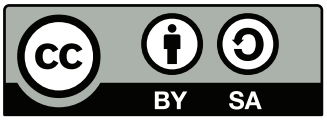
Freshmen Programming Contest

Contest Problem Set

May 2, 2018



A - Alphabetic Shift

You are trying to learn for an exam, but you notice that you do not have enough time to learn everything, completely in panic you decide to cheat on the test. You went into the office of your professor and took the exam. But once you got it in your hands, you notice you cannot read it easily. Your initial guess it that it is encoded with a Ceasar cipher. That is a cipher that shifts every letter a certain amount to the right, starting again at 'A' once you go past 'Z'.

For example: A ceasar cipher of 10 will shift every letter 10 alphabetic places to the right:

'ABCDEFGHIJKLMNOPQRSTUVWXYZ' will yield:

'LMNOPQRSTUVWXYZABCDEFGHIJK'

If you encode 'LOREM IPSUM DOLOR' with a Ceasar cipher of 10, you will end up with 'VYBOW SZCEW NYVYB'.

So if you decode "VYBOW SZCEW NYVYB" with Ceasar cipher 10, you go back 10 places to the left. resulting back in 'LOREM IPSUM DOLOR'.

However, as you are not sure how many places your professor shifted the alphabet, you want to write a program that takes as input the encoded string, and gives as output all 26 possible Ceasar decoded strings in order. So you can then manually determine which may be the actual solution.

Input

One line containing a single string (which may have spaces) which is the question and the answer. You may assume the string only contains uppercase A-Z. The length of this string is at most 1000.

Output

26 lines with on each line one decoded string. In order, starting with 0, up to a shift of 25.

Examples

In sample 1 you will see that the text has been encoded with a shift of 25. So a shift of 25 to the left (the bottom line) is the correctly decoded string. Do note that you do not have to find the correct decoding. Just give all 26 of them, ordered from 0 to 25.

input 1	**output 1**
VGZSHRSGDRLZKKDRSOQHLDSVN	VGZSHRSGDRLZKKDRSOQHLDSVN UFYRGQRFCQKYJJCQRNPGKCRUM TEXQFPQEBPJXIIBPQMOFJBQTL SDWPEOPDAOIWHHAOPLNEIAPSK RCVODNOCZNHVGGZNOKMDHZORJ QBUNCMNBYMGUFFYMNJLCGYNQI PATMBLMAXLFTEEXLMIKBFXMPH OZSLAKLZWKESDDWKLHJAEWLOG NYRKZJKYVJDRCCVJKGIZDVKNF MXQJYIJXUICQBBUIJFHYCUJME LWPIXHIWTHBPAATHIEGXBTILD KVOHWGHVSGAOZZSGHDFWASHKC JUNGVFGURFZNYYRFGCEVZRGJB ITMFUEFTQEYMXXQEFBDUYQFIA HSLETDESPDXLWWPDEACTXPEHZ GRKDSCDROCWKVVOCDZBSWODGY FQJCRBCQNBVJUUNBCYARVNCFX EPIBQABPMAUITTMABXZQUMBEW DOHAPZAOLZTHSSLZAWYPTLADV CNGZOYZNKYSGRRKYZVXOSKZCU BMFYNXYMJXRFQQJXYUWNRJYBT ALEXMWXLIWQEPPIWXTVMQIXAS ZKDWLVWKHVPDOOHVWSULPHWZR YJCVKUVJGUOCNNGUVRTKOGVYQ XIBUJTUIFTNBMMFTUQSJNFUXP WHATISTHESMALLESTPRIMETWO

In sample 2 you will see that the text has been encoded with a shift of 4. So the 5th line will yield the correctly decoded string.

