

[6 차] 문제 1

문제 설명

$n \times n$ 크기 격자 모양 정원에 칸마다 핀 꽃 또는 피지 않은 꽃을 심었습니다. 이 정원의 꽃이 모두 피는데 며칠이 걸리는지 알고 싶습니다. 핀 꽃은 하루가 지나면 앞, 뒤, 양 옆 네 방향에 있는 꽃을 피웁니다.

정원 크기 n 과 현재 정원의 상태를 담은 2 차원 배열 `garden` 이 주어졌을 때 모든 꽃이 피는데 며칠이 걸리는지 `return` 하도록 `solution` 메소드를 작성해 주세요.

매개변수 설명

정원 크기 n 과 현재 정원 상태를 담은 2차원 배열 `garden`이 `solution` 메소드의 매개변수로 주어집니다.

- 정원 크기 n 은 1보다 크고 100 보다 작거나 같은 자연수입니다.
- 정원 상태를 담은 2차원 배열 `garden`의 원소는 0 또는 1 입니다.
- 이미 핀 꽃은 1로 아직 피지 않은 꽃은 0으로 표현합니다.
- 정원에 최소 꽃 한 개는 피어 있습니다.

return 값 설명

꽃이 모두 피는데 며칠이 걸리는지 `return` 합니다.

예시

n	garden	return
3	[[0, 0, 0], [0, 1, 0], [0, 0, 0]]	2
2	[[1, 1], [1, 1]]	0

예시 설명

예시 #1

첫 날 정원은 아래와 같습니다.

0	0	0
0	1	0
0	0	0

1 일이 지난 정원의 상태는 아래와 같습니다.

0	1	0
1	1	1
0	1	0

2 일이 지난 정원의 상태는 아래와 같습니다.

1	1	1
1	1	1
1	1	1

따라서, 2 일이 지나면 정원의 모든 꽃이 핍니다.

예시 #2

첫 날 화단의 상태는 아래와 같습니다.

1	1
1	1

따라서, 0 일이 지나면 정원의 모든 꽃이 핍니다.

[6 차] 문제 2

문제 설명

한 줄에 K 자를 적을 수 있는 메모장에 영어 단어들을 적으려고 합니다. 영어 단어는 정해진 순서로 적어야 하며, 단어와 단어 사이는 공백 하나로 구분합니다. 단, 한 줄의 끝에 단어 하나를 완전히 적지 못한다면, 그 줄의 나머지 부분을 모두 공백으로 채우고 다음 줄부터 다시 단어를 적습니다.

예를 들어 한 줄에 10 자를 적을 수 있고, 주어진 단어가 순서대로 ["nice", "happy", "hello", "world", "hi"] 인 경우 각 줄에 다음과 같이 적을 수 있습니다.('_'는 공백을 나타냅니다.)

- ◆ 첫째 줄 : "nice_happy"
- ◆ 둘째 줄 : "hello____"
- ◆ 셋째 줄 : "world_hi"

이때, 둘째 줄에 "hello"를 적으면 단어를 적을 수 있는 남은 칸은 5 칸이며, "world"를 이어서 적으려면 공백 하나를 포함하여 총 6 칸이 필요합니다. 따라서 단어가 잘리게 되므로 남은 칸을 모두 공백으로 채운 후 다음 줄에 "world"부터 다시 단어를 적어 나갑니다.

한 줄에 적을 수 있는 글자 수 K 와 적을 단어가 순서대로 담긴 배열 words 가 매개변수로 주어질 때 단어를 모두 적으면 몇 줄이 되는지 return 하도록 solution 메소드를 완성해 주세요.

매개변수 설명

한 줄에 적을 수 있는 글자 수 K와 적을 단어가 순서대로 담긴 배열 words가 solution 메소드의 매개변수로 주어집니다.

