Parking System

Time Limit 1000 ms

세호는 주차장을 관리하고 있다.

이 주차장에는 최대 50,000대까지 관리가 가능한 시스템을 갖추고 있다.

주차 시간 관리는 10분단위로 이뤄지며 10분을 1 term이라고 한다.

매 tick 마다 1 term 씩 시간이 흘러간다.

차를 parking 하게 되면 알파벳 대문자와 숫자로 이뤄진 5~10 자리 차 번호와 알파벳 소문자와 숫자로 이뤄진 5~10자리 코드를 받는다.

차번호는 차를 인식하기 위해서 사용되고 코드는 주인이 맞는지 확인하기 위해서 사용된다.

각 차들은 각 차마다 정해진 기간 동안만 주차를 하도록 계약이 되어 있다.

세호가 주차관리를 시작하기 전에 현재 주차되어 있는 차량들의 정보를 받는다.

이때 전달되는 정보는 차번호, 코드 그리고 계약 기간이 전달 된다.

계약 기간은 term으로 주어지며 계약 시간이 만료되면 출차를 해야 한다.

세호는 매 tick마다 출차되어야 하는 차가 몇대인지 알고 싶다.

차는 주차 계약 기간이 만료되지 않았어도 출차를 할 수 있다.

그리고 각 차주들은 주차중인 차에 대해서 계약 갱신을 할 수 있다.

계약을 갱신하게 되면 그 시점부터 다시 계약이 시작되므로 정해진 계약 기간 이후에 출차를 하게 된다.

가령 현재 tick이 3이라고 하고 주차를 할 때 계약 기간을 5로 정했다면 tick이 8이 되었을때 출차를 해야 한다.

그런데 tick이 6이 되었을 때 재 계약을 했다면 해당 차량의 출차 시기는 8 tick이 아니라 11 tick으로 변경이 된다.

주차장에서는 본인이 아닌 사람이 재계약을 하지 못하도록 코드를 확인하고 있다.

차번호와 코드번호가 기존 계약 정보와 일치하지 않는 경우 계약 갱신은 무효가 된다.

다음에 주어지는 함수들을 작성하라.

**void** Init();

: 매 TC 초기에 불려지며 주차장에 차가 한 대도 없는 것처럼 초기화 한다.

**void** Parking(**char** number[11], **char** code[11], int contractTerm);

: 차를 주차한다.

: 차넘버와 신분확인용 코드 그리고 계약 기간이 주어진다.

: 차넘버는 알파벳 대문자와 숫자의 조합이며 길이는 5~10이다.

: 코드는 알파벳 소문자와 숫자의 조합이며 길이는 5~10이다.

: 계약 기간은 1~3,000 사이의 숫자로 주어진다.

: 이미 주차되어 있는 차량에 대해서는 다시 이 함수를 호출하지 않는다.

: 이전에 출차했던 차량은 다시 주차하지 않는다.

**void** Unparking(**char** number[11]);

: 계약 기간이 만료되지 않는 주차된 차를 출차시킨다.

: 출차를 직접 확인하기 때문에 Tick() 함수를 통해 확인되는 출차될 차량 수에 포함시키지 않는다.

: 간혹 시스템 오류로 인해 주차장에 있지 않는 차번호로 출차 정보가 들어오는 경우가 있다. 무시하면 된다.

**void** Renew(**char** number[11], **char** code[11]);

: 주차되어 있는 차량의 계약 기간을 갱신한다.

: 간혹 시스템 오류로 인해 주차장에 있지 않는 차에 대해서 계약 갱신이 요청되는 경우가 있다. 무시하면 된다.

: 주차된 차량이더라도 코드가 다를 경우 계약을 갱신해주면 안된다.

int Tick():

: 이 함수가 불려질 때마다 한 term의 시간이 흘러간다.

: 만약 변경된 시점에 출차해야 하는 차량이 있다면 총 몇대인지를 반환한다.

: 가량 현재 3 tick이고 4 tick에 출차해야 하는 차량이 2대 있다고 하는 경우 이 함수가 호출되면서 4 tick이 되면서 2를 반환해야 한다.

: 만약 출차해야 하는 차가 없다면 0을 반환한다.

: 계약 기간이 만료되어 출차하게 되는 차는 반드시 출차하게 되어 주차장에 존재 하지 않는다.

최대 계약기간은 3,000이며 매 TC는 최대 3,000 tick까지만 확인한다.