

# 벼락치기

백준 14728

202021018 오윤주

# 문제

5 14728번

제출

맞힌 사람

숏코딩

재채점 결과

채점 현황

내 제출

🔗 난이도 기여

질문 검색

벼락치기

성공



5 골드 V

시간 제한

메모리 제한

제출

정답

맞힌 사람

정답 비율

2 초

256 MB

3526

1695

1370

47.818%

## 문제

ChAOS(Chung-ang Algorithm Organization and Study) 회장이 되어 일이 많아진 준석이는 시험기간에도 일 때문에 공부를 하지 못하다가 시험 전 날이 되어버리고 말았다. 다행히도 친절하신 교수님께서 아래와 같은 힌트를 시험 전에 공지해 주셨다. 내용은 아래와 같다.

1. 여러 단원을 융합한 문제는 출제하지 않는다.
2. 한 단원에 한 문제를 출제한다. 단, 그 단원에 모든 내용을 알고 있어야 풀 수 있는 문제를 낼 것이다.

이런 두가지 힌트와 함께 각 단원 별 배점을 적어 놓으셨다. 어떤 단원의 문제를 맞추기 위해서는 그 단원의 예상 공부 시간만큼, 혹은 그보다 더 많이 공부하면 맞출 수 있다고 가정하자. 이때, ChAOS 회장 일로 인해 힘든 준석이를 위하여 남은 시간 동안 공부해서 얻을 수 있는 최대 점수를 구하는 프로그램을 만들어 주도록 하자.

## 입력

첫째 줄에는 이번 시험의 단원 개수  $N(1 \leq N \leq 100)$ 과 시험까지 공부 할 수 있는 총 시간  $T(1 \leq T \leq 10000)$ 가 공백을 사이에 두고 주어진다.

둘째 줄부터  $N$  줄에 걸쳐서 각 단원 별 예상 공부 시간  $K(1 \leq K \leq 1000)$ 와 그 단원 문제의 배점  $S(1 \leq S \leq 1000)$ 가 공백을 사이에 두고 주어진다.

# 구현

예제 입력 1 복사

```
3 310
50 40
100 70
200 150
```

예제 출력 1 복사

```
220
```

1. dp 사용하기
2.  $dp[i][j]$  = i개의 단원 중에 j시간 공부
3. dp로 최댓값 구하기

# 구현

```
import sys

input = sys.stdin.readline

N, T = map(int, input().split())

time, scor = [0], [0]    // 리스트 생성

for i in range(N):
    t, s = map(int, input().split())
    time.append(t)
    scor.append(s)        // 단원과 시간 입력받기

dp = [[0 for i in range(T+1)] for j in range(N+1)]
for i in range(1, N+1):
    for j in range(1, T+1):
        if j >= time[i]: // j가 현재과목시간보다 크거나 같으면 현재과목 공부 가능
            dp[i][j] = max(dp[i-1][j], dp[i-1][j-time[i]] + scor[i])
        else:
            dp[i][j] = dp[i-1][j] // 아니면 시간 없으므로 이전 시간

print(dp[N][T])
```

**감사합니다.**