Tree Construction

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given three integers n, d and l. Find a tree of diameter d with n nodes, out of which l are leaves, or claim that such a tree does not exist.

The **diameter** of a tree is defined as the maximum pairwise distance between two nodes of the tree, where the distance between two nodes equals the number of edges on the unique path between them.

A **leaf** in a tree is a vertex with exactly one edge adjacent to it.

Input

The first line contains t ($1 \le t \le 10^5$), the number of testcases. Each of the next t lines contains three space separated integers n, d and l, where:

- $2 \le l \le n \le 2 \cdot 10^5$
- $1 \le d < n$, and $2 \le l \le n$.
- The sum of n over all testcases doesn't exceed $2 \cdot 10^5$.

Output

For each testcase:

- If the tree with the required properties doesn't exist, print -1 on a new line.
- Else, print $n-1$ lines, containing the edges of a valid tree. The nodes of the tree should be numbered $1, 2, \ldots n$. You can output the edges in any order.

Example

standard input	standard output
2 3 2 3 4 2 3	-1 1 2 1 3 1 4

Note

In the first testcase, $n = 3, d = 2, l = 3$. There does not exist a tree with 3 nodes, all of which are leaves, so you should print -1.

In the second testcase, $n = 4, d = 2, l = 3$. It can be easily verified that the edges in the above output represent a tree with 4 vertices, out of which 3 are leaves, and has a diameter of 2.