 바로 오른쪽에 반복 횟수를 표기하는 방법으로 문자열을 압축하는 프로그램을 작성하세요. 단 반복횟수가 1인 경우 생략합니다.

입출력 예:

s	answer
"KKHSSSSSSSE"	"K2HS7E"
"AAABCCCDD"	"A3BC3D2"
"TABCEFGTT"	"TABCEFGT2"
"GGGGGEFFFFFFK"	"G5EF6K"
"ABCDEEFGGGHI"	"ABCDE2FG3HI"

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.

잃어버린 강아지

현수는 농사지를 땅을 찾아 강아지를 데리고 산으로 들로 땅을 찾아 다니고 있었다. 숲속에서 낮잠을 자던 현수는 강아지가 도망가버려 강아지를 잃게 되었다. 강아지가 어디로 갔는지 모르는 현수는 강아지를 찾아 나섰다. 다행히 강아지에게 위치 추적기가 달려 있어 핸드폰 실시간 위성지도로 현수의 위치와 강아지의 위치, 그리고 근처의 지도를 현수는 알 수 있습니다. 지도의 크기는 항상 10*10이며, 각각의 칸에는 각각 나무, 빈칸, 강아지, 그리고 현수가 있을 수 있습니다. 지도는 다음과 같이 주어진다.

0 - 빈칸, 1 - 나무, 2 - 현수, 3 - 강아지

강아지와 현수는 항상 고정된 방법으로 지도를 다닌다. 먼저 북쪽(지도에서 위쪽)으로 출발하되, 계속 한쪽방향으로 가다가 나무나 지도의 끝에 이르면 90도 시계방향으로 회전하게 된다. 한 칸을 이동하거나, 방향을 회전할 때에는 1분이 소요된다.

만약 이동, 또는 회전을 한 후 현수와 강아지가 같은 칸에 있게 되면 현수가 강아지를 찾게 된다. 현수와 강아지가 있는 숲의 지도정보가 board에 주어지면 몇 분 후에 현수가 강아지를 찾을 수 있는지 구하는 프로그램을 작성하세요. 10,000분 후에도 찾을 수 없으면 0을 반환합니다.

입출력 예:

board	answer
<pre>[[0, 0, 0, 0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0, 2, 0, 0], [1, 0, 0, 0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 3, 0, 0, 0, 1], [0, 0, 0, 1, 0, 1, 0, 0, 0, 0], [0, 1, 0, 1, 0, 0, 0, 0, 0, 0]]</pre>	51
<pre>[[1, 0, 0, 0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 1, 0, 0, 0], [0, 0, 1, 1, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 1, 0, 0, 0, 0], [1, 0, 0, 0, 0, 0, 1, 0, 1, 0], [0, 0, 0, 1, 0, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0, 0, 2, 1], [0, 0, 0, 1, 0, 1, 0, 0, 0, 1], [0, 1, 0, 1, 0, 0, 0, 0, 0, 3]]</pre>	17

좌석 번호

세계 최고의 알고리즘 전문가인 현수의 강연을 보기위해 많은 사람들이 찾아왔습니다.

강연장에는 가로로 c 개, 세로로 r 개의 좌석이 $c \times r$ 격자형태로 배치되어 있다. 각 좌석의 번호는 해당 격자의 좌표 (x,y) 로 표시된다.

아래 그림은 가로 6개, 세로 5개 좌석으로 구성된 6×5 격자형 좌석배치입니다. 각 격자에 표시된 (x,y) 는 해당 좌석의 번호를 말합니다. 가장 왼쪽 아래의 좌석번호는 $(1,1)$ 이며, 가장 오른쪽 위 좌석의 번호는 $(6, 5)$ 이다.

