Problem C
Sister Cities
Time limit: 1 second

The ACM kingdom has n cities, numbered 0, 1, ..., $n-1$, where n is even. No cities cross the equatorial. So each city is either in the northern or southern hemisphere, but not both.

The ACM Queen wants to form sister cities. So she asks each city c to propose a list ℓ_c of exactly 2^k cities, with which c is willing to have sister city relationship. Willingness is known to be mutual: If a city d is in ℓ_c, then c is in ℓ_d. Of course, k is a non-negative integer specified by the ACM Queen. To facilitate communication between the northern and southern hemispheres, ℓ_c has to contain only cities not in the hemisphere containing c, for each city c. I.e., if c is in the northern (southern) hemisphere, then all cities in ℓ_c must be in the southern (northern, respectively) hemisphere.

Please help the ACM Queen order all cities as c_0, c_1, ..., c_{n-1} such that for each even number $0 \le i \le n-2$, c_i and c_{i+1} are willing to be sister cities (i.e., c_i is in $\ell_{c_{i+1}}$). Sensibly, the ordering is intended to make c_i and c_{i+1} sister cities for each even number $0 \le i \le n-2$. Note that c_0, c_1, ..., c_{n-1} should be a permutation of 0, 1, ..., $n-1$.

Should two or more solutions exist, please just output one of them.

Input File Format

The first line is n, where $n \le 100000$. The second line is $n \cdot 2^{k-1}$, where $k \le 16$. For each pair (c, d) of cities such that c and d are willing to be sister cities (i.e., c is in ℓ_d), there is exactly one remaining line giving c and d (or, in the opposite order, d and c). Two numbers in a line are separated by space(s).

Output Format

Order all cities as c_0, c_1, ..., c_{n-1} such that for each even number $0 \le i \le n-2$, c_i and c_{i+1} are willing to be sister cities. Then output c_i and c_{i+1} in one line, for each even number $0 \le i \le n-2$. So there should be $n/2$ lines of output. Note that there may be many correct outputs.

Sample Input

```
10
20
9 0
9 2
9 8
9 3
1 2
1 8
3 1
4 1
8 6
6 3
6 4
0 6
5 3
5 4
5 0
5 2
7 4
0 7
2 7
8 7
```

Output for the Sample Input

```
0 7
1 4
2 5
3 9
6 8
```