



# Station

## Move from one station to another

Taiwan has a rail system that connects all train stations. The rail system has  $n$  stations indexed from 0 to  $n - 1$ . Every two adjacent train stations are 1 kilometer apart, and some stations have lodge service. Also the first and the last stations do have lodge service.

Jian-Jia wants to travel through Taiwan along this railway system. Jian-Jia will start from the first station and stops at the last station. Since Jian-Jia bought a discount ticket, he can only travel at most  $k$  kilometers per day. In addition, Jian-Jia only wishes to stop at stations that have lodge service. Please determine the minimum number of days for Jian-Jia to travel from the first to the last station.

## Example

We assume that there are 10 stations, and station 0, 1, 2, 3, 4, 6, 7, 9 have lodge service. Let  $k$  be 4, i.e., Jian-Jia can only travel 4 kilometers per day, then he needs at least 3 days to travel from station 0 to 9. For example, he can move from station 0 to station 3 in the first day, station 3 to 7 in the second day, and station 7 to 9 in the third day. If  $k$  is 1 then it is *impossible* to travel from the first station to the last station.

## Statement

Given the locations of all stations with lodge service and the limit  $k$ , determine whether it is possible for Jian-Jia to travel from the first station to the last station, and if possible, the minimum number of days for Jian-Jia to do so. You will implement the following function.

- `findMinDays(n, k, lodge)`
  - $n$  is the number of stations.
  - `lodge` is the array indicating whether a station has lodge service. For example, if station  $i$  has lodge service then `lodge[i]` will be 1, and 0 otherwise. We assume that both `lodge[0]` and `lodge[n-1]` are 1.
  - The return value is the minimum number of days for Jian-Jia to travel from the first station to the last station if possible. If not possible then the return value is -1.

## Subtask 1 (10 points)

- $2 \leq n \leq 200$ , and  $1 \leq k \leq 20$ .

### Subtask 2 (20 points)

- $2 \leq n \leq 50000$ , and  $1 \leq k \leq 20$ .

### Subtask 3 (70 points)

- $2 \leq n \leq 500000$ , and  $1 \leq k \leq 3000$ .

## Implementation details

You have to submit exactly one file, called `station.c`, `station.cpp` or `station.pas`. This file implements the subprograms described above using the following signatures. You also need to include a header file `station.h`.

### C/C++ program

```
#include "station.h"
int findMinDays(int n, int k, int lodge[]);
```

### Pascal program

```
type
  LodgeType = array[0..499999] of longint;
function findMinDays(n, k: longint; lodge: LodgeType): longint;
```

This subprogram must behave as described above. Of course you are free to implement other subprograms for their internal use. Your submissions must not interact in any way with standard input/output, nor with any other file.

### Sample grader

The sample grader reads the input in the following format, and print the result from `findMinDays`.

- line 1:  $n$ , the number of stations.
- line 2: `lodge[0]`, ..., `lodge[n - 1]`.