(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)	(6, 5)
(1, 4)	(2, 4)	(3, 4)			(6, 4)
(1, 3)	(2, 3)				(6, 3)
(1, 2)	(2, 2)				(6, 2)
(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)

사람들은 온 순서대로 $(1,1)$ 위치 좌석부터 시작하여 시계방향으로 돌아 들어가면서 빈 좌석에 앉습니다. 만약 5번째로 온 사람은 $(1, 5)$ 좌석에 앉고, 8번째로 온 사람은 $(4, 5)$ 좌석에 앉으며, 12번째 온 사람은 $(6, 3)$ 좌석에, 20번째 온 사람은 $(2, 3)$ 좌석에 앉게됩니다.

매개변수 c 와 r 에 강연장의 크기가 주어지면, k 번째로 온 사람이 앉을 좌석번호를 반환하는 프로그램을 작성하세요.

만일 모든 좌석이 배정되어 k 번째 온 사람이 앉을 좌석이 없을 경우 $[0, 0]$ 을 반환합니다.

입출력 예:

c	r	k	answer
6	5	12	[6, 3]
6	5	20	[2, 3]
6	5	30	[4, 3]
6	5	31	[0, 0]

제한사항:

- $5 \leq c, r \leq 1,000$ 이다.
- $1 \leq k \leq 100,000,000$ 이다.

비밀번호

현수는 자신의 홍채 지문을 읽으면 비밀번호가 자동으로 입력되는 소프트웨어를 만들고 있습니다. 이 소프트웨어는 1부터 9까지의 숫자가 3 * 3 격자모양으로 되어 있는 키패드에서 비밀번호의 순서대로 이동하면서 입력되는 방식입니다.

키패드의 숫자배치는 항상 변합니다.

소프트웨어는 비밀번호의 첫 숫자에서 시작하여 이웃한(상하좌우, 대각선) 8개의 방향으로 이동하면서 입력됩니다. 이웃한 번호로의 이동시간은 1초가 걸립니다. 그리고 이웃하지 않은 숫자로의 이동은 이웃한 숫자를 통해서 이동하는 형태를 취하며, 이웃한 숫자로 이동때마다 1초씩 걸립니다.

즉 키패드에 숫자가 아래와 같이 배치되고,

1	2	3
4	5	6
7	8	9

시작위치가 2라면 1, 3, 4, 5, 6 으로 이동은 1초가 걸리고, 시작위치 2에서 7, 8, 9로는 2초가 걸립니다. 물론 번호가 입력되는 시간은 무시합니다.

매개변수 keypad에 키패드의 숫자배열이 주어지고, password에 입력해야 할 비밀번호가 주어지면 비밀번호가 모두 입력되는데 걸리는 총 시간을 반환하는 프로그램을 작성하세요.

입출력 예:

keypad	password	answer
[2, 5, 3, 7, 1, 6, 4, 9, 8]	"7596218"	8
[1, 5, 7, 3, 2, 8, 9, 4, 6]	"63855526592"	12
[2, 9, 3, 7, 8, 6, 4, 5, 1]	"323254677"	13
[1, 6, 7, 3, 8, 9, 4, 5, 2]	"3337772122"	8

제한사항:

- password의 길이는 200,000을 넘지 않습니다.

입력예제 1 설명 :

키패드의 배열은 아래와 같습니다. 비밀번호 "7596218" 를 입력하는데 걸리는 시간은

2	5	3
7	1	6
4	9	8

7(0초), 5(1초), 9(2초), 6(1초), 2(2초), 1(1초), 8(1초) 가 걸립니다.

총 시간은 0 + 1 + 2 + 1 + 2 + 1 + 1 = 8 초입니다.

스카이라인

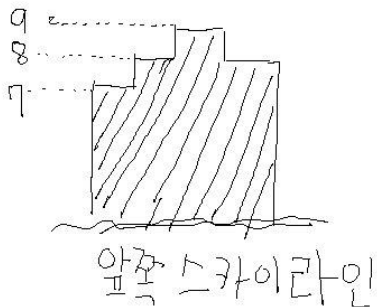
N*N 격자판에 N*N개의 도시 빌딩의 높이정보가 주어집니다.

