

[3 차] 문제 1

문제 설명

정수로 이루어진 두 리스트 arrA와 arrB가 주어질 때, arrA를 회전해 arrB로 만들 수 있는지 알아보려 합니다. 리스트의 회전이란 모든 원소를 오른쪽으로 한 칸씩 이동시키고, 마지막 원소는 리스트의 맨 앞에 넣는 것을 말합니다.



이를 위해 다음과 같이 프로그램 구조를 작성했습니다.

1. arrA와 arrB의 길이가 다르면 false를 return 합니다.
2. 두 리스트의 구성 성분이 달라 회전했을 때 같아질 가능성이 없다면 false를 return 합니다.
3. arrA 리스트를 두 번 이어 붙여 길이가 2 배인 리스트로 만듭니다.
4. arrA의 부분 리스트 중 arrB와 같은 리스트가 있으면 true를, 그렇지 않으면 false를 return 합니다.

두 리스트 arrA와 arrB가 매개변수로 주어질 때, arrA를 회전해 arrB로 만들 수 있으면 true를, 그렇지 않으면 false를 return 하도록 solution 함수를 작성하려 합니다. 위 구조를 참고하여 코드가 올바르게 동작할 수 있도록 빈칸에 주어진 func_a, func_b, func_c 함수와 매개변수를 알맞게 채워주세요.

매개변수 설명

두 리스트 arrA와 arrB가 solution 함수의 매개변수로 주어집니다.

- ♦ arrA의 길이는 1 이상 1,000 이하입니다.
- ♦ arrA의 원소는 0 이상 1,000 이하의 정수입니다.
- ♦ arrB의 길이는 1 이상 1,000 이하입니다.
- ♦ arrB의 원소는 0 이상 1,000 이하의 정수입니다.

Return 값 설명

arrA를 회전해 arrB로 만들 수 있으면 true를, 그렇지 않으면 false를 return 해 주세요.

예시

arrA	arrB	return
[1, 2, 3, 4]	[3, 4, 1, 2]	True
[1, 2, 3, 4]	[1, 4, 2, 3]	False

예시 설명**예시 #1**

arrA 리스트를 오른쪽으로 2 칸 회전하면 arrB 와 같아집니다.

예시 #2

arrA 리스트는 회전해도 arrB 와 같아질 수 없습니다.

[3 차] 문제 2**문제 설명**

앞뒤를 뒤집어도 똑같은 문자열을 팰린드롬(palindrome)이라고 합니다. 예를 들어, "aba"는 팰린드롬이며 "abccca"는 팰린드롬이 아닙니다.

어떤 문자열의 부분 문자열 중 팰린드롬인 문자열이 여럿일 수 있습니다.

이 중 k 번째로 큰 팰린드롬을 알고 싶습니다. k 번째로 큰 팰린드롬이란, 모든 팰린드롬을 사전 순으로 나열했을 때 k 번째에 위치하는 팰린드롬을 뜻합니다. 이를 위해 다음과 같이 프로그램 구조를 세웠습니다.

1. 팰린드롬 문자열을 저장할 리스트 palindromes 를 선언합니다.
2. 주어진 문자열의 모든 부분 문자열을 찾아 다음을 수행합니다.
 - 2-1. 부분 문자열이 팰린드롬 문자열인지 확인하고, 팰린드롬 문자열이라면 palindromes 에 같은 문자열이 이미 들어 있는지 확인합니다.
 - 2-2. palindromes 에 같은 문자열이 없으면, 현재 팰린드롬 문자열을 추가합니다.
3. palindromes 를 정렬합니다.
4. 리스트 길이가 k 보다 작다면 "NULL"을 리턴합니다.
5. 그렇지 않으면 리스트의 k-1 번째 원소를 리턴합니다.

