

제곱수 정렬

오름차순 정렬되어 있는 길이가 n 인 수열이 주어집니다. 수열의 원소를 제공하여 오름차순 정렬한 배열을 반환하는 프로그램을 작성하세요.

주의 : sort 함수를 사용하면 안됩니다.

입출력 예:

nums	result
[-4, -1, 0, 3, 10]	[0, 1, 9, 16, 100]
[-7, -3, 2, 3, 11]	[4, 9, 9, 49, 121]

제한사항:

- nums의 길이 $3 \leq n \leq 100,000$

가장 높은 증가수열

길이가 n 인 수열이 주어지면 이 수열에서 연속된 부분 증가수열을 찾습니다. 각 부분증가수열은 높이가 있습니다. 증가수열의 높이란 증가수열의 첫항과 마지막항의 차를 의미합니다.

수열이 주어지면 여러 증가수열 중 가장 높은 부분증가수열을 찾는 프로그램을 작성하세요.

만약 수열이 [5, 2, 4, 7, 7, 3, 9, 10, 11]이 주어지면 가장 높은 부분증가수열은 [3, 9, 10, 11]이고, 높이는 8입니다.

주의 : 이웃하는 두 수가 같을 경우 증가수열로 보지 않습니다.

입출력 예:

nums	result
[8, 12, 2, 3, 7, 6, 12, 20, 3]	14
[5, 2, 4, 7, 7, 3, 9, 10, 11]	8

제한사항:

- nums의 길이 $3 \leq n \leq 100,000$
- 배열 nums의 원소는 자연수입니다.

입력예제 2 설명 :

가장 높은 부분증가수열은 [6, 20]입니다.

바이토닉 수열

바이토닉 수열이란 수열이 증가했다가 감소하는 수열을 의미합니다.

길이가 n 인 수열이 매개변수 `nums`에 주어지면 이 수열이 바이토닉 수열인지 판별하는 프로그램을 작성하세요.

만약 `[1, 2, 3, 4, 2, 1]`이면 바이토닉 수열입니다. 하지만 `[1, 2, 2, 3, 2, 1]`과 같이 같은 값이 연속으로 있으면 바이토닉 수열이라 하지 않습니다.

입출력 예:

nums	result
[1, 2, 3, 4, 5, 3, 1]	YES
[1, 3, 4, 5, 5, 6, 4, 3]	NO
[1, 2, 3, 4, 5]	NO

제한사항:

- `nums`의 길이 $3 \leq n \leq 30$
- 배열 `nums`의 원소는 자연수입니다.

최대길이 바이토닉

바이토닉 수열이란 수열이 증가했다가 감소하는 수열을 의미합니다.

길이가 n 인 수열이 주어지면 이 수열의 연속부분수열 중 가장 긴 바이토닉 수열을 찾아 그 길이를 반환하는 프로그램을 작성하세요.

만약 $[1, 3, 2, 5, 7, 4, 2, 5, 1]$ 수열이 주어지면 이 수열의 연속부분수열 중 가장 긴 바이토닉 수열은 $[2, 5, 7, 4, 2]$ 이고, 답은 5입니다.

입출력 예:

nums	result
$[1, 3, 2, 5, 7, 4, 2, 5, 1]$	5

제한사항:

- $nums$ 의 길이 $3 \leq n \leq 10,000$
- 배열 $nums$ 의 원소는 자연수입니다.

수열의 경우수

바이토닉 수열이란 수열이 증가했다가 감소하는 수열을 의미합니다.

길이가 n 인 수열이 주어지면 이 수열의 연속부분수열 중 바이토닉 수열이 몇 개 있는지 알아내는 프로그램을 작성하세요. 만약 $[1, 3, 2, 5, 7, 4, 2, 5, 1]$ 수열이 주어지면 이 수열의 연속부분 수열 중 바이토닉 수열은 $[1, 3, 2]$, $[2, 5, 7, 4]$, $[2, 5, 7, 4, 2]$, $[5, 7, 4]$, $[5, 7, 4, 2]$, $[2, 5, 1]$ 로 6개가 있습니다.

입출력 예:

nums	result
$[1, 3, 2, 5, 7, 4, 2, 5, 1]$	6

제한사항:

- nums 의 길이 $3 \leq n \leq 100$
- 배열 nums 의 원소는 자연수입니다.

거리 두기

현수는 영화관에 도착했습니다. 영화상영 시간보다 약간 늦은 현수는 남은 좌석을 빨리 선택하고 영화를 보려고 합니다.