input 2	**output 2**
ALEXMWXLIPSRKIWXVMZIVSRIEVXLEQEDSR	ALEXMWXLIPSRKIWXVMZIVSRIEVXLEQEDSR ZKDWLVWKHORQJHVWULYHURQHDUWKDPDCRQ YJCVKUVJGNQPIGUVTKXGTQPGCTVJCOCBQP XIBUJTUIFMPOHFTUSJWFSPOFBSUIBNBAPO WHATISTHELONGESTRIVERONEARTHAMAZON VGZSHRSGDKNMFDRSQHUDQNMDZQSGZLZYNM UFYRGQRFCJMLECQRPGTCPMLCYPRFYKYXML TEXQFPQEBILKDBPQOFSBOLKBXOQEXJXWLK SDWPEOPDAHKJCAOPNERANKJAWNPDWIWVKJ RCVODNOCZGJIBZNOMDQZMJIZVMOCVHVUJI QBUNCMNBYFIHAYMNLCPYLIHYULNBUGUTIH PATMBLMAXEHGZXLMKBOXKHGXTKMATFTSHG OZSLAKLZWDGFYWKLJANWJGFWSJLZSESRGF NYRKZJKYVCFEXVJKIZMVIFEVRIKYRDRQFE MXQJYIJXUBEDWUIJHYLUHEDUQHJXQCQPED LWPIXHIWTADCVTHIGXKTGDCTPGIWPBPODC KVOHWGHVSZCBUSGHFWJSFCBSOFHVOAONCB JUNGVFGURYBATRFGEVIREBARNEGUNZNMBA ITMFUEFTQXAZSQEFDUHQDAZQMDFTMYMLAZ HSLETDESPWZYRPDECTGPCZYPLCESLXLKZY GRKDSCDROVYXQOCDBSFOBYXOKBDRKWKJYX FQJCRBCQNUXWPNBCARENAXWNJACQJVJIXW EPIBQABPMTWVOMABZQDMZWVMIZBPIUIHWV DOHAPZAOLSVUNLZAYPCLYVULHYAOHTHGVU CNGZOYZNKRUTMKYZXOBKXUTKGXZNGSGFUT BMFYNXYMJQTSLJXYWNAJWTSJFWYMFRFETS

input 3	**output 3**
AAA	AAA ZZZ YYY XXX WWW VVV UUU TTT SSS RRR QQQ PPP OOO NNN MMM LLL KKK JJJ III HHH GGG FFF EEE DDD CCC BBB

B - BINGO!

A common activity for students during a lecture is playing BINGO. However, they don't play the regular boring game with the numbered balls. Instead, their BINGO sheets contain in every square an event that might (or might not) happen during a lecture, and once the event happens, this square can be crossed off.

Since you'd also like to pay attention during the lecture while playing this game, you decide to build a program that automatically plays BINGO for you. Given a BINGO sheet with a grid of $n \times n$ squares, and a list of events that happen during the lecture, how many events will pass before you can shout "BINGO!"?

Note that you may shout "BINGO!" whenever you crossed off all squares in any row, column, or diagonal of the grid.
The middle square in your grid can always be crossed off for free.

Input

One line containing two integers: one odd integer n, with $1 \leq n \leq 10^3$, and one integer m, with $0 \leq m \leq 10^6$.
n lines, each containing n space-separated events. Event names consist of only alphanumeric characters, and every event only happens at most once.
m lines, with on every line an event that happens during the lecture.

Output

One integer, indicating after how many events you can shout "BINGO!".
If you cannot shout "BINGO!" during this lecture, output a sad smiley face: `:-(`

Examples

input 1	output 1
3 10 WordMispronounced LoudMic Gaming Sleeping FreeLunch Latecomer 2MinutesSilence TeacherAngry Eating MicBreaks Sleeping SlideshowContainsMistake WordMispronounced Gaming PhoneRings Eating Latecomer TeacherAngry 2MinutesSilence	7

input 2	output 2
3 5 EventA EventB EventC EventD EventE EventF EventG EventH EventI EventJ EventA EventK EventB EventD	:-(

C - Cryptography

Dave has just completed the Massive Open Online Course (MOOC) Cryptography on the popular website Coursera.org.
Eager to create his own cryptography system – against the advise of the teacher Dan Boneh to never, ever, ever implement your own crypto-system – he searches for a SKP (Special Key Prime).
A SKP is a prime that is preferably a large number, because the larger the number the more secure it is to use as a key.

Remember that a prime is a number that is only divisible by 1 and itself. For example 2 is a prime because it's only divisible by 1 and 2. 15 however is not a prime since beside 1 and 15, also 3 and 5 happen to divide this number. The number 1 is considered to not be a prime.

