Problem H. Highly Composite Permutations

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 mebibytes

Positive integer x is called *composite* if it has strictly more than two positive integer divisors. For example, integers 4, 30 and 111 are composite, while 1, 7 and 239 are not.

Integer sequence $p = \langle p_1, p_2, \ldots, p_n \rangle$ is called a *permutation of length* n if it contains every integer between 1 and n, inclusive, exactly once.

We'll call permutation $p = \langle p_1, p_2, \ldots, p_n \rangle$ *highly composite* if for every i between 1 and n, inclusive, the sum of the first i elements of p (that is, $p_1 + p_2 + \ldots + p_i$) is composite.

Given a single integer n, find a highly composite permutation of length n.

Input

The only line of the input contains a single integer n ($1 \le n \le 100$).

Output

If no highly composite permutation of length n exists, output a single integer -1. Otherwise, output n integers $p_1, p_2, \ldots, p_n$ such that $p = \langle p_1, p_2, \ldots, p_n \rangle$ is a highly composite permutation.

If there are multiple highly composite permutations of length n, you may output any of them.

Examples

standard input	standard output
13	9 13 6 5 3 2 8 4 1 12 11 10 7
2	-1

Note

In the first example test case, the first element of the permutation, 9, is composite, the sum of the first two elements of the permutation, $9 + 13 = 22$, is composite, the sum of the first three elements of the permutation, $9 + 13 + 6 = 28$, is composite, and so on.

In the second example test case, only two permutations of the required length exist, $\langle 1, 2 \rangle$ and $\langle 2, 1 \rangle$, and neither of them is highly composite.