E. Klever Permutation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given two integers n and k ($k \le n$), where k is even.

A permutation of length n is an array consisting of n distinct integers from 1 to n in any order. For example, $[2,3,1,5,4]$ is a permutation, but $[1,2,2]$ is not a permutation (as 2 appears twice in the array) and $[0,1,2]$ is also not a permutation (as $n=3$, but 3 is not present in the array).

Your task is to construct a k-*level* permutation of length n.

A permutation is called k-*level* if, among all the sums of continuous segments of length k (of which there are exactly $n-k+1$), any two sums differ by no more than 1.

More formally, to determine if the permutation p is k-*level*, first construct an array s of length $n-k+1$, where $s_i = \sum_{j=i}^{i+k-1} p_j$, i.e., the i-th element is equal to the sum of $p_i, p_{i+1}, \dots, p_{i+k-1}$.

A permutation is called k-*level* if $\max(s) - \min(s) \le 1$.

Find **any** k-*level* permutation of length n.

Input

The first line of the input contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. This is followed by the description of the test cases.

The first and only line of each test case contains two integers n and k ($2 \le k \le n \le 2 \cdot 10^5$, k is even), where n is the length of the desired permutation.

It is guaranteed that the sum of n for all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output **any** k-*level* permutation of length n.

It is guaranteed that such a permutation always exists given the constraints.

Example

input Copy

```
5
2 2
3 2
10 4
13 4
7 4
```

output Copy

```
2 1
1 3 2
1 8 4 10 2 7 5 9 3 6
```

```
4 10 1 13 5 9 2 12 6 8 3 11 7
1 6 3 7 2 5 4
```

Note

In the second test case of the example:

- $p_1 + p_2 = 3 + 1 = 4$;
- $p_2 + p_3 = 1 + 2 = 3$.

The maximum among the sums is 4, and the minimum is 3.