Sum of one-sequence

We say that a sequence of integers is a one-sequence if the difference between any two consecutive numbers in this sequence is 1 or -1 and its first element is 0. More precisely: [a_1, a_2, ..., a_n] is a one-sequence if

- for any k, such that $1 <= k < n$: $|a_k - a_{k+1}| = 1$ and
- $a_1 = 0$

Task

Write a program that for each test case:

- reads two integers describing the length of the sequence and the sum of its elements;
- finds a one-sequence of the given length whose elements sum up to the given value or states that such a sequence does not exist;
- writes the result to the standard output.

Input

The number of test cases *t* is in the first line of input, then *t* test cases follow separated by an empty line.

In the first line of a test case there is a number *n*, such that $1 <= n <= 10\,000$, which is the number of elements in the sequence. In the second line there is a number *S*, which is the sum of the elements of the sequence, such that $|S| <= 50\,000\,000$.

Output

For each test case there should be written *n* integers (each integer in a separate line) that are the elements of the sequence (*k*-th element in the *k*-th line) whose sum is *S* or the word "No" if such a sequence does not exist. If there is more than one solution your program should output any one.

Consequent test cases should by separated by an empty line.

Example

Input:
1
8
4

Output:
0
1
2
1
0
-1