- K는 5 이상 30 이하인 자연수입니다.
- words 배열의 길이는 1 이상 100 이하입니다.
- words 배열에 담겨 있는 모든 단어는 알파벳 소문자로만 이루어져 있으며, 각 단어의 길이는 K 이하입니다.

return 값 설명

단어를 모두 적으면 몇 줄이 되는지 return 해 주세요.

예시

K	words	return
10	["nice", "happy", "hello", "world", "hi"]	3

예시 설명

문제의 예시와 같으며, 단어를 모두 적으면 3 줄이 됩니다.

[6 차] 문제 3

문제 설명

자연수가 들어 있는 배열에서 숫자 K 개를 선택하려고 합니다. 이때, 선택한 숫자 중 가장 큰 수와 가장 작은 수의 차이가 최소가 되도록 해야 합니다.

예를 들어 배열에 들어 있는 숫자가 [9, 11, 9, 6, 4, 19] 이고, K = 4 라면

- 숫자 4 개를 [9, 11, 9, 6]로 뽑으면 (가장 큰 수 - 가장 작은 수) = (11 - 6) = 5 가 됩니다.
- [9, 9, 6, 4] 와 같이 숫자를 뽑아도 (가장 큰 수 - 가장 작은 수) = (9 - 4) = 5 가 됩니다.

그러나 가장 큰 수와 가장 작은 수의 차이가 5 보다 작아지도록 숫자 4 개를 선택하는 방법은 없습니다.

자연수가 들어 있는 배열 arr, 선택해야 하는 숫자 개수 K 가 매개변수로 주어질 때 선택한 숫자 중 가장 큰 수와 가장 작은 수의 차이가 최소가 되도록 arr 에서 숫자 K 개를 선택했을 때 그때의 가장 큰 수와 가장 작은 수의 차이를 return 하도록 solution 메소드를 완성해 주세요.

매개변수 설명

자연수가 들어 있는 배열 arr, 선택해야 하는 숫자 개수 K가 solution 메소드의 매개변수로 주어집니다.

- ♦ arr 배열의 길이는 5 이상 1,000 이하입니다.
- ♦ arr의 원소는 1 이상 10,000 이하인 자연수입니다.
- ♦ K 는 4 이상 50 이하인 자연수입니다.

return 값 설명

선택한 숫자 중 가장 큰 수와 가장 작은 수의 차이가 최소가 되도록 arr 에서 숫자 K 개를 선택했을 때 그때의 가장 큰 수와 가장 작은 수의 차이를 return 해 주세요.

예시

arr	K	return
[9, 11, 9, 6, 4, 19]	4	5

예시 설명

문제에 나온 예와 같습니다.

[6 차] 문제 4

문제 설명

1부터 n까지 숫자가 적힌 카드 뭉치를 다음 규칙에 따라 카드 뭉치를 섞었을 때 아래에서부터 k번째에 있는 카드에 적힌 숫자를 알아내려고 합니다. 카드 뭉치 가장 아래에는 1이 쓰여 있으며, 위로 갈수록 숫자가 커집니다. 카드 뭉치를 섞는 규칙은 다음과 같습니다.

1. 카드 뭉치를 정확히 반으로 나눕니다.
2. 두 개로 나뉜 카드 뭉치를 교차로 섞어서 카드 뭉치 하나를 만듭니다. 이때, 항상 1이 적힌 카드가 있는 뭉치를 아래로 놓고 섞습니다.
3. 위 과정을 반복합니다.

예를 들어, 1 부터 6 까지 숫자가 적힌 카드 뭉치를 3 번 섞는다면 과정별로 아래에서부터 적힌 숫자는 다음과 같습니다.

횟수	카드 뭉치
1	[1, 4, 2, 5, 3, 6]
2	[1, 5, 4, 3, 2, 6]
3	[1, 3, 5, 2, 4, 6]

이때, 아래에서 3 번째에 있는 카드에 적힌 숫자는 5 입니다.