문자열 s 와 숫자 k 가 매개변수로 주어질 때, k 번째로 큰 팰린드롬 문자열을 return 하도록 solution 함수를 작성하려 합니다. 위 구조를 참고하여 코드가 올바르게 동작하도록 빈칸에 주어진 func_a, func_b, func_c 함수와 매개변수를 알맞게 채워주세요.

매개변수 설명

문자열 s와 숫자 k가 solution 함수의 매개변수로 주어집니다.

- ♦ s의 길이는 1 이상 100 이하입니다.
- ♦ s는 알파벳 소문자로만 구성됩니다.
- ♦ k는 200 이하인 자연수입니다.

Return 값 설명

k 번째로 큰 팰린드롬 문자열을 return 해 주세요.

- ♦ 단, s 로 만들 수 있는 팰린드롬 수가 k 개 미만이면 "NULL"을 return 해 주세요.

예시

s	k	return
"abcba"	4	"bcb"
"ccddcc"	7	"NULL"

예시 설명**예시 #1**

"abcba"의 부분 문자열 중 팰린드롬인 문자열을 사전 순으로 나열하면 "a", "abcba", "b", "bcb", "c" 입니다. 이 중 4 번째로 큰 팰린드롬은 "bcb"입니다.

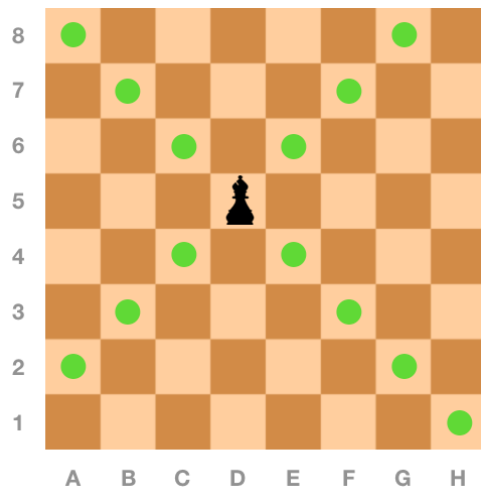
예시 #2

"ccddcc"의 부분 문자열 중 팰린드롬인 문자열을 사전 순으로 나열하면 "c", "cc", "ccddcc", "cddc", "d", "dd" 입니다. 팰린드롬인 문자열이 총 6 개이므로, 7 번째로 큰 팰린드롬은 없습니다. 따라서, "NULL"을 return 합니다.

[3 차] 문제 3**문제 설명**

체스에서 비숍(Bishop)은 아래 그림과 같이 대각선 방향으로 몇 칸이든 한 번에 이동할 수 있습니다. 만약, 한 번에 이동 가능한 칸에 체스 말이 놓여 있다면 그 체스 말을 잡을 수 있습니다.

8 x 8 크기의 체스판 위에 여러 개의 비숍(Bishop)이 놓여 있습니다. 이때, 비숍(Bishop)들에게 한 번에 잡히지 않도록 새로운 말을 놓을 수 있는 빈칸의 개수를 구하려고 합니다.



위 그림에서 원이 그려진 칸은 비숍에게 한 번에 잡히는 칸들이며, 따라서 체스 말을 놓을 수 있는 빈칸 개수는 50 개입니다.

8 x 8 체스판에 놓인 비숍의 위치 `bishops` 가 매개변수로 주어질 때, 비숍에게 한 번에 잡히지 않도록 새로운 체스 말을 놓을 수 있는 빈칸 개수를 `return` 하도록 `solution` 함수를 완성해 주세요.

매개변수 설명

체스판에 놓인 비숍의 위치 `bishops`가 `solution` 함수의 매개변수로 주어집니다.

- ◆ `bishops`는 비숍의 위치가 문자열 형태로 들어 있는 리스트입니다.
- ◆ `bishops`의 길이는 1 이상 64 이하입니다.
- ◆ 비숍이 놓인 위치는 알파벳 대문자와 숫자로 표기합니다.
 - 알파벳 대문자는 가로 방향, 숫자는 세로 방향 좌표를 나타냅니다.
 - 예를 들어 위 그림에서 비숍이 있는 칸은 "D5"라고 표현합니다.
- ◆ 한 칸에 여러 비숍이 놓이거나, 잘못된 위치가 주어지는 경우는 없습니다.

