A. Factorial

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

Print the factorial of the given integer number n. The factorial of n is equal to $1 \cdot 2 \cdot \ldots \cdot n$.

Input

The only line contains n ($1 \leq n \leq 10$).

Output

Print the factorial of n.

Examples

input

```
3
```

output

```
6
```

input

```
5
```

output

```
120
```

B. Expression

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

You are given a line «`n#m`», where «`n`» and «`m`» are digits, and «`#`» is a sign «`+`» or «`-`». Print the value of the given expression.

Input

The only given line has a string in form «`n#m`», where «`n`» and «`m`» are digits (from 0 to 9), and «`#`» is a sign «`+`» or «`-`».

Output

Print the value of the given expression.

Examples

input

```
1-5
```

output

```
-4
```

C. Table

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

The integer numbers from 1 to nm was put into rectangular table having n rows and m columns. The numbers was put from left to right, from top to bottom, i.e. the first row contains integers $1, 2, \ldots, m$, the second — $m+1, m+2, \ldots, 2*m$ and so on.

After it these numbers was written on the paper in another order: from top to bottom, from left to right. First, the numbers in the first column was written (from top to bottom) and so on.

Print the k-th number on the paper.

Input

The only line in the input contains three integer numbers n, m and k ($1 \le n, m \le 20000$, $1 \le k \le nm$).

Output

Print the required number.

Examples

input

```
3 4 11
```

output

```
8
```

input

```
20000 10000 200000000
```

output

```
200000000
```

D. Presents

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

You are given the prices of three presents. Also there are three sisters. It is known that the most valuable present is for the eldest sister. The second (by price) is for the second sister. And the less valuable present is for the youngest sister. If two (or three) presents have the same price, corresponding sisters may get the in a random way.

Input

The only line contains three integer numbers a_1, a_2, a_3 ($1 \le a_1, a_2, a_3 \le 100$) — the prices of the presents.

Output

Print three numbers i_1, i_2, i_3 ($1 \le i_1, i_2, i_3 \le 3$), all of them should be distinct. The first number stands for the seniority of the sister which will get the first present (1 stands for the eldest, 3 for the youngest). The second and third numbers mean the seniority of the sisters which get the second and the third present respectively.

If there are multiple answers, print any of them.

Examples

input

```
11 13 1
```

output

```
2 1 3
```

input

```
30 10 30
```

output

```
1 3 2
```

Note

In the second sample another possible answer is "2 3 1".

E. Prime Segment

time limit per test: 2 seconds

memory limit per test: 64 megabytes

input: standard input

output: standard output

Positive integer number x is called *prime*, if it has exactly two positive integer divisors. For example, 2, 3, 17, 97 are primes, but 1, 10, 120 are not.

You are given an integer number n, find the shortest segment $[a, b]$, which contains n (i.e. $a \le n \le b$) and a, b are primes.

Input

The only given line contains an integer number n ($2 \le n \le 10000$).

Output

Print the space separated pair of the required numbers a, b.

Examples

input

```
10
```

output

```
7 11
```

input

```
97
```

output

```
97 97
```

F. Domain

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

This problem doesn't contain real-world specifications about domains, just use the problem statement as a formal document to solve the problem.

The given string S is a domain name if it contains the characters "a"-"z", "0"-"9" and dots. No two dots shoud follow one after another (consecutive). The dots split the given string S into the parts, the last (most right) part should have the length 2 or 3. Domain can't start or end with a dot.

You are given the string S, check if it is domain name.

Input

The only line of the input contains given string S. The string may contain any characters with ASCII codes from 33 to 127, inclusive. The string length is between 1 and 100, inclusive.

Output

Print "YES" if the given string S is a domain name, or print "NO" if it is not.