일렬로 된 n 개의 좌석정보가 매개변수 `nums`에 주어지면, 이미 앉아 있는 사람들 중 가장 가까운 사람과 최대한 멀리 떨어져 앉을 자석을 선택해야 그 거리를 반환하는 프로그램을 작성하세요.

입출력 예:

nums	result
[1, 0, 0, 0, 1, 0, 0, 1, 0, 1]	2

제한사항:

- `nums`의 길이 $3 \leq n \leq 100$
- 배열 `nums`의 원소 1은 이미 사람이 앉은 좌석이고, 0은 빈 좌석입니다.

입력예제 1 설명 :

총 10개의 좌석이 왼쪽부터 0번 좌석으로 해서 9번 좌석까지 존재한다면 현수가 2번 좌석에 앉으면 가장 가까운 사람과의 거리가 2가 됩니다. 만약 6번 좌석에 앉으면 가장 가까운 사람과의 거리는 1입니다.

키보드

당신은 키보드의 자판 중에서 스페이스바 외에 n 개의 자판만 사용할 수 있습니다.

만약 $n=5$ 이면, 당신은 "time to time"이라는 문장을 완성할 수 있습니다.

만약 "Time to Time"라는 문장을 완성하려면 대문자 'T'를 써야 하는데 이 때는 shift키와 소문자 't'키를 누르면 됩니다. 그래서 "Time to Time"문장을 완성하기 위해서는 $n=6$ 이어야 가능합니다.

하나의 문장을 n 개의 키로 완성할 수 있으면 그 문장에 쓰인 문자개수만큼 점수를 얻을 수 있습니다. "Time to Time" 문장을 완성했을 경우 12점을 얻을 수 있습니다.

매개변수에 하나의 문장을 담은 s문자열 변수와 사용 가능한 키의 개수인 n 이 주어지면 여러분이 얻을 수 있는 점수를 리턴하는 함수를 작성하세요.

만약 문장을 완성할 수 없다면 -1을 리턴합니다.

입출력 예

s	n	result
"time to time"	5	12
"time to study"	7	-1
"Big Life is Good"	10	16
"Life is Good"	7	-1

제한사항:

- $1 \leq s$ 의 길이 $\leq 1,000$
- $1 \leq n \leq 20$

봉우리

지도 정보가 $N \times N$ 격자판에 주어집니다. 각 격자에는 그 지역의 높이가 쓰여있습니다. 각 격자판의 숫자 중 자신의 상하좌우 숫자보다 큰 숫자는 봉우리 지역입니다. 봉우리 지역이 몇 개 있는 지 알아내는 프로그램을 작성하세요.

격자의 가장자리는 0으로 초기화 되었다고 가정한다.

만약 $N=5$ 이고, 격자판의 숫자가 다음과 같다면 봉우리의 개수는 10개입니다.

0	0	0	0	0	0	0
0	5	3	7	2	3	0
0	3	7	1	6	1	0
0	7	2	5	3	4	0
0	4	3	6	4	1	0
0	8	7	3	5	2	0
0	0	0	0	0	0	0

■ 입력설명

매개변수 board에 $N \times N$ ($2 \leq N \leq 100$) 크기의 격자판 정보가 주어집니다. 각 자연수는 100을 넘지 않는다.

■ 출력설명

봉우리의 개수를 반환하세요.

■ 매개변수 형식 1

[[5, 3, 7, 2, 3], [3, 7, 1, 6, 1], [7, 2, 5, 3, 4], [4, 3, 6, 4, 1], [8, 7, 3, 5, 2]]

■ 반환값 형식 1

10

스카이라인

$N \times N$ 격자판에 $N \times N$ 개의 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.

도시의 각 빌딩들의 높이를 높이는데 도시의 스카이라인은 변함이 없이 각 빌딩의 높이를 최대한 높이하고자 합니다. 각 빌딩의 높이를 증가시킬 수 있는 최대 높이의 합을 구하는 프로그램을 작성하세요.

■ 입력설명

매개변수 board에 $N(3 \leq N \leq 100)$ 길이의 수열이 주어집니다. 수열의 원소는 자연수입니다.

■ 출력설명

증가할 수 있는 빌딩의 최대 높이의 합을 반환합니다.

■ 매개변수 형식 1

[[2, 5, 7, 3], [6, 8, 9, 7], [3, 2, 4, 5], [7, 2, 5, 8]]

■ 반환값 형식 1

28

■ 매개변수 형식 2

[[3, 7, 6, 2], [5, 3, 8, 7], [3, 2, 5, 7], [7, 7, 5, 3]]

■ 반환값 형식 2

33

회장선거

현수네 반은 오늘 회장선거를 합니다. 현수네 반 N명의 학생은 각자 자기가 좋아하는 학생을 회장후보로 추천합니다. 한 학생이 여러명을 추천할 수 있습니다.

