UVa Online Judge

12104 A Rod in a Path

Alice has a rod. One day, she draws a path on a grid and puts the rod on it. The path begins at $(0, 0)$ and continues to the right. The first and last segments are always horizontal, so there are always an odd number of segments. If we number the segments $1, 2, \ldots, n$, odd-numbered segments are all horizontal, while other segments (if any) are vertical. Initially, one endpoint B of the rod is located at $(0, 0)$, and the other endpoint A is at $(L, 0)$, where L is the length of the rod. The length of the first segment is at least L.

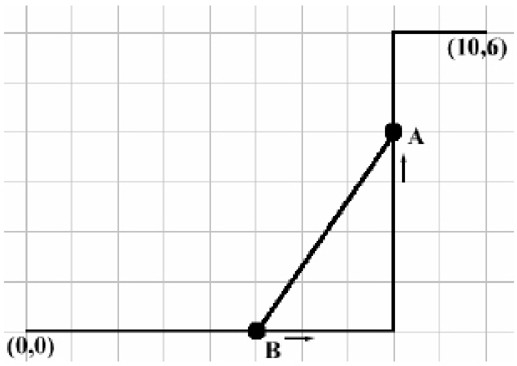

Fig 1. The grid, the path and the rod

When moving the rod, both endpoints A and B must be always on the path, though other parts may be outside. The rod is hard, so its length (i.e. distance between A and B) is always L.

Write a program to compute the minimum distance A must cover to reach the rightmost endpoint of the path.

Input

The input consists of several test cases. The first line of each case contains two integers n and L ($1 \le n \le 10, 1 \le L \le 30$), described above. The second line contains n non-zero integers l_i ($-30 \le l_i \le 30$), the lengths and directions of path segments. The absolute value of l_i denotes length of the i-th segment. If it is horizontal, l_i is positive. That means, horizontal segments are always left-to-right. If it is vertical, positive means down-to-up (increasing y coordinate), negative means up-to-down (decreasing y coordinate). The last test case is followed by a single zero, which should not be processed.

Output

For each test case, print the case number and the minimum distance to two decimal places. If it's not possible to reach the rightmost point, print '-1'.

Sample Input

```
3 5
8 6 2
5 2
3 1 1 -4 1
0
```

Sample Output

```
Case 1: 11.00
Case 2: 10.00
```