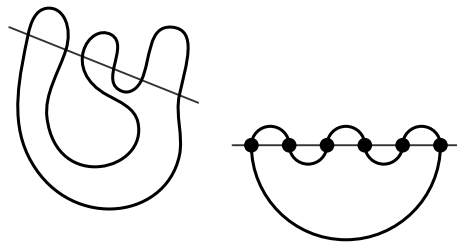


Problem F. k -th Meander

Input file: `kth-meander.in`
Output file: `kth-meander.out`
Time limit: 3 seconds
Memory limit: 256 mebibytes

In this problem, you have to find the lexicographically k -th meander of order n .

Consider a straight line drawn on a plane. A *meander* of order n is a closed planar curve which intersects the straight line $2n$ times. One can imagine the meander as a twisting road loop which intersects a straight segment of a river via bridges. The intuitive definition above must however be refined with a few formal conditions: the curve can have no self-intersections or self-touchings, all $2n$ points where the curve intersects the line are different and are indeed intersections, not touchings, and furthermore, the curve and the line have no common points except the above-mentioned ones.



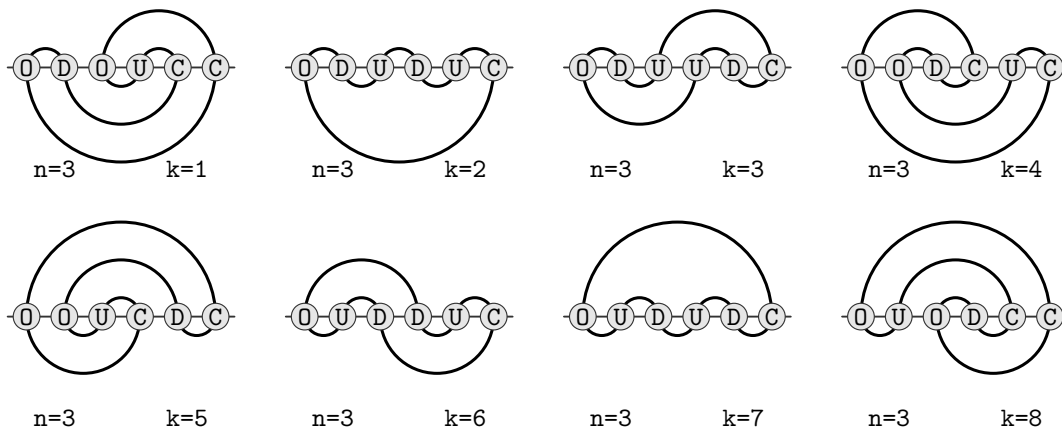
Let us define a convenient way to draw a meander. Without loss of generality, let the straight line be placed horizontally, and the intersection points put regularly with unit distances between them. Note that the intersection points divide the curve into $2n$ parts. Certain n parts are above the line, and the other n parts below it. We draw each of these parts as semicircles based on the corresponding segments of the line as diameters and located either above or below the line, as needed. One can prove that the picture we got is a meander: in particular, the curve constructed in this way has no self-intersections or self-touchings. The inverse is also true: every meander can be transformed to such picture by a continuous (homeomorphic) transformation.

Now, let us learn to write a meander down as a string. Consider a meander drawn according to the above procedure. Let us walk along the line from left to right and take a note every time we meet an intersection point. For each such point, there are two parts of the curve connected with it: one above and one below the line. Each of these parts is a semicircle going either to the left or to the right from the current intersection point. We will distinguish between four possible cases and label each case with an English letter:

- If both semicircles go to the right, we shall say that the curve *opens* at this intersection and denote it with the letter **O** (open).
- If both semicircles go to the left, we shall say that the curve *closes* at this intersection and denote it with the letter **C** (close).
- If the lower semicircle goes to the left and the upper one to the right, we shall say that the curve *goes up* at this intersection and denote it with the letter **U** (up).
- If the upper semicircle goes to the left and the lower one to the right, we shall say that the curve *goes down* at this intersection and denote it with the letter **D** (down).

Moving from left to right, write down the $2n$ letters which we used to label the intersections. It turns out that this information is enough to uniquely restore the drawing of a meander obtained by the procedure above. One can imagine this process as drifting with the current of the river (from left to right) and writing down the configuration of roads in the immediate vicinity of the bridges.

Two meanders are considered different if their notations are different. As an example, all eight different meanders of order 3 are presented below.



For a fixed order n , take all different meanders of this order, and then sort them lexicographically by their notations. Given the order n and the number k , find the notation of the lexicographically k -th meander of order n .

Input

The first line of input contains two integers n and k : the order and the number of a meander ($1 \leq n \leq 25$, $1 \leq k \leq 10^{11}$). It is guaranteed that a meander with the given parameters exists.

Output

Print a string consisting of $2n$ letters: the notation of the meander which has the number k in a lexicographically sorted list of meanders of order n .

Examples

| kth-meander.in | kth-meander.out |
|----------------|-----------------|
| 3 1 | ODOUCC |
| 3 2 | ODUDUC |
| 3 3 | ODUJDC |
| 3 4 | OODCUC |
| 3 5 | OOUCDC |
| 3 6 | OUDDUC |
| 3 7 | OUDJDC |
| 3 8 | OUODCC |

Explanations

The pictures above correspond to the examples.