Write a program to print all the permutations of \{1, \ldots, n\} with exactly \(k\) cycles, where \(1 \leq k \leq n\). For example, consider the permutation \((4, 3, 2, 5, 1, 7, 6)\). At position 1 there is a 4, at position 4 there is a 5, and at position 5 there is a 1. Therefore, one of the cycles is \(1 \rightarrow 4 \rightarrow 5 \rightarrow 1\). The other two cycles are \(2 \rightarrow 3 \rightarrow 2\) and \(6 \rightarrow 7 \rightarrow 6\). The permutation \((3, 2, 1)\) has the two cycles \(1 \rightarrow 3 \rightarrow 1\) and \(2 \rightarrow 2\), and the permutation \((3, 4, 5, 6, 7, 1, 2)\) only has the cycle \(1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 2 \rightarrow 4 \rightarrow 6 \rightarrow 1\).

**Input**

Input consists of \(n\) and \(k\), with \(1 \leq k \leq n\).

**Output**

Print all the permutations of \{1, \ldots, n\} with \(k\) cycles.

**Information about the checker**

You can print the solutions to this exercise in any order.

**Hint**

A possible program does not build the permutations consecutively from left to right, but jumping over the solution, using a function

\[
    \text{void } f(\text{int } i, \text{ int } ini, \text{ int } cells, \text{ int } cycles);
\]

where \(i\) is the next cell to fill, \(ini\) is where the current cycle—still to be closed—starts, \(cells\) is the number of cells still free, and \(cycles\) is the number of cycles yet to be created.

**Sample input 1**

\[
\begin{align*}
\text{3} & 1 \\
\end{align*}
\]

**Sample output 1**

\[
\begin{align*}
\text{(2, 3, 1)} \\
\text{(3, 1, 2)} \\
\end{align*}
\]

**Sample input 2**

\[
\begin{align*}
\text{3} & 2 \\
\end{align*}
\]

**Sample output 2**

\[
\begin{align*}
\text{(2, 1, 3)} \\
\text{(1, 3, 2)} \\
\text{(3, 2, 1)} \\
\end{align*}
\]

**Sample input 3**

\[
\begin{align*}
\text{3} & 3 \\
\end{align*}
\]

**Sample output 3**

\[
\begin{align*}
\text{(1, 2, 3)} \\
\end{align*}
\]