Return 값 설명

비숍에게 한 번에 잡히지 않도록 새로운 체스 말을 놓을 수 있는 빈칸의 개수를 `return` 해 주세요.

예시

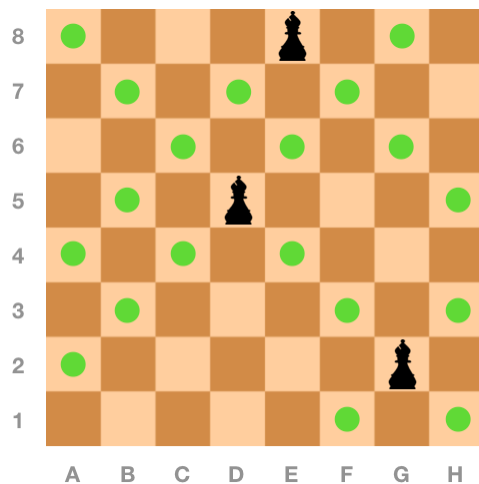
<code>bishops</code>	<code>return</code>
<code>["D5"]</code>	50
<code>["D5", "E8", "G2"]</code>	42

예시 설명

예시 #1

문제에 나온 예시와 같습니다.

예시 #2



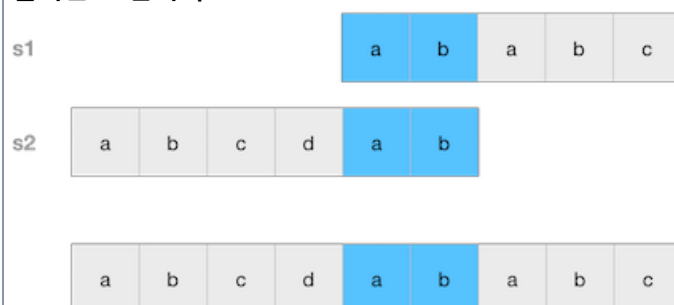
그림과 같이 원이 그려진 칸은 비숍에게 한 번에 잡히는 칸들이며, 따라서 체스 말을 놓을 수 있는 빈칸 개수는 42 개입니다.

[3 차] 문제 4

문제 설명

두 문자열 $s1$ 과 $s2$ 를 붙여서 새 문자열을 만들려 합니다. 이때, 한 문자열의 끝과 다른 문자열의 시작이 겹친다면 겹치는 부분은 한 번만 적습니다.

예를 들어 $s1 = "ababc"$, $s2 = "abcdab"$ 일 때, 아래와 같이 $s1$ 뒤에 $s2$ 를 붙이면 새 문자열의 길이는 9 입니다.



그러나 $s2$ 뒤에 $s1$ 을 붙이면 새 문자열의 길이는 8 로, 더 짧게 만들 수 있습니다.

s1 a b a b c
 s2 a b c d a b

 a b a b c d a b

두 문자열 s1 과 s2 가 매개변수로 주어질 때, s1 과 s2 를 붙여서 만들 수 있는 문자열 중 가장 짧은 문자열의 길이를 return 하도록 solution 함수를 완성해 주세요.

~~~~~

**제한 조건**

두 문자열 s1과 s2가 solution 함수의 매개변수로 주어집니다.

- s1과 s2의 길이는 1 이상 100 이하입니다.
- s1과 s2는 알파벳 소문자로만 이루어져 있습니다.

~~~~~

예시

s1	s2	return
"ababc"	"abcdab"	8

예시 설명

문제에 주어진 예시와 같습니다.