2	5	7	3
6	8	9	7
3	2	4	5
7	2	5	8

옆쪽

앞쪽

도시를 앞에서 봤을 때 스카이라인은 왼쪽부터 해서 [7, 8, 9, 8]의 높이정보로 보이고, 옆에서 보면 위쪽에서부터 해서 [7, 9, 5, 8]입니다.



매개변수 board에 도시의 각 빌딩의 높이 정보가 주어지면 도시의 스카이라인은 변함이 없이 각 빌딩의 높이를 최대한 높이하고자 합니다. 각 빌딩의 높이를 증가시킬 수 있는 최대 높이의 합을 구하는 프로그램을 작성하세요.

입출력 예:

board	result
[[2, 5, 7, 3], [6, 8, 9, 7], [3, 2, 4, 5], [7, 2, 5, 8]]	28
[[3, 7, 6, 2], [5, 3, 8, 7], [3, 2, 5, 7], [7, 7, 5, 3]]	33
[[2, 5, 7, 3, 5], [6, 8, 9, 7, 3], [3, 2, 4, 5, 7], [7, 2, 5, 8, 6], [1, 2, 3, 4, 5]]	53
[[10, 11, 12, 11, 15], [15, 12, 13, 23, 11], [11, 13, 23, 25, 1], [11, 2, 3, 11, 13], [5, 7, 9, 10, 12]]	89

제한사항:

- board.length의 길이는 100을 넘지 않는 홀수입니다.

프로그래머스 - 외톨이 알파벳

<https://school.programmers.co.kr/learn/courses/15008/lessons/121683?language=java>

정답코드

```
from collections import defaultdict
def solution(input_string):
    s = input_string
    answer = []
    sH = defaultdict(int)
    sH[s[0]] += 1
    for i in range(1, len(s)):
        if s[i-1] != s[i]:
            sH[s[i]] += 1

    for [key, val] in sH.items():
        if val > 1:
            answer.append(key)

    if len(answer) == 0:
        return "N"
    answer.sort()
    return "".join(answer)
```

프로그래머스 Lv.1 - 대충 만든 자판

<https://school.programmers.co.kr/learn/courses/30/lessons/160586>

정답코드

```
from collections import defaultdict
def solution(keymap, targets):
    answer = []
    sH = defaultdict(int)
    for x in keymap:
        for i in range(len(x)):
            if x[i] in sH:
                if i+1 < sH[x[i]]:
                    sH[x[i]] = i+1
            else:
                sH[x[i]] = i+1

    for x in targets:
        score = 0
        for y in x:
            if y in sH:
                score += sH[y]
            else:
                score = -1
                break
        answer.append(score)
    return answer
```

프로그래머스 Lv.1 - 성격 유형 검사하기

<https://school.programmers.co.kr/learn/courses/30/lessons/118666>

정답코드

```
from collections import defaultdict
def solution(survey, choices):
    answer = []
    mark = ["RT", "CF", "JM", "AN"]
    sH = defaultdict(int)
    for i in range(len(survey)):
        score = choices[i] - 4
        if score < 0:
            sH[survey[i][0]] -= score
        else:
            sH[survey[i][1]] += score

    for x in mark:
        if sH[x[0]] < sH[x[1]]:
            answer.append(x[1])
        else:
            answer.append(x[0])

    return "".join(answer)
```


프로그래머스 Lv.0 - 정수를 나선형으로 배치하기

<https://school.programmers.co.kr/learn/courses/30/lessons/181832>

정답코드

```
def solution(n):
    answer = [[0]*n for _ in range(n)]
    dx = [-1, 0, 1, 0]
    dy = [0, 1, 0, -1]
    x, y = 0, 0
    d = 1
    k = n * n
    cnt = 1
    while(cnt < k):
        nx = x + dx[d]
        ny = y + dy[d]
        if nx < 0 or nx >= n or ny < 0 or ny >= n or answer[nx][ny] != 0:
            d = (d + 1) % 4
            continue
        answer[x][y] = cnt
        cnt += 1
        x = nx
        y = ny
    answer[x][y] = cnt
    return answer
```