

## 최대 길이 연속수열

매개변수 `nums`에 정렬되지 않은 수열이 주어지면 수열의 원소로 만들 수 있는 가장 긴 연속된 수열의 길이를 구하여 반환하는 프로그램을 작성하세요.

연속된 수열이란 1씩 증가하는 수열을 의미합니다.

만약 `nums = [8, 1, 9, 3, 10, 2, 4, 0, 2, 3]` 이면 이 수열의 원소로 만들 수 있는 가장 긴 연속된 수열은 `[0, 1, 2, 3, 4]`입니다.

입출력 예:

nums	answer
[8, 1, 9, 3, 10, 2, 4, 0, 2, 3]	5
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 0, 0, 0, 0]	10
[3, 3, 3, 3, 3, 3, 3, 3]	1
[-3, -1, -2, 0, 3, 3, 5, 6, 2, 2, 1, 1]	7
[-5, -3, -1, -4, 3, 3, 5, 6, 2, 2, 1, 1, 7]	3

제한사항:

- `nums`의 길이는 300,000을 넘지 않습니다.
- `nums[i]`의 값은 -1,000,000,000에서 1,000,000,000까지이며, 중복된 값도 있습니다.

입력예제 1번 설명:

만들 수 있는 가장 긴 연속수열은 `[0, 1, 2, 3, 4]`입니다.

## 자리 배치

현수네 학교는 운동장에서 전체학년이 시험을 봅니다. 같은 학년은 시험지가 같으므로 자리를 배치할 때 같은 학년끼리는 이웃해서 앉으면 안됩니다.

모든 학년은 7개 학년이며 A학년부터 G학년까지 있습니다.

자석은 일렬로 되어 있으며 전체학생들이 AAABBCDEFFG로 앉아 있다면 선생님은 같은 학년이 이웃하여 앉아 있지 않게 학생들을 ABAFABCDEF로 재배치하면 됩니다.

매개변수 s에 학생들이 마음대로 앉아있는 처음상태가 주어지면 선생님이 같은 학년은 이웃하지 않게 재배치하는 경우 중 하나를 반환하는 프로그램을 작성하세요.

모든 입력은 같은 학년이 이웃하지 않게 배치할 수 있도록 주어집니다.

입출력 예:

s	answer
"AAABBCDEFFG"	"ABAFABCDEF"
"ABCABCAD"	"ABACABCD"
"ABCDEFGGAAABDEFGFFGE"	"AFAEFGABDEFGABCDEF"
"BBCCEEFFSAAEABGGE"	"EABEABCEFGABCEFGS"
"ABABABAABCCEGG"	"ABABABACGABCEG"

제한사항:

- s의 길이는 10,000을 넘지 않습니다.
- 각 학년은 A, B, C, D, E, F, G입니다.



## 피부과

현수는 피부과를 운영합니다. 고객들이 레이저 시술을 받기위해 기다리는 대기실이 작다는 불만이 접수되어 대기실을 확장하려고 합니다. 그래서 현수는 대기실에서 동시에 대기하는 최대 인원수를 알고 싶습니다.

매개변수 laser에 각 레이저 종류별 시술을 마치는데 필요한 시간(분), enter에 각 고객의 피부과 방문시간과 해당 고객이 시술받을 레이저 종류 정보가 주어지면 시술을 받기 위해 대기실에서 동시에 기다리는 최대인원이 몇 명 발생하는지 그 명수를 반환하는 프로그램을 작성하세요. 레이저 시술이 끝난 고객은 바로 집으로 간다고 간주합니다.

입출력 예:

laser	enter	answer
[30, 20, 25, 15]	["10:23 0", "10:40 3", "10:42 2", "10:52 3", "11:10 2"]	3
[30, 20, 25, 15]	["10:23 0", "10:40 3", "10:42 2", "10:52 3", "15:10 0", "15:20 3", "15:22 1", "15:23 0", "15:25 0"]	4
[30, 20, 25, 15]	["10:20 1", "10:40 1", "11:00 1", "11:20 1", "11:40 1"]	0

제한사항:

- enter의 길이는 100,000을 넘지 않습니다.
- enter 원소의 입력순은 시간순으로 주어집니다.
- enter[i]의 첫 번째 문자는 i번 고객이 방문한 시간, 두 번째 문자는 i번 고객이 시술받을 레이저 번호입니다. 시간은 HH:MM형태이면, 09:00부터 20:00까지입니다.
- laser의 길이는 10을 넘지 않습니다.
- laser[i]의 값은 i번 레이저가 한 사람의 고객을 시술하는데 걸리는 시간입니다.

입력예제 1번 설명 :

0번 고객은 기다리지 않고 바로 10:23분에 시술에 들어가 30분동안 시술하고 10:53에 끝납니다. 이 시간동안 1번, 2번, 3번 고객이 대기실에 기다리고 있습니다. 즉 10:52분에 대기실에서 3명이 기다리게 되고 이 명수가 동시에 대기실에서 기다리는 최대인원이 됩니다.

## CPU 스케줄링

0번부터  $n-1$ 번까지 작업번호가 있는  $n$ 개의 작업이 있습니다.

각 작업은 [호출시간, 실행시간]으로 정보가 표현됩니다. 예를 들어 어떤 작업의 정보가 [2, 3] 이면 0초부터 시작하여 흐르는 시간의 흐름에서 2초에 작업호출되어 대기상태에 들어가며 자신의 차례가 되면 3초동안 실행된 뒤 종료됩니다.

cpu는 다음과 같은 규칙에 의해서 작업을 실행시킵니다.

- 1) 한 번에 한 개의 작업을 할 수 있고, 한 번 실행된 작업은 중간에 멈추지 않습니다.
- 2) 대기상태에 있는 작업이 많을 경우 그 중 실행시간이 가장 작은 작업을 먼저 처리하며, 실행시간이 같은 작업의 경우는 작업번호가 작은 것을 먼저 처리합니다.
- 3) cpu는 한 작업이 끝나면 바로 다른 작업을 할 수 있습니다. 만약 어떤 작업이 5초에 끝나면 5초에 바로 다른 작업을 할 수 있습니다.

매개변수 tasks에 0번 작업부터 순서대로 각 작업의 정보가 주어진다면 cpu가 처리하는 작업의 순서대로 작업번호를 배열에 담아 반환하는 프로그램을 작성하세요.

입출력 예:

tasks	answer
[[2, 3], [1, 2], [4, 2], [3, 1]]	[1, 3, 2, 0]
[[5, 2], [7, 3], [1, 3], [1, 5], [2, 2], [1, 1]]	[5, 4, 2, 0, 1, 3]
[[1, 2], [2, 3], [1, 3], [3, 3], [8, 2], [1, 5], [2, 2], [1, 1]]	[7, 0, 6, 1, 4, 2, 3, 5]
[[999, 1000], [996, 1000], [998, 1000], [999, 7]]	[1, 3, 0, 2]

제한사항:

- tasks의 길이는 10,000 이하의 자연수입니다.
- 각 작업의 호출시간과 실행시간은 10,000을 넘지 않습니다.

입력예제 1번 설명 :

1초에 1번 작업을 실행해서 3초에 끝냅니다.

3초에 3번 작업을 실행해서 4초에 끝냅니다.

4초에 2번 작업을 실행해서 6초에 끝냅니다.

6초에 0번 작업을 실행해서 9초에 끝냅니다.