추천횟수가 k번 이상인 학생들만 회장선거에 출마할 수 있습니다. 회장선거에 출마한 학생들은 자기를 추천해준 학생들에게 감사의 이메일을 보내기로 했습니다.

0번 학생부터 N-1번 학생까지 각 학생이 감사이메일을 받는 횟수를 알아내는 프로그램을 작성하세요.

만약 0번 학생이 3번학생을 추천 했다면 [0, 3]의 순서쌍 입력정보가 들어옵니다.

만약 반 학생이 4명이고, 각 학생의 추천정보가 [0, 1], [0, 3], [1, 2], [2, 0], [2, 3], [3, 0]이고, k=2이면 회장후보는 0번과 3번이다. 그리고 각 학생이 받는 감사이메일 횟수는 0번은 1통, 1번은 0통, 2번은 2통, 3번은 1통입니다.

■ 입력설명

매개변수 n($3 \leq n \leq 100$)에 반학생수가 전달됩니다.

매개변수 votes에 추천정보가 주어지고, 매개변수 k에 값이 전달됩니다.

모든 학생이 추천을 하는 것은 아닙니다. 할 수도 있고, 안 할 수도 있습니다.

■ 출력설명

각 학생이 받는 감사메일 횟수를 배열의 형태로 반환합니다.

■ 매개변수 형식 1

4, [[0, 1], [0, 3], [1, 2], [2, 0], [2, 3], [3, 0]], 2

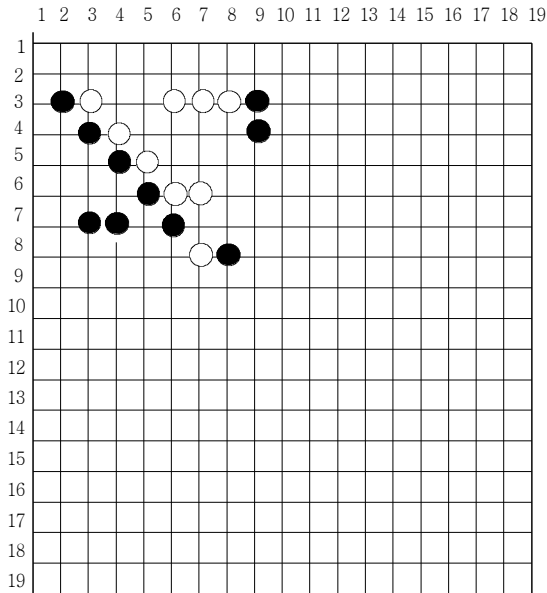
■ 반환값 형식 1

[1, 0, 2, 1]

반환값 설명 : 0번 학생은 1통, 1번은 0통, 2번은 2통, 3번은 1통입니다.

오목

오목은 바둑판에 검은 바둑알과 흰 바둑알을 교대로 놓아서 겨루는 게임이다. 바둑판에는 19개의 가로줄과 19개의 세로줄이 그려져 있는데 가로줄은 위에서부터 아래로 1번, 2번, ..., 19번의 번호가 붙고 세로줄은 왼쪽에서부터 오른쪽으로 1번, 2번, ..., 19번의 번호가 붙는다.



위의 그림에서와 같이 같은 색의 바둑알이 연속적으로 다섯 알이 놓이면 그 색이 이기게 된다. 여기서 연속적이란 가로, 세로 또는 대각선 방향 모두를 뜻한다. 즉, 위의 그림은 검은색이 이긴 경우이다. 하지만 여섯알 이상이 연속적으로 놓인 경우에는 이긴 것이 아니다. 입력으로 바둑판의 어떤 상태가 주어졌을 때, 검은색이 이겼는지, 흰색이 이겼는지 또는 아직 승부가 결정되지 않았는지를 판단하는 프로그램을 작성하시오. 단, 검은색과 흰색이 동시에 이기거나 검은색 또는 흰색이 두 군데 이상에서 동시에 이기는 경우는 입력으로 들어오지 않는다

입력설명

매개변수 board에 19줄에 각 줄마다 19개의 숫자로 표현되는데, 검은 바둑알은 1, 흰 바둑알은 2, 알이 놓이지 않은 자리는 0으로 표시되는 2차원 배열이 주어집니다.

출력설명

검은색이 이겼을 경우에는 1을, 흰색이 이겼을 경우에는 2를, 아직 승부가 결정되지 않았을 경우에는 0을 반환합니다.

▣ 매개변수 형식 1

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 2, 0, 0, 2, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
```

▣ 반환값 형식 1

1