Examples

input

```
codeforces.com
```

output

```
YES
```

input

```
mail.v-kontakte.ru
```

output

```
NO
```

G. Path Canonization

time limit per test: 2 seconds

memory limit per test: 64 megabytes

input: standard input

output: standard output

A path in some Unix-similar file system is given. The path consists of elements separated with characters "`/`". For example: "`/usr/share/mysql/../tomcat6/conf/server.xml`". The path starts with the root directory (i.e. starts with the character "`/`"). Each element means a name of file or directory, or it is one of two special elements: "`.`" или "`..`". First of them stands for the current directory (for example, path "`/./usr/././share`" is equal to "`/usr/share`"). The second element "`..`" stands for the moving to the parent directory (for example, path "`/usr/share/../lib`" is equal to "`/usr/lib`").

You task is to convert the given path to such a path, which doesn't contain special elements "`.`" and/or "`..`". If it is impossible, print "`-1`". The only reason for it is an attempt to move to the parent directory from the root.

Input

The only line contains the given path. The path starts with "`/`" and consists of elements separated with "`/`". No two "`/`" follow one after another (consecutively). The only path which can end with "`/`" is the root directory path equal to "`/`".

Each element may contain "`a`"-"`z`", "`0`"-"`9`" and dots. Any element different from specials "`.`" and "`..`" contains at least one character different from the dots.

The path length is between 1 and 1000 inclusively.

Output

Print the required path or "`-1`".

Examples

input

```
/usr/share/mysql/../tomcat6/conf/server.xml
```

output

```
/usr/share/tomcat6/conf/server.xml
```

input

```
/a/./././..
```

output

```
/
```

H. Table Bowling

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

Table bowling tournament participant completed the competition according to the given final standings table. The table is given as a sequence of lines, each line has a format "name score". Your task is to prepare another table consisting of lines in the form "place name". Sort participant by score (desc.) and by the name lexicographically in the case of a tie. Places are numerated from 1. If more than one participant has some score, all of them share the places and you should output something like "12-14 john".

Please, look into the samples for clarification.

Input

The first line contains n ($1 \leq n \leq 100$) — the number of rows in the table. Following n lines contain the given table. Each line has the form "name score", where "name" is a sequence of lowercase Latin letters, and "score" — is an integer number between 0 and 1000, inclusive. All the names are distinct. The length of each name is between 1 and 10 characters, inclusive. There is single space between the name and the score in each line.

Output

Print the required table. Look at the sample outputs for clarifications.

Examples

input

```
5
vasya 10
ted 11
petya 10
katya 33
mike 44
```

output

```
1 mike
2 katya
3 ted
4-5 petya
4-5 vasya
```

input

```
3
a 1
b 13
c 1
```

output

```
1 b
2-3 a
2-3 c
```

I. Sort the Table

time limit per test: 2 seconds

memory limit per test: 64 megabytes

input: standard input

output: standard output

You are given a rectangular table containing words. Each of its columns has its own name. You are also given the list of rules of sorting in form "`FIELD_NAME SORT_ORDER`", where `SORT_ORDER` is either `ASC` (nondescending order) or `DESC` (nonascending order). Rules in the list are separated by a single comma with a single space. You have to sort rows of the table primarily by the first rule, then, in case of a tie sort by the second rule. And so on. If two rows are equal in terms of every rule, then preserve their relative order. You can assume that each element of the table has type "string", so you have to use lexicographic comparison.

Input

The first line contains column names. The second line contains the list of rules. The rest of the input data contains the table. All the words and column names are separated by single spaces. The number of rows and columns is between 1 and 100, inclusive. Names of columns and elements are strings containing only uppercase and lowercase Latin letters and digits, having the length between 1 and 10, inclusive.

Output

Print the table after the sorting.

Examples

input

```
NAME GROUP AGE
GROUP ASC, AGE DESC
Alex 412 19
Peter 422 19
Sergey 412 18
Andrey 311 18
```

output

```
Andrey 311 18
Alex 412 19
Sergey 412 18
Peter 422 19
```