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
```

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

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

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:

Author	ma5termind
Difficulty	Medium
Max Score	40
Submitted By	11557

NEED HELP?

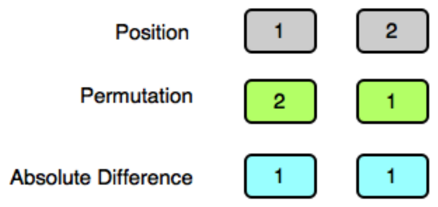
View discussions
View editorial
View top submissions

RATE THIS CHALLENGE

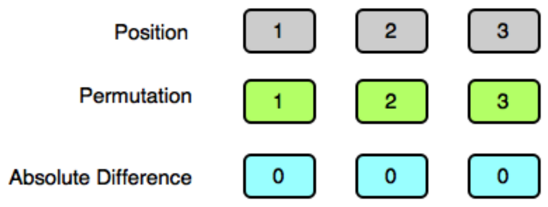
☆ ☆ ☆ ☆ ☆

MORE DETAILS

Download problem statement
Download sample test cases
Suggest Edits



Test Case 1:



Test Case 2:

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

Current Buffer (saved locally, editable)

Java 7

```java
import java.io.*;
import java.math.*;
import java.text.*;
import java.util.*;
import java.util.regex.*;

public class Solution {

    /*
     * Complete the absolutePermutation function below.
     */
    static int[] absolutePermutation(int n, int k) {
        /*
         * Write your code here.
         */

    }

    private static final Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) throws IOException {
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT_PATH")));

        int t = Integer.parseInt(scanner.nextLine().trim());

        for (int tItr = 0; tItr < t; tItr++) {
            String[] nk = scanner.nextLine().split(" ");

            int n = Integer.parseInt(nk[0].trim());

            int k = Integer.parseInt(nk[1].trim());

            int[] result = absolutePermutation(n, k);

            for (int resultItr = 0; resultItr < result.length; resultItr++) {
                bufferedWriter.write(String.valueOf(result[resultItr]));

                if (resultItr != result.length - 1) {
                    bufferedWriter.write(" ");
                }
            }

            bufferedWriter.newLine();
        }

        bufferedWriter.close();
    }
}

```

Line: 1 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature