Absolute Permutation

We define P to be a permutation of the first n natural numbers in the range $[1, n]$. Let $pos[i]$ denote the value at position i in permutation P using 1-based indexing.

P is considered to be an *absolute permutation* if $|pos[i] - i| = k$ holds true for every $i \in [1, n]$.

Given n and k, print the lexicographically smallest absolute permutation P. If no absolute permutation exists, print `-1`.

For example, let $n = 4$ giving us an array $pos = [1, 2, 3, 4]$. If we use 1 based indexing, create a permutation where every $|pos[i] - i| = k$. If $k = 2$, we could rearrange them to $[3, 4, 1, 2]$:

```
pos[i] i |Difference|
3 1 2
4 2 2
1 3 2
2 4 2
```

Function Description

Complete the *absolutePermutation* function in the editor below. It should return an integer that represents the smallest lexicographically smallest permutation, or -1 if there is none.

absolutePermutation has the following parameter(s):

- *n*: the upper bound of natural numbers to consider, inclusive
- *k*: the integer difference between each element and its index

Input Format

The first line contains an integer t, the number of test cases.
Each of the next t lines contains 2 space-separated integers, n and k.

Constraints

- $1 \leq t \leq 10$
- $1 \leq n \leq 10^5$
- $0 \leq k < n$

Output Format

On a new line for each test case, print the lexicographically smallest absolute permutation. If no absolute permutation exists, print `-1`.

Sample Input

```
3
2 1
3 0
3 2
```

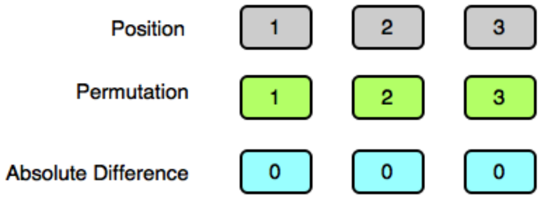
Sample Output

```
2 1
1 2 3
-1
```

Explanation

Test Case 0:

Test Case 1:



Test Case 2:
No absolute permutation exists, so we print `-1` on a new line.