처음에 가장 위에 있는 카드에 적힌 숫자 n, 섞는 횟수 mix, 알고 싶은 카드의 위치 k 가 매개변수로 주어질 때 아래에서부터 k 번째에 있는 카드에 적힌 숫자를 return 하도록 solution 메소드를 작성했

습니다. 그러나, 코드 일부분이 잘못되어 있기 때문에 몇몇 입력에 대해서는 올바르게 동작하지 않습니다. 주어진 코드에서 한 줄만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정하세요.

매개변수 설명

처음에 가장 위에 있는 카드에 적힌 숫자 n, 섞는 횟수 mix, 알고 싶은 카드의 위치 k가 solution 메소드의 매개변수로 주어집니다.

- n은 10 이상 50 이하인 짝수입니다.
- 카드는 0회 이상 10회 이하로 섞입니다.
- k는 1 이상 n 이하인 자연수입니다.

return 값 설명

아래에서 k 번째에 있는 카드에 적힌 숫자를 return 해 주세요.

예시

n	mix	K	return
6	3	3	5

예시 설명

문제에 나온 예와 같습니다.

[6 차] 문제 5

문제 설명

A 게임은 4x4 격자 모양의 보드의 가장 왼쪽 위에서 가장 오른쪽 아래로 말을 이동시키면서 각 구역에 있는 코인을 획득하는 게임입니다. 이때, 말은 오른쪽 또는 아래쪽으로만 이동할 수 있습니다.

예를 들어, 보드가 아래와 같다면

6	7	1	2
3	5	3	9
6	4	5	2
7	3	2	6

아래의 경우가 코인을 최대 획득할 수 있는 경우이고 이때 획득하는 코인은 38 입니다.

6	7	1	2
3	5	3	9
6	4	5	2
7	3	2	6

각 구역에서 획득할 수 있는 코인 양을 담은 2 차원 배열 board 가 매개변수로 주어질 때 최대로 획득할 수 있는 코인의 양을 return 하도록 solution 메소드를 작성했습니다. 그러나, 코드 일부가 잘못되어 있기 때문에 몇몇 입력에 대해서는 올바르게 동작하지 않습니다. 주어진 코드에서 한 줄만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정하세요.

매개변수 설명

각 구역에서 획득할 수 있는 코인 양을 담은 2차원 배열 board가 solution 메소드의 매개변수로 주어집니다.

- board는 4x4 크기인 2차원 배열입니다.
- 각 구역에서 획득할 수 있는 코인의 양은 1 이상 9 이하인 자연수입니다.

return 값 설명

최대로 획득할 수 있는 코인의 양을 return 합니다.

예시

board	return
[[6, 7, 1, 2], [3, 5, 3, 9], [6, 4, 5, 2], [7, 3, 2, 6]]	38

예시 설명

문제에 나온 예와 같습니다.

[6 차] 문제 6

문제 설명

4 x 4 크기인 정사각형 종이가 1 x 1 크기인 격자 칸으로 나누어져 있습니다. 이 종이를 가로축 혹은 세로축에 평행한 격자 선을 따라 한 번 접었을 때 만나는 격자 칸에 적힌 숫자의 합이 최대가 되도록 하려 합니다. 종이를 접을 때는 만나는 격자 칸이 정확히 일치하도록 해야 합니다.

예를 들어 다음과 같이 4 x 4 크기인 종이가 있을 때,

1	4	16	1
20	5	15	8
6	13	36	14
20	7	19	15

종이는 점선 중 하나를 따라서 접을 수 있습니다. 이때, 붉은색 점선을 따라 종이를 접으면 36과 19가 적힌 칸이 정확히 만납니다. 두 숫자의 합은 55이며, 이때가 최댓값입니다.

4 x 4 크기인 정사각형 종이의 각 격자 칸에 적힌 숫자가 담긴 배열 grid 가 매개변수로 주어질 때 종이를 접었을 때 만나게 되는 격자 칸에 적힌 숫자의 합 중 최댓값을 return 하도록 solution 메소드를 작성했습니다. 그러나, 코드 일부가 잘못되어 있기 때문에 몇몇 입력에 대해서는 올바르게 동작하지 않습니다. 주어진 코드에서 한 줄만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정하세요.