[3 차] 문제 5

문제 설명

핸드폰 화면에 문구를 출력해 주는 전광판 어플이 있습니다. 문구는 "happy-birthday"로 설정하였습니다. 전광판 어플은 다음과 같은 규칙으로 화면에 문구를 출력해 줍니다.

- 어플은 화면에 14 자 문구를 출력합니다.
- 문구는 1 초에 왼쪽으로 한 칸씩 움직입니다.
- 문구 이외의 부분은 "_"로 표시됩니다.
- 어플은 설정한 문구를 화면에 반복해 출력합니다.
- 어플은 문구가 다 지나가면 설정한 문구를 반복해서 보여줍니다.

예를 들어, 처음에는 화면에 "_____ "가 보입니다.

3 초 뒤에는 화면에 "_____hap"가 보입니다.

14 초 뒤에는 화면에 "happy-birthday"가 보입니다.

20 초 뒤에는 화면에 "birthday____"가 보입니다.

28 초 뒤에는 모든 문자열이 지나간 후 "_____"가 보입니다.

29 초 뒤에는 다시 첫 번째 문자부터 나타나며, "_____h"가 보입니다.

문구를 담은 문자열 phrases 와 초를 담은 second 가 매개변수로 주어질 때, 화면에 보이는 문자열을 출력하도록 solution 함수를 작성해 주세요.

단, '_'는 공백을 나타냅니다.

매개변수 설명

문구를 담은 문자열 phrases와 초를 담은 second가 solution 함수의 매개변수로 주어집니다.

- phrases는 "happy-birthday"입니다.
- second는 1 이상 10,000 이하인 자연수입니다.

Return 값 설명

solution 함수는 화면에 보이는 문자열을 return 합니다.

예시

phrases	second	return
"happy-birthday"	3	"_____hap"

예시 설명

화면 처음 상태인 "_____"에서 3 초가 지나면 화면에는 글자 3 개가 왼쪽으로 옮겨진 상태인 "_____hap"가 보입니다.

[3 차] 문제 6

문제 설명

어떤 수를 서로 다른 소수 3 개의 합으로 표현하는 방법의 수를 구하려 합니다.

예를 들어 33 은 총 4 가지 방법으로 표현할 수 있습니다.

1. $3+7+23$
2. $3+11+19$
3. $3+13+17$
4. $5+11+17$

자연수 n 이 매개변수로 주어질 때, n 을 서로 다른 소수 3 개의 합으로 표현하는 방법의 수를 `return` 하도록 `solution` 함수를 작성하려 합니다. 빈칸을 채워 전체 코드를 완성해 주세요.

※ 1,000 이하인 소수는 168 개 있습니다.

매개변수 설명

n 이 `solution` 함수의 매개변수로 주어집니다.

- n 은 1,000 이하인 자연수입니다.

Return 값 설명

n 을 서로 다른 소수 3 개의 합으로 표현하는 방법의 수를 `return` 해 주세요.

- 만약, n 을 서로 다른 소수 3 개의 합으로 표현할 수 없다면 0을 `return` 해 주세요.

예시

n	return
33	4
9	0

예시 설명

예시 #1

문제에 나온 예와 같습니다.

예시 #2

9는 서로 다른 세 소수의 합으로 나타낼 수 없습니다.

[3 차] 문제 7

문제 설명

카프리카 수는 다음을 만족하는 수를 뜻합니다.

- 자신의 제곱수를 둘로 나누어 더한 값이 자기 자신과 같습니다.
- 단, 둘로 나뉘는 수는 모두 양수여야 합니다.

예를 들어, 55^2 는 3,025입니다. 3,025는 3과 025, 30과 25, 302와 5로 나눌 수 있습니다. 이때 $30+25 = 55$ 이므로 55는 카프리카 수입니다.

