Problem H. Elimination Series Once More

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

2^n contestants take part in a programming contest. They are labeled from 1 to 2^n, and the contestant labeled i has a level of a_i. It is guaranteed that $a_1, a_2, \ldots, a_{2^n}$ forms a permutation of length 2^n (i.e., each a_i is an integer in the range $[1, 2^n]$, and $a_1, a_2, \ldots, a_{2^n}$ are pairwise distinct).

The contest takes the form of an elimination series. There are n rounds; each round will knock out exactly half of the contestants. In each round, let m be the current number of contestants, then the contestant with the smallest label (i.e., the label of each contestant remains unchanged throughout the contest) competes with the contestant with the second smallest label, the contestant with the third smallest label competes with the contestant with the fourth smallest label, and so on. In each competition, the contestant with a lower level is eliminated, while the contestant with a higher level will advance to the next round.

Little D is the contestant labeled x. He wants to use magic **before the contest** to win as many competitions as possible. Specifically, he can perform the following operation no more than k times: choose two contestants **other than himself** and swap their levels. Note that the operation can only be done **before the contest**.

For each $x = 1, 2, \ldots, 2^n$, print the maximum number of competitions he can win.

Input

The first line contains two integers n ($1 \le n \le 20$) and k ($0 \le k < 2^n$).

The second line contains 2^n integers $a_1, a_2, \ldots, a_{2^n}$. It is guaranteed that a is a permutation.

Output

For each $x = 1, 2, \ldots, 2^n$, print the maximum number of competitions Little D can win.

Example

standard input	standard output
2 1 1 2 3 4	0 1 1 2