매개변수 설명

4 x 4 크기인 정사각형 종이의 각 격자 칸에 적힌 숫자가 담긴 배열 grid가 solution 메소드의 매개변수로 주어집니다.

- ♦ 각 격자 칸에 적힌 수는 1 이상 100 이하인 자연수입니다.

return 값 설명

격자 선을 따라 종이를 한 번 접었을 때 만나는 격자 칸에 적힌 숫자의 합 중 최댓값을 return 해주세요.

- ♦ 격자 선은 문제의 예시와 같이 가로, 혹은 세로 방향으로 평행한 점선을 말합니다.

예시

grid	return
[[1, 4, 16, 1], [20, 5, 15, 8], [6, 13, 36, 14], [20, 7, 19, 15]]	55

예시 설명

문제에 나온 예와 같습니다.

[6 차] 문제 7

문제 설명

UP AND DOWN 게임은 다음과 같은 규칙에 따라 진행되는 게임입니다.

- ♦ 먼저 출제자가 1 이상 ~ K 이하인 자연수 중 하나를 마음 속으로 생각합니다.
- ♦ 게임 참가자는 1 이상 ~ K 이하인 자연수 중 아무거나 하나를 말합니다.
- ♦ 만약, 참가자가 말한 숫자가 출제자가 생각한 숫자보다 작다면 출제자는 "UP"이라고 말합니다.
- ♦ 만약, 참가자가 말한 숫자가 출제자가 생각한 숫자보다 크다면 출제자는 "DOWN"이라고 말합니다.
- ♦ 참가자는 출제자가 말하는 "UP", "DOWN" 힌트를 잘 활용해서 출제자가 처음에 생각한 숫자를 맞추면 됩니다.

출제자가 처음에 생각할 수 있는 자연수 범위 K, 게임 참가자가 말한 숫자가 순서대로 담긴 배열 numbers, 게임 출제자가 참가자가 말한 각 숫자에 대해 답한 내용이 순서대로 담긴 배열 up_down 이 매개변수로 주어집니다. 배열에 주어진 순서대로 게임이 진행됐다고 했을 때 현재 정답이 될 수 있는 숫자는 몇 개인지 return 하도록 solution 메소드를 작성하려고 합니다. 빈칸을 채워 전체 코드를 완성해 주세요.

매개변수 설명

출제자가 처음에 생각할 수 있는 자연수 범위 K, 게임 참가자가 말한 숫자가 순서대로 담긴 배열 numbers, 게임 출제자가 참가자가 말한 각 숫자에 대해 답한 내용이 순서대로 담긴 배열 up_down 이 solution 메소드의 매개변수로 주어집니다.

- ♦ K는 1 이상 10,000 이하인 자연수입니다.

- numbers 배열의 길이는 1 이상 10 이하입니다.
- numbers 배열의 원소는 1 이상 K 이하인 자연수입니다.
- up_down 배열의 길이는 numbers 배열의 길이와 같습니다.
- up_down 배열은 문자열 "UP", "DOWN", "RIGHT"으로만 이루어져 있습니다.
- "RIGHT"이 나오면 정답을 맞췄다는 것이며, 게임은 더는 진행되지 않습니다.
- 항상 올바르게 게임이 진행된 경우만 입력으로 주어집니다.

return 값 설명

배열에 주어진 순서대로 게임이 진행됐다고 했을 때 현재 정답이 될 수 있는 숫자는 몇 개인지 return 해 주세요.

- 출제자가 "RIGHT"을 말한 상태로 게임이 끝났다면 정답을 맞춘 것이므로 1 을 return 하면 됩니다.