자연수 k 가 매개변수로 주어질 때, k 이하인 모든 카프리카 수를 리스트에 담아 오름차순으로 정렬하여 `return` 하도록 `solution` 함수를 작성했습니다. 그러나, 코드 일부분이 잘못되었기 때문에

코드가 올바르게 동작하지 않습니다. 주어진 코드에서 한 줄만 변경해 모든 입력에 대해 올바르게 동작하도록 수정해 주세요.

매개변수 설명

자연수 k 가 solution 함수의 매개변수로 주어집니다.

- ♦ k 는 50 이상 100,000 이하인 자연수입니다.

Return 값 설명

k 이하인 모든 카프리카 수를 리스트에 담아 오름차순으로 정렬하여 return 해 주세요.

- ♦ 100,000 이하인 카프리카 수는 100 개보다 작습니다.

예시

k	return
500	[9, 45, 55, 99, 297]

예시 설명

$$9 \times 9 = 81 \rightarrow 8 + 1 = 9$$

$$45 \times 45 = 2,025 \rightarrow 20 + 25 = 45$$

$$55 \times 55 = 3,025 \rightarrow 30 + 25 = 55$$

$$99 \times 99 = 9,801 \rightarrow 98 + 01 = 99$$

$$297 \times 297 = 88,209 \rightarrow 88 + 209 = 297$$

[3 차] 문제 8

문제 설명

교실에 선풍기가 4 대 있습니다. 선풍기는 한 대당 학생 k 명에게 바람을 보냅니다. 모든 학생에게 바람을 보내기 위해서 선풍기를 몇 대 더 구매해야 할지 구하려고 합니다.

예를 들어, 선풍기 한 대당 학생 3 명에게 바람을 보낼 수 있을 때, 한 교실당 학생 수가 [15, 17, 19, 10, 23] 명이고 선풍기가 4 대 있으면 교실당 학생 [3, 5, 7, 0, 11] 명을 위해 선풍기를 더 구매해야 합니다. 이때, 교실 별로 선풍기가 [1, 2, 3, 0, 4] 대씩 더 필요하니 총 선풍기 10 대를 추가로 구매해야 합니다.

선풍기가 바람을 보내는 학생 수 k 와 교실별 학생 수가 담긴 리스트 student가 주어질 때, 선풍기를 최소 몇 대 사야 하는지를 return 하도록 solution 함수를 작성했습니다. 그러나, 코드 일

부분이 잘못되어 있기 때문에 몇몇 입력에 대해서는 올바르게 동작하지 않습니다. 주어진 코드에서 한 줄만 변경해서 모든 입력에 대해 올바르게 동작하도록 수정하세요.

매개변수 설명

선풍기가 바람을 보내는 학생 수 k와 교실별 학생 수가 담긴 리스트 student가 solution 함수의 매개변수로 주어집니다.

- 선풍기가 바람을 보내는 학생 수 k는 1 이상 25 이하인 정수입니다.
- 한 교실엔 1명 이상 1,000명의 학생이 있습니다.
- 교실은 1개 이상 100개 이하가 있습니다.

Return 값 설명

선풍기를 최소 몇 대 사야 하는지를 return 합니다.

예시

k	student	return
1	[4, 4, 4, 4]	0
3	[15, 17, 19, 10, 23]	10

예시 설명

예시 #1

이미 설치된 선풍기 4 대로 모든 학생을 시원하게 해 줄 수 있습니다.

예시 #2

문제에 나온 예와 같습니다.

[3 차] 문제 9

문제 설명

모 매장에서는 팝업스토어를 열려고 합니다. 팝업스토어란 한정 기간 문을 여는 매장입니다. 팝업스토어는 k 일 동안 연속해서 열 예정입니다. n 일 동안의 추정 매출액이 주어질 때, 언제 팝업스토어를 열어야 가장 매출이 높을지 알아보려 합니다.

n 일 간의 추정 매출액이 담긴 리스트 revenue 와 팝업스토어를 열 날의 수 k 가 매개변수로 주어질 때, 최대 매출액 합을 return 하도록 solution 함수를 작성했습니다. 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

추정 매출액이 담긴 리스트 `revenue`와 팝업스토어를 열 날의 수 `k`가 `solution` 함수의 매개변수로 주어집니다.

