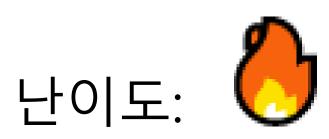
DP - 퇴사



문제1. 퇴사

시간 제한	메모리 제한
2 초	512 MB

상담원으로 일하고 있는 백준이는 퇴사를 하려고 한다. 오늘부터 N+1일째 되는 날 퇴사를 하기 위해서, 남은 N일 동안 최대한 많은 상담을 하려고 한다.

백준이는 비서에게 최대한 많은 상담을 잡으라고 부탁을 했고, 비서는 하루에 하나씩 서로 다른 사람의 상담을 잡아놓았다.

각각의 상담은 상담을 완료하는데 걸리는 기간 Ti와 상담을 했을 때 받을 수 있는 금액 Pi로 이루어져 있다.

Time	3	5	1	1	2	4	2
Price	10	20	10	20	15	40	200
DP	0	0	0	0	0	0	0

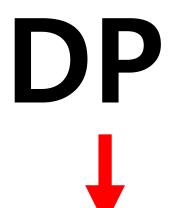
Time	3	5	1	1	2	4	2
Price	10	20	10	20	15	40	200
DP	0	0	0	0	0	0	0

불가능

Time	3	5	1	1	2	4	2
Price	10	20	10	20	15	40	200
DP	0	0	0	0	0	0	0

불가능 불가능

Time	3	5	1	1	2	4	2
Price	10	20	10	20	15	40	200
DP	0	0	0	0	15	0	0
					가능	불가능	불가능



Time	3	5	1	1	2	4	2
Price	10	20	10	20	15	40	200
DP	0	0	0	35	15	0	0

가능

가능

불가능

불가능

Time	3	5	1	1	2	4	2
Price	10	20	10	20	15	40	200
DP	0	0	45	35	15	0	0
			가능	가능	가능	불가능	불가능

Time	3	5	1	1	2	4	2
Price	10	20	10	20	15	40	200
DP	0	20 vs 45	45	35	15	0	0
		가능	가능	가능	가능	불가능	불가능

선택 안하는 게 더 높은 가치

Time	3	5	1	1	2	4	2	
Price	10	20	10	20	15	40	200	
DP	45 vs 10+35	45	45	35	15	0	0	
	가능	가능	가능	가능	가능	불가능	불가능	
선택해도 되고 안해도 된다.								

코드

```
# 퇴사 DP
n = int(input())
T, P = [0 \text{ for i in range}(n+1)], [0 \text{ for i in range}(n+1)]
for i in range(n):
   a,b = map(int, input().split())
   T[i] = a
   P[i] = b
# dp[i]는 i번째날까지 일을 했을 때, 최대값이다.
dp =[0 for i in range(n+1)]
for i in range(len(T)-2, -1, -1): # 역순으로 진행
   if T[i]+i <= n: # 날짜를 초과하지 않을 경우.
       dp[i] = max(P[i]+dp[i+T[i]], dp[i+1])
   else: # 날짜를 초과할 경우.
       dp[i] = dp[i+1]
print(dp[0])
```