예시

K	numbers	up_down	return
10	[4, 9, 6]	["UP", "DOWN", "UP"]	2
10	[2, 1, 6]	["UP", "UP", "DOWN"]	3
100	[97, 98]	["UP", "RIGHT"]	1

예시 설명

예제 #1

말한 숫자	출제자 대답	가능성 있는 숫자
4	"UP"	5, 6, 7, 8, 9, 10
9	"DOWN"	5, 6, 7, 8
6	"UP"	7, 8

따라서 현재 상황에서 가능성 있는 숫자는 2 개입니다.

예제 #2

말한 숫자	출제자 대답	가능성 있는 숫자
2	"UP"	3, 4, 5, 6, 7, 8, 9, 10
1	"UP"	3, 4, 5, 6, 7, 8, 9, 10
6	"DOWN"	3, 4, 5

따라서 현재 상황에서 가능성 있는 숫자는 3 개입니다.

예제 #3

말한 숫자	출제자 대답	가능성 있는 숫자
97	"UP"	98, 99, 100
98	"RIGHT"	98

따라서 정답을 맞췄으므로 1 을 return 하면 됩니다.

[6 차] 문제 8

문제 설명

수열 S 가 주어질 때 이 수열의 연속된 부분 수열 중 지그재그 수열 길이의 최댓값을 구하려고 합니다.

지그재그 수열이란 첫 번째 원소부터 인접한 원소의 차이가 증가 \rightarrow 감소 \rightarrow 증가 \rightarrow 감소 ... 혹은 감소 \rightarrow 증가 \rightarrow 감소 \rightarrow 증가 ... 순으로 나타나는 수열을 말합니다. 단, 수열의 길이는 3 이상이어야 합니다.

예를 들어 수열이 [2, 5, 7, 3, 4, 6, 1, 8, 9]인 경우 연속된 부분 수열 [5, 7, 3, 4]가 부분 수열 중 가장 긴 지그재그 수열이 됩니다.

부분 수열 중 가장 긴 지그재그 수열의 길이를 구하기 위해 다음과 같이 프로그램 구조를 작성했습니다.

1. 각 숫자가 바로 이전 숫자보다 증가했는지, 혹은 감소했는지 표시한 배열을 만듭니다.
 - 1-1. "증가"는 "INC", "감소"는 "DEC"로 표시합니다.
 - 1-2. 첫 번째 숫자는 증가도, 감소도 하지 않았다는 의미에서 -1 로 표시합니다.
2. 1 단계에서 만든 증가, 감소 배열을 이용해 각 숫자를 마지막으로 하는 지그재그 수열 중 가장 긴 것의 길이를 담은 배열을 만듭니다.
 - 2-1. 바로 전 숫자가 "증가"이고 현재 숫자가 "감소"이거나, 전 숫자가 "감소"이고 현재 숫자가 "증가"이면, 현재 숫자를 마지막으로 하는 지그재그 수열 중 가장 긴 것의 길이는 (바로 전 숫자를 마지막으로 하는 지그재그 수열 중 가장 긴 것의 길이 + 1)입니다.
 - 2-2. 그렇지 않으면 현재 숫자를 마지막으로 하는 지그재그 수열 중 가장 긴 것의 길이는 2 입니다.
 - 2-3. 단, 첫 번째 숫자의 길이는 1 로 초기화합니다.
3. 2 단계에서 구한 배열에 담긴 값 중 최댓값이 부분 수열 중 가장 긴 지그재그 수열의 길이입니다.
 - 3-1. 만약 최댓값이 2 라면 0 을 return 합니다.
 - 3-2. 그 외에는 최댓값을 return 합니다.

수열이 담긴 배열 S 가 매개변수로 주어질 때 길이가 3 이상인 부분 수열 중 가장 긴 지그재그 수열의 길이를 return 하도록 solution 메소드를 작성하려고 합니다. 위 구조를 참고하여 코드가 올바르게 동작할 수 있도록 빈칸에 주어진 func_a, func_b, func_c 메소드와 매개변수를 알맞게 채워주세요.

매개변수 설명

수열이 담긴 배열 S 가 solution 메소드의 매개변수로 주어집니다.