Luckily his friend Trudy is quite good at guessing large numbers that could be prime. Your task is given a number by Trudy, to decide whether this is actually a prime or not.

Input

You are given a number $0 \leq n \leq 10^{10}$, the number that Trudy has guessed for Dave to use as a SKP.

Output

You should output "SAFE" (without the quotes) iff the number n is a prime, else your program should output "BROKEN" (again, without the quotes).

Examples

input 1	output 1
2	SAFE

input 2	output 2
15	BROKEN

input 3	output 3
104729	SAFE

D - Doner Time!

Every student knows the practice: after having had one too many drinks in the \PUB, you will go to the Doner Shop to treat yourself with some comfort food.

But experience tells you that you and your friends will not recall the way to the Doner Shop when you are drunk. Besides there are a lot of Doner Shops in the city. Therefore you decide to write a program that will tell you how far away the nearest Doner store is from your current location. And tells you with Doner store is closest.

You may assume you can reach at least one Doner Shop from your current location. You always start at the crossing numbered 1.

Input

A line with two integers: $N, 1 < N \leq 10000$, the number of crossings in the city, and $S, 1 < S \leq 100000$, the number of streets in the city.

S lines with three space seperated integers, $a, b, l, 1 \leq a, b \leq N, 1 \leq l \leq 1000$. Which indicate that a street with length l connects crossing a with crossing b. All streets are bi-directional.

An integer $m, 1 \leq m \leq 1000$, the number of Doner Shops in the city.

M lines with one integer $c, 1 \leq c \leq N$ which indicates that on crossing c you can find a Doner shop.

Output

One line with two space separated integers:

C The number of the crossing at which you can find the closest Doner Shop. If multiple doner stores are equally far away from you, give the one that has the lowest number.

L The length of the shortest route to the closest Doner Shop.

Examples

input 1	output 1
6 6 1 2 5 1 5 1 5 3 2 2 3 1 2 4 1 3 6 3 2 4 6	4 5

input 2	output 2
11 13 1 2 5 2 3 30 2 4 16 4 3 17 2 5 6 5 6 7 6 10 8 10 9 12 8 9 11 8 11 14 7 8 10 1 7 9 5 8 13 3 11 10 3	10 26

E - Efficient Printing

The Factorial Poster Company (FPC) prints posters that display the result of any factorial that their customers wish for. Recently, they got an order from Professor D.R. Ingenious, who wants to do an experiment with very large factorial numbers. The FPC want to be as efficient with printing as possible, and therefore they decided on a way to save paper. Since the larger factorial numbers end in a lot of zeroes, they decide to cut off this number of zeroes z and replace it with "$\cdot 10^z$".

You are given the task to calculate, for every order of Prof. Ingenious, how many zeroes z can be cut off from the poster, so that the FPC know how much poster paper they will save.

Input

One line containing one integer n, with $0 \leq n \leq 10^{18}$.

Output

One line containing one integer z, the amount of trailing zeroes of $n!$. Note that any other zeroes in the result of $n!$ do not count, see the second example.

Examples

input 1	**output 1**
1	0

input 2	**output 2**
7	1

input 3	**output 3**
42	9

This page is intentionally left (almost) blank.

F - Floor Price Calculator

A year ago, your friend John moved to Delft. Recently, he called you to ask your for your help: he wants to redecorate his house. You spent many hours looking for all kinds of furniture and decorations. Now there's only one thing left: he wants a new floor in his house. You and John decide to visit a store nearby to look for a nice one.

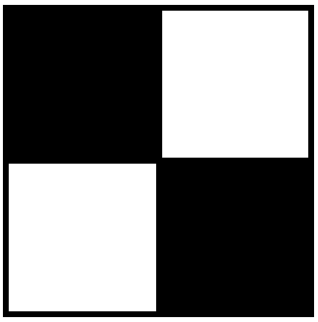


Figure 1 – A floor with five squares.

