Segunda Competencia

A. Happy Birthday UN

2.0 s, 128 MB

The Universidad Nacional de Colombia (UN) was created on September 22 of 1867, a Sunday. In 2017 the UN had its sesquicentenary, i.e. it was its 150th birthday. The sesquicentenary of the UN was on a Friday. As it is really important to know all the details corresponding to the date of this huge celebration for every year, we want to know which day of the week corresponds to a specific birthday of the Universidad Nacional de Colombia.

Remember that a year usually has 365 days, except for some years, called leaps-years which have 366 days. To know which year is a leap-year we have the following rule: *Every year that is exactly divisible by four is a leap year, except for years that are exactly divisible by 100, but these centurial years are leap years if they are exactly divisible by 400*. For example, the years 1700, 1800, and 1900 were not leap years, but the years 1600 and 2000 were.

Input

The input consists of a number n ($0 \leq n \leq 10000$)

Output

Output just one word corresponding to the day of the week at the n - th birthday of the UN.

Print the day on the following format (without quotation marks): 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday' or 'Sunday'.

input
0
output
Sunday

input
150
output
Friday

B. Production in Yekaterinburg

2.0 s, 256 MB

Yekaterinburg is a Russian city located in the border between Europe and Asia, in the Ural Mountains. It's the fourth largest city in Russia, with more than 1.4 million residents. Its main economical activity is related to the production of industrial machines. The city's factories produce a large part of the machines used in Russia, and they are exported to several countries. In fact, the production of industrial tools is famous in the country. The tools are build by highly specialized machines and, for each tool to be produced the machines take some pre-determined time to produce it.

One of the factories has only one of those machines and its manager need your help to increase productivity. The tool orders arrive in the factory along the day, that is, they are not necessarily known at the beginning of the day. The manager thinks the employees are not choosing a good order to process the orders, and wants to analyze orders from previous days. This way, he asks you to determine, for a given day, the minimum instant at which all orders can be processed and finalized.

Input

On the first line, a single integer T, the number of test cases.

Each test case has a single integer N on the first line, the number of orders to process in that day. The i-th of the next N lines has two integers s_i and d_i, the time at which the i-th order is available and the time it takes to process it, respectively.

Limits

- $1 \leq N \leq 10^5$
- The sum of N over all test cases won't exceed $5 \cdot 10^5$
- $1 \leq s_i, d_i \leq 10^4$

Output

For each test case, print the minimum instant at which all orders can be processed and finalized.

input
2 5 10 2 6 1 4 3 1 2 1 4 4 1 1 1 3 5 2 5 3
output
13 10

C. Keep Your Style

2.5 s, 256 MB

The UNAL programming coaches have lost a bet, they bet the 6 UNAL teams would occupy the first six positions in the regional finals. Now, they have to shave their heads!. Some of them have more hair than others, and consequently, it takes more time to shave a head completely. However, all of the coaches really love their hair, therefore there is a probability that some (posibly all) of them daunt at the very last moment and do not permit the hairdresser to shave their heads.

Norbert, the hairdresser who loves probability, would like to order the coaches' schedule such that the average time in the hair salon is minimized for all the coaches. First all the coaches are there at the same time, then they start going one by one to Norbert, if by the moment a coach has to go to the hairdresser, he or she daunts then he or she simply leaves the hair salon and it is the turn of the next coach, after the head of a coach is shaved then that coach leaves the hair salon. The time between turns is negligible.

For example, suppose that shaving Diego's head takes 2 minutes and shaving Ivan's takes 3 minutes, but Diego has probability of 0.5 of not daunting meanwhile Ivan for sure will shave his head. If Ivan goes first, he will stay 3 minutes in the hair salon and Diego will stay there 3 minutes if daunting or 5 minutes if not (3 of them waiting for Ivan to finish), in this case the average expected time of the coaches in the hair salon would be 3.5, note this is not the optimal arrangement.

Now, Norbert knows the time it takes to shave each head and the probability of a coach to accept to have the head shaved in the barbershop, help him to know the minimum average expected time in the hair salon of the coaches.

Input

The first line of input is an integer n ($1 \le n \le 5 * 10^5$) - the number of coaches.

The next n lines contain each an integer x ($0 \le x \le 100$) and a decimal number y ($0 \le y \le 1$) separated by a single space - the time in minutes it takes to shave the head of the i - th coach and his probability of *not* daunting, respectively.

Output

Print the minimum expected average time. The relative error of your answer should not exceed 10^{-6}.

input
2 2 0.5 3 1.0
output
2.500000000

input
2 0 0.4 20 0.6
output
6.000000000

D. Marbles Lucky Distribution

1.0 s, 256 MB

Juan have N red marbles, M blue marbles, and K bottles. He will put a certain number of marbles on each of the K bottles such that no bottle remains empty and every marble is inside a bottle.

Andres is a Santa Fe fan, so he will pick one bottle at random with an uniform distribution, then he will pick a marble inside of it at random with an uniform distribution, with the hope is a red marble. As Juan is a Millonarios fan, he wants to distribute the marbles in the bottles such that the probability of Andres picking a blue marble is maximized. Juan has a busy life, therefore he needs your help to determine the best arrangement for the marbles and the probability of Andres getting a blue marble.