- ♦ S 의 길이는 3 이상 100 이하입니다.
- ♦ S 의 원소는 1 이상 100 이하인 자연수이며, 같은 숫자가 중복해서 나타나지 않습니다.

return 값 설명

길이가 3 이상인 부분 수열 중 가장 긴 지그재그 수열의 길이를 return 해 주세요.

- ♦ 만약 지그재그 수열이 없다면 0 을 return 해 주세요.

예시

S	return
[2, 5, 7, 3, 4, 6, 1, 8, 9]	4
[4, 3, 2, 1, 10, 6, 9, 7, 8]	7
[1, 2, 3, 4, 5]	0

예시 설명

예제 #1

문제 예시와 같습니다.

예제 #2

[2, 1, 10, 6, 9, 7, 8]이 부분 수열 중 가장 긴 지그재그 수열입니다.

예제 #3

부분 수열 중 지그재그 수열이 없습니다.

[6 차] 문제 9

문제 설명

스택 두 개를 이용해 Queue 자료구조를 만들었을 때 Queue 자료 구조의 pop(또는 dequeue) 함수를 구현하려고 합니다. Queue란 먼저 삽입한 데이터를 먼저 빼내는 자료구조를 뜻합니다. pop 함수를 만들기 위해 다음과 같이 프로그램 구조를 작성했습니다.

1. 스택 2가 비었다면 스택 1에 아무 것도 남지 않을 때까지 스택 1에서 pop한 값을 스택 2에 push한다.
2. 스택 2에서 pop한 값을 리턴한다.

두 배열 stack1, stack2가 매개변수로 주어질 때 두 배열을 스택으로 이용해 Queue 자료 구조의 pop 함수를 구현하려고 합니다. 위 구조를 참고하여 코드가 올바르게 동작할 수 있도록 빈칸에 주어진 func_a, func_b, func_c 메소드와 매개변수를 알맞게 채워주세요.

※ 배열 index가 0인 부분을 스택의 bottom으로 생각합니다.

매개변수 설명

두 배열 stack1과 stack2가 solution 메소드의 매개변수로 주어집니다.

- ◆ stack1과 stack2는 길이가 0 이상 10 이하입니다.
- ◆ stack1과 stack2의 길이가 모두 0인 경우는 주어지지 않습니다.
- ◆ stack1과 stack2의 원소는 100 이하인 정수입니다.

return 값 설명

- ◆ stack1과 stack2로 구현한 큐에서 pop(또는 dequeue)한 값을 return 해 주세요.

예시

stack1	stack2	return
[1,2]	[3,4]	4
[1,2,3]	[]	1

예시 설명

예시 #1

stack2가 비지 않았으므로 stack2에서 pop한 결과인 4를 리턴합니다.

예시 #2

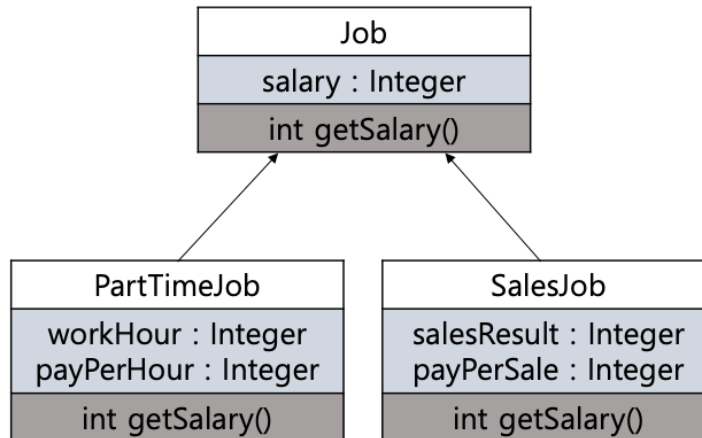
1. stack2가 비었으므로 stack1의 원소를 전부 pop해 stack2로 넣습니다. 이 결과 stack2에는 [3,2,1]이 들어갑니다.

