

Or Machine

Input file: **standard input**
Output file: **standard output**
Time limit: 4 seconds
Memory limit: 1024 megabytes

We are developing the Or Machine, a computer heavily optimized solely for one kind of operation: the |= operator in C++'s term.

The Or Machine has n registers, each containing a nonnegative integer less than 2^8 . We label them x_1, x_2, \dots, x_n . A program is represented by a list of l operations. Each operation is represented by a pair of integers (a, b) , meaning that the machine should update x_a with the bitwise OR of x_a 's and x_b 's values.

The Or Machine takes a program, the initial values of the registers, and a positive integer t . When run, the program performs each operation in the program one by one. When the last operation is performed, it goes back to the first operation and repeats the process. The machine stops after performing exactly t operations.

We want our machine to be much faster than general-purpose computers, and hardware optimization is probably not enough. Can you help us with some software optimization?

Input

The first line contains three integers, n , l , and t ($1 \leq n, l \leq 2^{18}$, $1 \leq t \leq 10^{18}$). l is the length of the program.

The program is given on the next l lines. Each line contains two integers a and b ($1 \leq a, b \leq n$) representing the pair of registers that participate in the given operation.

The final line contains n integers, the initial values of the registers x_1, \dots, x_n ($0 \leq x_i < 2^8$).

Output

Output n integers on a single line, the values of the registers x_1, \dots, x_n after t operations.

Example

standard input	standard output
5 4 5	15 7 5 3 10
1 2	
2 3	
2 4	
4 4	
8 0 5 3 10	