E. Brackets

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

A two dimensional array is called a bracket array if each grid contains one of the two possible brackets — "(" or ")".
A path through the two dimensional array cells is called monotonous if any two consecutive cells in the path are side-adjacent and each cell of the path is located below or to the right from the previous one.

A two dimensional array whose size equals $n \times m$ is called a correct bracket array, if any string formed by writing out the brackets on some monotonous way from cell $(1, 1)$ to cell (n, m) forms a correct bracket sequence.

Let's define the operation of comparing two correct bracket arrays of equal size (a and b) like that. Let's consider a given two dimensional array of priorities (c) — a two dimensional array of same size, containing different integers from 1 to nm. Let's find such position (i, j) in the two dimensional array, that $a_{i,j} \neq b_{i,j}$. If there are several such positions, let's choose the one where number $c_{i,j}$ is minimum. If $a_{i,j} =$ "(", then $a < b$, otherwise $a > b$. If the position (i, j) is not found, then the arrays are considered equal.

Your task is to find a k-th two dimensional correct bracket array. It is guaranteed that for the given sizes of n and m there will be no less than k two dimensional correct bracket arrays.

Input

The first line contains integers n, m and k — the sizes of the array and the number of the sought correct bracket array ($1 \leq n, m \leq 100$, $1 \leq k \leq 10^{18}$). Then an array of priorities is given, n lines each containing m numbers, number $p_{i,j}$ shows the priority of character j in line i ($1 \leq p_{i,j} \leq nm$, all $p_{i,j}$ are different).

Please do not use the %lld specificator to read or write 64-bit integers in C++. It is preferred to use the cin, cout streams or the %I64d specificator.

Output

Print the k-th two dimensional correct bracket array.

Examples

input
1 2 1 1 2

output
()

input
2 3 1 1 2 3 4 5 6

output
(() ())

input
3 2 2 3 6 1 4 2 5

output
())(()

Note

In the first sample exists only one correct two-dimensional bracket array.

In the second and in the third samples two arrays exist.

A bracket sequence is called regular if it is possible to obtain correct arithmetic expression by inserting characters «+» and «1» into this sequence. For example, sequences «(()())()», «()» and «(()(()))» are regular, while «)(», «(()» and «(()))(» are not.