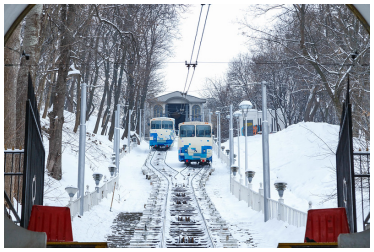
F Funicular Frenzy

You are on a skiing trip in the Alps and need to take a funicular.[1] However, there usually is a long queue for the funicular to bring you to the top of the mountain. Being someone who hates wasting time in the morning, you want to find the best moment to start queueing in order to minimize queueing time.



Two funicular carriages passing each other on the halfway point of the track. CC BY 2.0 by Marco Verch on Flickr

The funicular station is open for n minutes per day. A carriage transports c people at once, and one carriage leaves exactly every minute. For every minute the funicular is open today, you know the number of people arriving.

You want to arrive when the station is open, exactly at the start of a minute, like everyone else. Note that you are a sociable person and if there are other people arriving at the same minute as you, you let them go first, after which you stand in the queue.

Calculate at which minute you should arrive to have minimal waiting time, or determine that it is impossible to catch a ride today. If there are two times achieving the same minimal waiting time, give the earliest occasion.

Input

The input consists of:

- One line with two integers n and c ($1 \leq n \leq 10^5$, $1 \leq c \leq 10^9$), the number of minutes the funicular is open today and the number of people one funicular carriage takes up the mountain each minute.
- One line with n integers $a_0, \ldots, a_{n-1}$ ($0 \leq a_i \leq 10^9$), the number of new people showing up i minutes after the funicular opens.

Output

If it is not possible to take the funicular today, output "`impossible`". Else, output the least number of minutes after opening time you should enter the queue, such that the waiting time is minimized.

Sample Input 1	Sample Output 1
`5 1` `5 0 0 0 0`	`impossible`

[1] A funicular is a type of cable railway system laid on a steep slope, where two counterbalanced carriages are attached to opposite ends of a haulage cable, which is looped over a pulley at the upper end of the track.

Sample Input 2

```
5 4
8 6 4 2 0
```

Sample Output 2

```
0
```

Sample Input 3

```
10 10
12 11 10 9 8 7 6 5 4 3
```

Sample Output 3

```
5
```