TEST 8

1.

1937년 Collatz란 사람에 의해 제기된 이 추측은, 주어진 수가 1이 될 때까지 다음 작업을 반복하면, 모든 수를 1로 만들 수 있다는 추측입니다. 작업은 다음과 같습니다.

1-1. 입력된 수가 짝수라면 2로 나눕니다.

1-2. 입력된 수가 홀수라면 3을 곱하고 1을 더합니다.

2. 결과로 나온 수에 같은 작업을 1이 될 때까지 반복합니다.

예를 들어, 주어진 수가 6이라면 6 -> 3 -> 10 -> 5 -> 16 -> 8 -> 4 -> 2 -> 1 이 되어 총 8번 만에 1이 됩니다. 위 작업을 몇 번이나 반복해야 하는지 반환하는 함수, solution을 완성해주세요.

단, 주어진 수가 1인 경우에는 0을, 작업을 500번 반복할 때까지 1이 되지 않는다면 -1을 반환해 주세요.

제한사항

▶ 입력된 수, num은 1 이상 8,000,000 미만인 정수입니다.

입출력 예

n	result
6	8
16	4
626331	-1

#2> 이 되어 총 4번 만에 1이 됩니다.

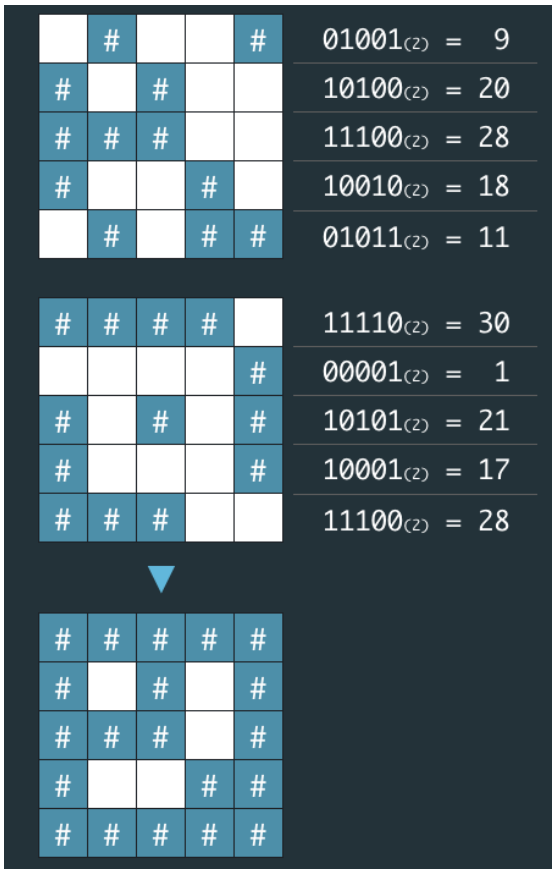
#3> 626331은 500번을 시도해도 1이 되지 못하므로 -1을 리턴해야합니다.

TEST 8

2.

진하는 평소 응경이가 비상금을 숨겨놓는 장소를 알려줄 비밀지도를 손에 넣었습니다. 그런데 이 비밀지도는 숫자로 암호화되어 있어 위치를 확인하기 위해서는 암호를 해독해야 합니다. 다행히 지도 암호를 해독할 방법을 적어 놓은 메모도 함께 발견했습니다.

1. 지도는 한 변의 길이가 n 인 정사각형 배열 형태로, 각 칸을 공백(“ “) 또는 벽(“#”) 두 종류로 이루어져 있습니다.
2. 전체 지도는 두 장의 지도를 겹쳐서 얻을 수 있습니다. 각각 “지도 1” 과 “지도 2” 라고 하겠습니다. 지도 1 또는 지도 2 중 어느 하나라도 벽인 부분은 전체 지도에서도 벽입니다.
3. 지도 1과 지도 2는 각각 정수 배열로 암호화되어 있습니다.
4. 암호화된 배열은 지도의 각 가로줄에서 벽 부분을 1, 공백 부분을 0으로 부호화했을 때 얻어지는 이진수에 해당하는 값의 배열입니다.



진하가 응경이의 비상금을 손에 넣을 수 있도록, 비밀지도의 암호를 해독하는 작업을 도와줄 프로그램을 작성해주세요.

TEST 8

제한사항

▶ 입력으로 지도의 한 변 크기 n 과 2개의 정수 배열 arr1, arr2가 주어집니다.

> $1 \leq n \leq 16$

> arr1, arr2는 길이 n인 정수 배열로 주어집니다.

> 정수 배열의 각 원소 x를 이진수로 변환했을 때의 길이는 n 이하입니다. 즉, $0 \leq x \leq 2^n-1$을 만족합니다.

> 원래의 비밀지도를 해독하여 “#”, 공백 으로 구성된 문자열 배열로 출력해주세요.

입출력 예

n	arr1	arr2	Result
5	[9, 20, 28, 18, 11]	[30, 1, 21, 17, 28]	[“#####”, “# # #”, “### #”, “# ## “, “#####”]
6	[46, 33, 33, 22, 31, 50]	[27, 56, 19, 14, 14, 10]	[“######”, “### # “, “## ## “, “ #### “, “ #####”, “### # “]

TEST 8

3.