After looking a bit around in the shop, John noticed a special floor that has the pattern of a checkerboard. The floor is very special since it is sold in one big piece and not in separate black and white tiles. John is interested in the floor and asks the shop keeper for more information. The size of his floor is 5x5 (thus $25m^2$) and a floor in this size will cost him 55 euro. He asks how this price is calculated: it turns out that the floor costs one euro for every square that can be found on the floor. Note that each black or white tile has a size of 1x1 meter. So for instance, the figure visible to the right, has a size of 2x2 meter. In the figure, we count a total of five squares.

John decides to buy the floor and it looks amazing in his house! You decide to also buy this particular floor but your house is way bigger and you would like to know how much you will have to pay. You decide to write a program to calculate the total price in euro for a given floor of n x n meters.

Input

An integer n with $n \leq 10^5$, representing a n x n floor.

Output

The price in euro you have to pay for the n x n floor.

Examples

input 1	output 1
1	1

input 2	output 2
5	55

input 3	output 3
98583	319368033160804

G - Guessing Game

Some days ago a friend of yours got himself a very powerful computer system. Using chatting software, you ask him what the system costs. He won't tell you the cost, but he will let you guess it once he has assembled his new system.

You have to pick a number, and he will message you back whether your guess is too low, too high or correct. Since you only have a limited amount of time to guess the correct price after he is done assembling, you decide to write a program to read his responses and automatically send your guesses for you.

The input given depends interactively on the output of your program.

Input

The input consists of a series of words, each line being one of three:
LOWER, your guess was too high.
HIGHER, your guess was too low.
GAME, you guessed correctly

Note: you will have to start with a guess before you will receive any input.

Output

The output consists of two options:
ASK x, guess the number x.
OVER y, the amount y, $0 \leq y \leq 10^9$, the system costs. This will end the guessing game.

This page is intentionally left (almost) blank.

H - Hungry Wolves

Farmer Fred has a farm with a circular-shaped field on which he is keeping horses. A few of his horses have already been killed by a pack of hungry wolves, therefore farmer Fred wants to construct fences around the field. Fred knows the area of his field. Your job is to create a program that takes as input the area of his circular shaped field in square meters and outputs the total length of the fence needed guard the horses from the wolves, in meters. Note that Fred is going to buy his fence from the Fence Protect Corporation (FPC) which only sells fences per 10 centimeter.

Input

A single integer f ($1 \leq f \leq 10^{18}$), the area of the field in square meters.

Output

Output the total length of fence needed for the field, in meters.

Examples

input 1	**output 1**
`12`	`12.3`

input 2	**output 2**
`46`	`24.1`

input 3	**output 3**
`96`	`34.8`

This page is intentionally left (almost) blank.

I - Impressive Beers

You and your friends find yourselves in a bar after class. As you are a really big fan of specialty beers you want to drink as many different beers as possible. Only issue of course being that you are still a student, and therefore you do not have enough money to buy every beer in the pub.

But of course, all beers are not created equally! You know of every type of beer the price and how much happiness it will give you. As you want to be as happy as possible, and being a Delft student, you want to find out how much happiness you can buy using an optimal strategy. So you decide to write a program that tells you how much happiness you can buy with a given amount of money. But since you like to drink different beers, you will never order the same beer twice, your program should take this into account.

Input

One line with two integers: $N, 1 \leq N \leq 500$ the number of different beers the pub offers, and $M, 1 \leq M \leq 10000$ the amount of money that you have.

Followed by N lines with on each line a type of beer which is indicated by two integers: $p, 1 \leq p \leq 1000$, the price of the beer and $h, 1 \leq h \leq 1000$, the happiness you will gain from this beer.

Output

A single line with a single integer H, the maximal happiness you can gain by buying different kinds of beers, within your budget.

Examples

In the first sample your max happiness will be 57, as you can buy the beers priced 10, 5 and 4 to come to a hapiness of 57 (with a singe coin left to spend, buy there is no more beer you could buy for this price). Every other combination will give you less hapiness or will cost more than your budget. (for example, the beers of 10 + 9 will give you only 50 hapiness).

input 1	**output 1**
5 20 20 50 10 30 5 15 4 12 9 20	57

In the second sample you will buy each beer, giving you 127 hapiness, but it will not give you any more satisfaction to buy the same beer multiple times, so you are stuck at 127 and have some money left.

input 2	**output 2**
5 100 20 50 10 30 5 15 4 12 9 20	127