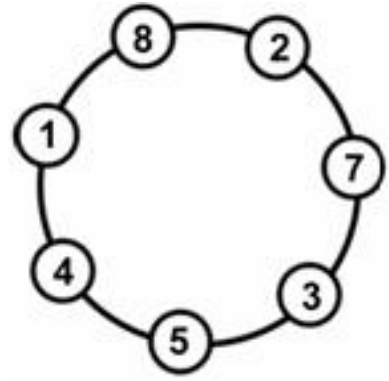
Suppose you want to own a roller coaster. Before you start, you might be interested in designing the course. The course is circular when seen from above, with n towers of equal distances on it. The figure below shows a course with $n = 7$ (numbers inside circles are heights of towers).



To make the towers look interesting, their heights should be distinct positive integers not greater than $n+1$. To let customers enjoy a large variety of excitement, the height differences between neighboring towers should be all different. Since there are n height differences, each integer value between 1 and n must appear exactly once. In the example above, the height differences are: 8-1=7, 8-2=6, 7-2=5, 7-3=4, 5-3=2, 5-4=1, 4-1=3. You can check that every integer between 1 and 7 appears exactly once.

Write a program to design the ride.

Input

The input consists of several test cases. Each case contains a single integer n ($2 \leq n \leq 1000$), the number of towers. The last test case is followed by a single zero, which should not be processed.

Output

For each test case, print the case number and n numbers if the design is possible, '`-1`' otherwise.

Sample Input

```
7
234
0
```

Sample Output

```
Case 1: 1 4 5 3 7 2 8
Case 2: -1
```