

C. Arrangement

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

In the year 2500 the annual graduation ceremony in the German University in Cairo (GUC) has run smoothly for almost 500 years so far.

The most important part of the ceremony is related to the arrangement of the professors in the ceremonial hall.

Traditionally GUC has n professors. Each professor has his seniority level. All seniorities are different. Let's enumerate the professors from 1 to n , with 1 being the most senior professor and n being the most junior professor.

The ceremonial hall has n seats, one seat for each professor. Some places in this hall are meant for more senior professors than the others. More specifically, m pairs of seats are in "senior-junior" relation, and the tradition requires that for all m pairs of seats (a_i, b_i) the professor seated in "senior" position a_i should be more senior than the professor seated in "junior" position b_i .

GUC is very strict about its traditions, which have been carefully observed starting from year 2001. The tradition requires that:

- The seating of the professors changes every year.
- Year 2001 ceremony was using lexicographically first arrangement of professors in the ceremonial hall.
- Each consecutive year lexicographically next arrangement of the professors is used.

The arrangement of the professors is the list of n integers, where the first integer is the seniority of the professor seated in position number one, the second integer is the seniority of the professor seated in position number two, etc.

Given n , the number of professors, y , the current year and m pairs of restrictions, output the arrangement of the professors for this year.

Input

The first line contains three integers n , y and m ($1 \leq n \leq 16$, $2001 \leq y \leq 10^{18}$, $0 \leq m \leq 100$) — the number of professors, the year for which the arrangement should be computed, and the number of pairs of seats for which the seniority relation should be kept, respectively.

The next m lines contain one pair of integers each, " $a_i b_i$ ", indicating that professor on the a_i -th seat is more senior than professor on the b_i -th seat ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$). Some pair may be listed more than once.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin` stream (you may also use the `%I64d` specifier).

Output

Print the order in which the professors should be seated in the requested year.

If by this year the GUC would have ran out of arrangements, or the given "senior-junior" relation are contradictory, print "The times have changed" (without quotes).

Examples

input
3 2001 2 1 2 2 3
output
1 2 3

input
7 2020 6 1 2 1 3 2 4 2 5 3 6 3 7
output
1 2 3 7 4 6 5

input
10 3630801 0
output
The times have changed

input
3 2001 3 1 2 2 3 3 1
output
The times have changed

Note

In the first example the lexicographically first order of seating is 1 2 3.

In the third example the GUC will run out of arrangements after the year 3630800.

In the fourth example there are no valid arrangements for the seating.

The lexicographical comparison of arrangements is performed by the < operator in modern programming languages. The arrangement a is lexicographically less that the arrangement b , if there exists such i ($1 \leq i \leq n$), that $a_i < b_i$, and for any j ($1 \leq j < i$) $a_j = b_j$.