주차장의 요금표와 차량이 들어오고(입차) 나간(출차) 기록이 주어졌을 때, 차량별로 주차 요금을 계산하려고 합니다. 아래는 하나의 예시를 나타냅니다.

요금표

기본 시간(분)	기본 요금(원)	단위 시간(분)	단위 요금(원)
180	5000	10	600

입/출차 기록

시각(시 : 분)	차량 번호	내역
05 : 34	5961	입차
06 : 00	0000	입차
06 : 34	0000	출차
07 : 59	5961	출차
07 : 59	0148	입차
18 : 59	0000	입차
19 : 09	0148	출차
22 : 59	5961	입차
23 : 00	5961	출차

자동차별 주차 요금

차량 번호	누적 주차 시간(분)	주차 요금(원)
0000	34 + 300 = 334	5000 + [(334 - 180) / 10] x 600 = 14600
0148	670	5000 + [(670 – 180) / 10] x 600 = 34400
5961	145 + 1 = 146	5000

TEST 8

어떤 차량이 입차된 후에 출차된 내역이 없다면, 23:59에 출차된 것으로 간주합니다.

> 0000 번 차량은 18:59에 입차된 이후, 출차된 내역이 없습니다. 따라서, 23:59에 출차된 것으로 간주합니다.

00:00 부터 23:49 까지의 입/출차 내역을 바탕으로 차량별 누적 주차 시간을 계산하여 요금을 일괄로 정산합니다.

누적 주차 시간이 기본 시간 이하라면, 기본 요금을 청구합니다.

누적 주차 시간이 기본 시간을 초과하면, 기본 요금에 더해서, 초과한 시간에 대해서 단위 시간마다 단위 요금을 청구합니다.

> 초과한 시간이 단위 시간으로 나누어 떨어지지 않으면, 올림 합니다.

> [a] : a보다 작지 않은 최소의 정수를 의미합니다. 즉, 올림 을 의미합니다.

주차 요금을 나타내는 정수 배열 fees, 자동차의 입/출차 내역을 나타내는 문자열 배열 records가 매개변수로 주어집니다. 차량 번호가 작은 자동차부터 청구할 주차 요금을 차례대로 정수 배열에 담아서 return 하도록 solution 함수를 완성해주세요.

TEST 8

제한사항

▶ fees의 길이 = 4

> fees[0] = 기본 시간(분) / 1 ≤ fees[0] ≤ 1,439

> fees[1] = 기본 요금(원) / 0 ≤ fees[1] ≤ 100,000

> fees[2] = 단위 시간(분) / 1 ≤ fees[2] ≤ 1,439

> fees[3] = 단위 요금(원) / 1 ≤ fees[3] ≤ 10,000

▶ 1 ≤ records의 길이 ≤ 1,000

> records의 각 원소는 "시각 차량번호 내역" 형식의 문자열입니다.

> 시각, 차량번호, 내역은 하나의 공백으로 구분되어 있습니다.

> 시각은 차량이 입차되거나 출차된 시각을 나타내며, HH:MM 형식의 길이 5인 문자열입니다.

- HH:MM 은 00:00 부터 23:59 까지 주어집니다.

> 차량번호는 자동차를 구분하기 위한, 0~9로 구성된 길이 4인 문자열입니다.

> 내역의 IN 은 입차를, OUT은 출차를 의미합니다.

> records의 원소들은 시각을 기준으로 오름차순으로 정렬되어 주어집니다.

> records는 하루 동안의 입/출차된 기록만 담고 있으며, 입차된 차량이 다음날 출차되는 경우는 입력으로 주어지지 않습니다.

> 같은 시각에, 같은 차량번호의 내역이 2번 이상 나타내지 않습니다.

> 마지막 시각(23:59)에 입차되는 경우는 입력으로 주어지지 않습니다.

> 아래의 예를 포함하여, 잘못된 입력은 주어지지 않습니다.

- 주차장에 없는 차량이 출차되는 경우

- 주차장에 이미 있는 차량 (차량번호가 같은 차량) 이 다시 입차되는 경우

TEST 8

fees	records	result
180 5000 10 600	["05:34 5961 IN", "06:00 0000 IN", "06:34 0000 OUT", "07:59 5961 OUT", "07:59 0148 IN", "18:59 0000 IN", "19:09 0148 OUT", "22:59 5961 IN", "23:00 5961 OUT"]	14600, 34400, 5000
120 0 60 591	["16:00 3961 IN", "16:00 0202 IN", "18:00 3961 OUT", "18:00 0202 OUT", "23:58 3961 IN"]	0, 591
1 461 1 10	["00:00 1234 IN"]	14841

#1> 문제 예시와 같습니다.

#2>

차량 번호	누적 주차 시간(분)	주차 요금(원)
0202	120	0
3961	120 + 1	0 + [(121 - 120) / 60] x 591 = 591

: 3961번 차량은 2번째 입차된 후, 출차된 내역이 없으므로, 23:59에 출차되었다고 간주합니다.

#3>

차량 번호	누적 주차 시간(분)	주차 요금(원)
1234	1439	461 + [(1439 - 1) / 1] x 10 = 14841

: 1234번 차량은 출차 내역이 없으므로, 23:59에 출차되었다고 간주합니다.