A challenge in an Amazon Hackathon programming competition requires the construction of a sequence using a specified number of integers within a range. The sequence must be strictly increasing at first and then strictly decreasing. The goal is to maximize the sequence array elements starting from the beginning. For example, [4, 5, 4, 3, 2] beats [3,4,5,4,3] because its first element is larger, and [4, 5, 6, 5, 4] beats [4,5,4,3,2] because its third element is larger. Given the length of the sequence and the range of integers, return the winning sequence. If it is not possible to construct such a sequence, return -1.
Write an algorithm that returns a winning sequence and -1 if the sequence is not possible.

Input
The input to the function/method consists of three arguments: num, an integer representing the size of sequence to create; lowerEnd, an integer representing the lower end of integer range; upperEnd, an integer representing the upper end of integer range.

Output
Return a list of integers representing the winning sequence and if the sequence is not possible then return a list with an integer -1.

Constraints
$3 <= num <= 10^5$
$1 <= lowerEnds <= upperEnds <= 10^5$

Example
Input:
num = 5
lowerEnd = 3
upperEnd = 10

Output:
[9,10,9,8,7]

Explanation:
In this case, [9, 10, 9, 8, 7] is the winning sequence. It maintains the constraints of being first strictly increasing and then strictly decreasing, and there is no way to have integers in the sequence that are greater than [9, 10, 9, 8, 7]. So the output is [9, 10, 9, 8, 7].