- `revenue`의 길이는 1이상 200,000 이하입니다.
- `revenue`의 원소는 10,000 이하의 자연수입니다.
- `k`는 1 이상 100,000 이하이고, `revenue`의 길이보다 작거나 같습니다.

Return 값 설명

최대 매출액 합을 `return` 해 주세요.

예시

revenue	k	return
[1, 1, 9, 3, 7, 6, 5, 10]	4	28
[1, 1, 5, 1, 1]	1	5

예시 설명

예시 #1

4 일간 매출액 합이 최대가 되는 경우는 [7, 6, 5, 10]입니다. 따라서 최대 매출액은 28 입니다.

예시 #2

1 일간 매출액 합이 최대가 되는 경우는 [5]입니다. 따라서 최대 매출액은 5 입니다.

[3 차] 문제 10

문제 설명

미용실과 레스토랑이 예약을 받는 기준은 다음과 같습니다.

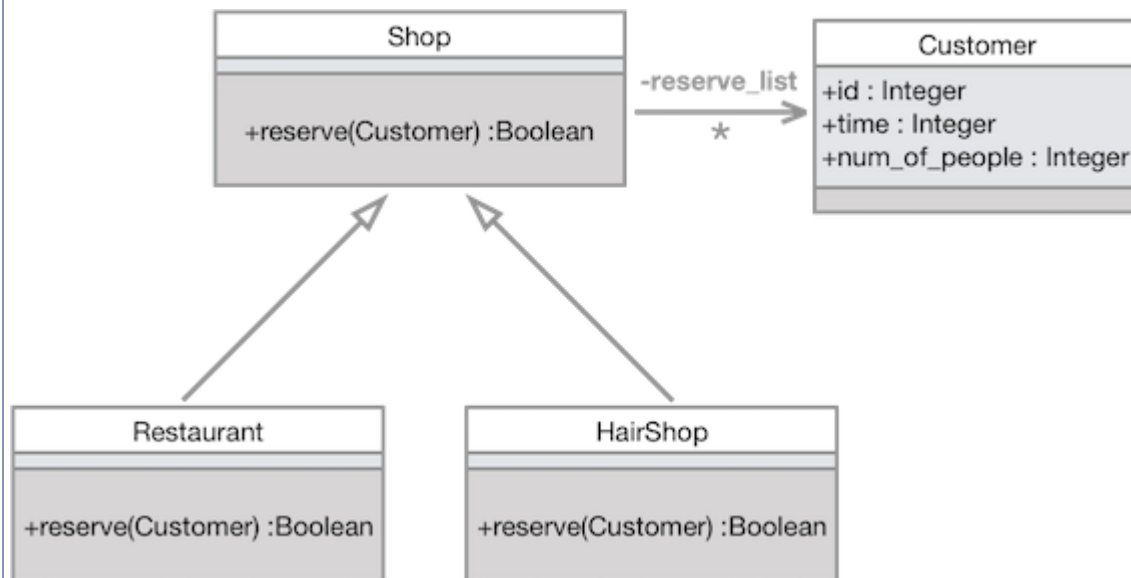
* 미용실

- * 인원수가 1명인 경우에만 예약 받습니다.
- * 다른 손님과 예약 시간이 겹칠 수 없습니다.

* 레스토랑

- * 인원수가 2명 이상 8명 이하인 경우에만 예약 받습니다.
- * 최대 두 팀까지 예약 시간이 겹칠 수 있습니다.

두 가게에서 예약을 받은 횟수를 계산하기 위해 다음과 같이 Customer, Shop, HairShop, Restaurant 클래스를 작성했습니다.



