A. Autocomplete

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Autocomplete is a program function that enables inputting the text (in editors, command line shells, browsers etc.) completing the text by its inputted part. Vasya is busy working on a new browser called 'BERowser'. He happens to be working on the autocomplete function in the address line at this very moment. A list consisting of n last visited by the user pages and the inputted part s are known. Your task is to complete s to make it an address of one of the pages from the list. You have to find the lexicographically smallest address having a prefix s.

Input

The first line contains the s line which is the inputted part. The second line contains an integer n $(1 \le n \le 100)$ which is the number of visited pages. Then follow n lines which are the visited pages, one on each line. All the lines have lengths of from 1 to 100 symbols inclusively and consist of lowercase Latin letters only.

Output

If s is not the beginning of any of n addresses of the visited pages, print s. Otherwise, print the lexicographically minimal address of one of the visited pages starting from s.

The lexicographical order is the order of words in a dictionary. The lexicographical comparison of lines is realized by the `'<'` operator in the modern programming languages.

Sample test(s)

input

```
next
2
nextpermutation
nextelement
```

output

```
nextelement
```

input

```
find
4
find
findfirstof
findit
fand
```

output

```
find
```

input

```
find
4
fondfind
fondfirstof
fondit
fand
```

output

```
find
```

B. Blog Photo

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One popular blog site edits the uploaded photos like this. It cuts a rectangular area out of them so that the ratio of height to width (i.e. the $height / width$ quotient) can vary from 0.8 to 1.25 inclusively. Besides, at least one side of the cut area should have a size, equal to some power of number 2 (2^x for some integer x). If those rules don't indicate the size of the cut are clearly, then the way with which the cut part possesses the largest area is chosen. Of course, both sides of the cut area should be integer. If there are several answers to this problem, you should choose the answer with the maximal height.

Input

The first line contains a pair of integers h and w ($1 \leq h, w \leq 10^9$) which are the height and width of the uploaded photo in pixels.

Output

Print two integers which are the height and width of the cut area.

Sample test(s)

input

```
2 1
```

output

```
1 1
```

input

```
2 2
```

output

```
2 2
```

input

```
5 5
```

output

```
5 4
```

C. Little Frog

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Once upon a time a little frog whose name was Vasya decided to travel around his home swamp. Overall there are n mounds on the swamp, located on one line. The distance between the neighboring mounds is one meter. Vasya wants to visit all the mounds in one day; besides, he wants to visit each one exactly once. For that he makes a route plan, to decide the order in which to jump on the mounds. Vasya can pick any mound as the first one. He thinks it boring to jump two times at the same distance. That's why he wants any two jumps on his route to have different lengths. Help Vasya the Frog and make the plan for him.

Input

The single line contains a number n $(1 \le n \le 10^4)$ which is the number of mounds.

Output

Print n integers p_i $(1 \le p_i \le n)$ which are the frog's route plan.

- All the p_i's should be mutually different.
- All the $|p_i – p_{i+1}|$'s should be mutually different $(1 \le i \le n - 1)$.

If there are several solutions, output any.

Sample test(s)

input

```
2
```

output

```
1 2
```

input

```
3
```

output

```
1 3 2
```

D. Physical Education

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya is a school PE teacher. Unlike other PE teachers, Vasya doesn't like it when the students stand in line according to their height. Instead, he demands that the children stand in the following order: $a_1, a_2, ..., a_n$, where a_i is the height of the i-th student in the line and n is the number of students in the line. The children find it hard to keep in mind this strange arrangement, and today they formed the line in the following order: $b_1, b_2, ..., b_n$, which upset Vasya immensely. Now Vasya wants to rearrange the children so that the resulting order is like this: $a_1, a_2, ..., a_n$. During each move Vasya can swap two people who stand next to each other in the line. Help Vasya, find the sequence of swaps leading to the arrangement Vasya needs. It is not required to minimize the number of moves.

Input

The first line contains an integer n ($1 \leq n \leq 300$) which is the number of students. The second line contains n space-separated integers a_i ($1 \leq a_i \leq 10^9$) which represent the height of the student occupying the i-th place must possess. The third line contains n space-separated integers b_i ($1 \leq b_i \leq 10^9$) which represent the height of the student occupying the i-th place in the initial arrangement. It is possible that some students possess similar heights. It is guaranteed that it is possible to arrange the children in the required order, i.e. a and b coincide as multisets.

Output

In the first line print an integer k ($0 \leq k \leq 10^6$) which is the number of moves. It is not required to minimize k but it must not exceed 10^6. Then print k lines each containing two space-separated integers. Line p_i, $p_i + 1$ ($1 \leq p_i \leq n - 1$) means that Vasya should swap students occupying places p_i and p_{i+1}.

Sample test(s)

input

```
4
1 2 3 2
3 2 1 2
```

output

```
4
2 3
1 2
3 4
2 3
```

input

```
2
1 100500
1 100500
```

output

```
0
```

E. Dead Ends

time limit per test: 5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Life in Bertown has become hard. The city has too many roads and the government spends too much to maintain them. There are n junctions and m two way roads, at which one can get from each junction to any other one. The mayor wants to close some roads so that the number of roads left totaled to $n - 1$ roads and it were still possible to get from each junction to any other one. Besides, the mayor is concerned with the number of dead ends which are the junctions from which only one road goes. There shouldn't be too many or too few junctions. Having discussed the problem, the mayor and his assistants decided that after the roads are closed, the road map should contain exactly k dead ends. Your task is to count the number of different ways of closing the roads at which the following conditions are met:

- There are exactly $n - 1$ roads left.
- It is possible to get from each junction to any other one.
- There are exactly k dead ends on the resulting map.

Two ways are considered different if there is a road that is closed in the first way, and is open in the second one.

Input

The first line contains three integers n, m and k $(3 \le n \le 10, n - 1 \le m \le n \cdot (n - 1) / 2, 2 \le k \le n - 1)$ which represent the number of junctions, roads and dead ends correspondingly. Then follow m lines each containing two different integers v_1 and v_2 $(1 \le v_1, v_2 \le n, v_1 \ne v_2)$ which represent the number of junctions connected by another road. There can be no more than one road between every pair of junctions. The junctions are numbered with integers from 1 to n. It is guaranteed that it is possible to get from each junction to any other one along the original roads.

Output

Print a single number — the required number of ways.

Sample test(s)

input

```
3 3 2
1 2
2 3
1 3
```

output

```
3
```

input

```
4 6 2
1 2
2 3
3 4
4 1
1 3
2 4
```

output

```
12
```

input

```
4 6 3
1 2
2 3
3 4
4 1
1 3
2 4
```

output

```
4
```