2. stack2에서 pop한 결과인 1을 리턴합니다.

[6 차] 문제 10

문제 설명

직업에 아르바이트, 판매사원이 있습니다. 직원들 급여의 총합을 계산하기 위해 아래와 같이 Job, PartTimeJob, SalesJob 클래스를 작성했습니다.



* Job :

- ♦ Job : 직업을 나타내는 클래스입니다.
- ♦ salary : 직업의 급여를 나타냅니다. 초기 급여는 0입니다.
- ♦ getSalary : 직업의 급여를 return 합니다.

* PartTimeJob :

- ♦ PartTimeJob : 아르바이트를 나타내는 클래스이며 Job을 상속합니다.
- ♦ workHour : 아르바이트를 한 시간입니다.
- ♦ payPerHour : 아르바이트의 시간 당 급여입니다.
- ♦ getSalary : 아르바이트 급여를 계산하여 return 합니다.

* SalesJob :

- ♦ SalesJob : 판매사원을 나타내는 클래스이며 Job을 상속합니다.
- ♦ salesResult : 판매사원의 판매실적입니다.
- ♦ payPerSale : 판매실적 당 급여입니다.
- ♦ getSalary : 판매사원의 급여를 계산하여 return 합니다.

주어진 아르바이트, 판매사원 급여의 총합을 계산하려고 합니다.

* 아르바이트는 기본적으로 '아르바이트를 한 시간 X 시간 당 급여'를 받으며 40 시간 이상 근무시 8 시간만큼의 급여를 추가로 받습니다.

* 판매사원은 기본적으로 '판매실적 * 판매실적 당 급여'를 받으며 판매실적이 10 건이 넘으면 급여를 2 배로, 20 건이 넘으면 급여를 3 배로 받습니다.

아르바이트의 정보가 담긴 2 차원 배열 partTimeJobs, 판매사원의 정보가 담긴 2 차원 배열 salesJobs 가 매개변수로 주어질 때 모든 직원들 급여의 총합을 return 하도록 solution 메소드를 작성하려고 합니다. 위 클래스 구조를 참고하여 주어진 코드의 빈 칸을 적절히 채워 전체 코드를 완성해 주세요.

매개변수 설명

아르바이트의 정보가 담긴 2차원 배열 partTimeJobs, 판매사원의 정보가 담긴 2차원 배열 salesJobs 가 solution 메소드의 매개변수로 주어집니다.

- ◆ partTimeJobs은 길이가 1이상 10이하인 2차원 배열입니다.
- ◆ partTimeJobs의 각 원소는 [아르바이트를 한 시간, 시간 당 급여]입니다.
 - 아르바이트를 한 시간은 1이상 100이하입니다.
 - 시간 당 급여는 1,000이상 20,000이하입니다.
- ◆ salesJobs은 길이가 1이상 10이하인 2차원 배열입니다.
- ◆ salesJobs의 각 원소는 [판매실적, 판매실적 당 급여]입니다.
 - 판매실적은 1이상 30이하입니다.
 - 판매실적 당 급여는 1,000이상 20,000이하입니다.

return 값 설명

모든 직원들 급여의 총합을 return 합니다.

예시

partTimeJobs	salesJobs	return
[[10, 5000], [43, 6800], [5, 12800]]	[[3, 18000], [12, 8500]]	718800

예시 설명

첫 번째 아르바이트에게 50,000($10 * 5,000$)을 지급합니다.

두 번째 아르바이트에게 346,800($43 * 6,800 + 8 * 6,800$)을 지급합니다.

세 번째 아르바이트에게 64,000($5 * 12,800$)을 지급합니다.

첫 번째 영업사원에게 54,000($3 * 18,000$)을 지급합니다.

두 번째 영업사원에게 204,000($12 * 8,500 * 2$)을 지급합니다.

따라서 급여의 총합은 718,800 입니다.