*** Customer:**

- ♦ Customer : 고객을 나타내는 클래스입니다.
- ♦ id : 고객 식별 번호를 나타냅니다.
- ♦ time : 고객이 신청한 예약 시간을 나타냅니다.
- ♦ num_of_people : 예약 인원 수를 나타냅니다.

*** Shop :**

- ♦ Shop : 가게를 나타내는 클래스입니다.
- ♦ reserve_list : 가게에 예약한 고객 리스트입니다.
- ♦ reserve : 고객을 매개변수로 받아 예약 고객 리스트에 추가 후 true를 return합니다.

*** HairShop :**

- ♦ HairShop : 미용실을 나타내는 클래스이며, Shop을 상속합니다.
- ♦ reserve : 고객을 매개변수로 받아 미용실의 예약 기준에 맞는지 검사합니다. 예약을 받았다면 예약 고객 리스트에 추가 후 true를 return하고, 그렇지 않으면 false를 return합니다.

*** Restaurant :**

- ♦ Restaurant는 레스토랑을 나타내는 클래스이며, Shop을 상속합니다.

- ♦ **reserve** : 고객을 매개변수로 받아 레스토랑의 예약 기준에 맞는지 검사합니다. 예약을 받았다면 예약 고객 리스트에 추가 후 **true**를 **return**하고, 그렇지 않으면 **false**를 **return** 합니다.

예약을 원하는 고객 정보가 담긴 리스트 **customers**와 **shops**가 매개변수로 주어질 때, 두 가게에서 예약 받은 횟수를 **return** 하도록 **solution** 함수를 작성하려고 합니다. 위 클래스 구조를 참고하여 주어진 코드의 빈칸을 적절히 채워 전체 코드를 완성해 주세요.

매개변수 설명

예약을 원하는 고객 정보가 담긴 리스트 **customers**와 **shops**가 **solution** 함수의 매개변수로 주어집니다.

* **customers**의 길이와 **shops**의 길이는 항상 같습니다.

* **customers**의 각 원소에는 고객 정보가 [고객 ID, 예약 시간, 예약 인원수] 순으로 들어 있습니다.

* 고객 ID는 1 이상 10,000 이하인 자연수입니다.

* 예약 시간은 0 이상 23 이하인 정수입니다.

* 예약 인원수는 1 이상 10 이하인 자연수입니다.

* 고객 ID는 중복해서 주어지지 않습니다.

* **customers**에는 고객이 예약을 신청한 순서대로 들어 있습니다.

* **shops**는 각 고객이 예약할 가게를 나타내며, "hairshop" 또는 "restaurant" 로만 구성되어 있습니다.

* **i** 번째 고객이 예약할 가게는 **shops[i]** 입니다.

* 예약을 신청한 고객의 수는 1 이상 10 이하입니다.

Return 값 설명

- ♦ 두 가게에서 예약 받은 횟수를 **return** 해 주세요.

예시

customers	shops	return
[[1000, 2, 1], [2000, 2, 4], [1234, 5, 1], [4321, 2, 1], [1111, 3, 10]]	["hairshop", "restaurant", "hairshop", "hairshop", "restaurant"]	3

예시 설명

고객별 예약 정보는 다음과 같습니다.

ID	예약 시간	인원수	가게
1000	2	1	hairshop
2000	2	4	restaurant

1234	5	1	hairshop
4321	2	1	hairshop
1111	3	10	restaurant

- * 첫 번째 고객은 미용실에서 예약 받습니다.
- * 두 번째 고객은 레스토랑에서 예약 받습니다.
- * 세 번째 고객은 미용실에서 예약 받습니다.
- * 네 번째 고객은 예약 받지 않습니다. 2 시에 미용실을 먼저 예약한 손님이 있기 때문입니다.
- * 다섯 번째 고객은 예약 받지 않습니다. 레스토랑에 한 번에 예약 가능한 인원수는 2 명 이상 8 명 이하입니다.