Input

The input consist of three integers separated by spaces, N M and K ($1 \le N, M, K \le 10^9$) - the number of red marbles, blue marbles and bottles respectively.

Output

Print the probability of Bob getting a blue marble such that the marble arrangement is optimal. Your answer will be considered correct, if its absolute or relative error does not exceed 10^{-6}.

input
50 50 2
output
0.747474747

E. Cryptography

1.0 s, 256 MB

The United Nations Against Leaks (UNAL) is trying to prove a method to encrypt e-mails to make them more secure to the world. However, this encryption method is not an easy process. First you have the original text to encode, then every character of the text is changed to any other character with a certain cost because the UNAL has to pay some money in order to change characters into other characters (this process can be repeated zero or various times). After this process is finished, we will have a new brand text which will be the message that will be send through the internet.

We already have the original text, the encrypted text, and which characters can be changed along with the corresponding cost. As the UNAL is paying for the encryption, they are hiring you to calculate the minimum cost of the e-mail encryption.

Input

The first line of the input consists of a string s of size $|s|$, ($1 \le |s| \le 10^5$) - the original message.

The second line consists of a string t of size $|t|$, ($|t| = |s|$) - the encrypted message.

The third line consists of a number m ($1 \le m \le 5 * 10^4$), the number of possible changes between characters.

Finally m lines follow, each containing a, b, c, where a and b ($a \ne b$) are ASCII characters and c is an integer ($1 \le c \le 10^3$), the cost of changing a into b.

It is guaranteed that s and t have the same length, don't have any whitespaces, and they are made using only printable characters ASCII (ASCII codes from 33 to 126).

Output

Print one line with the minimum cost of encrypting s to t. If there is no way to encrypt s to t print -1.

input
hello! world! 8 a b 1 a d 3 e o 5 h w 10 l r 12 l e 5 o a 2 o d 8
output
32

input
Aa ab 2 a b 10 a b 12
output
-1

F. Generating Texts

1.0 s, 512 MB

Lucko is a greedy boy that is looking for ways to gain some money, one easy way he has found is to invest in bets on different lotteries, but he knows that the probability of winning a lottery is really low, for this reason he decides to play on a new lottery called *UN-lotto*.

To play *UN-lotto* one chooses a string of length m, then the lottery generates a string of length n ($n \geq m$) randomly, both strings are formed only by lower-case letters from the english alphabet. You win if the string you have chosen is a subsequence of the generated text by *UN-lotto*.

Formally, for a string $s = s_1 s_2 ... s_n$ we say that $s_{a_1} s_{a_2} ... s_{a_m}$ is a subsequence of s of length m if $1 \leq a_1 < a_2 < ... < a_m \leq n$. For example, suppose that $n = 4$ and $m = 3$ and Lucko chooses the string `abc`, some of the winning generated strings for him would be `abcb`, `abbc` and `abzc`, and some of the losing generated strings would be `acba`, `azbz` and `zzzz`

Good news for Lucko, information about this lottery has been leaked:

- The text is generated taking each letter randomly from a probability distribution.
- The selection of the letter for each position in the text is independent.
- The probability distribution of the letters was also leaked.

Lucko has already chosen his string to play *UN-lotto*, help him using the leaked information to tell him which is his probability P / Q of winning. As Lucko only likes low integer numbers calculate the probability modulo $10^9 + 7$, i.e., calculate $P * Q^{-1} \mod 10^9 + 7$

Input

The first line of input contains 2 numbers n and m ($1 \leq m \leq n \leq 5000$) — the length of the string generated by *UN-lotto* and the length of the string chosen by Lucko respectively.

The second line of input contains s — the string chosen by Lucko.

Next 26 lines contains each a pair of integers separated by spaces, i.e. the i-th line contains p_i, q_i, where $\frac{p_i}{q_i}$ corresponds to the probability of taking the i-th letter, being `a` the first letter, `b` the second letter and so on. For every $i = 1, ..., 26$, $0 \leq p_i \leq 1000$, $1 \leq q_i \leq 1000$ and p_i / q_i is an irreducible fraction.

It is guaranteed that $\sum_{i=1}^{26} \frac{p_i}{q_i} = 1$.

Output

Output one integer — the probability modulo $10^9 + 7$ of Lucko winning *UN-lotto* with the string s.

input

```
5 1
a
1 1
0 1
... 24 more lines ...
```

output

```
1
```

input

```
5 3
aaa
1 2
1 2
0 1
... 23 more lines ...
```

output

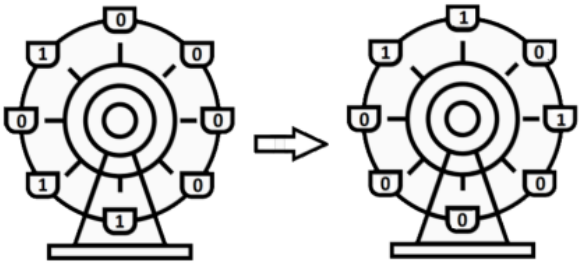
```
500000004
```

G. Intense Bit Wheel

1.0 s, 256 MB

There is a new intense giant wheel in UNAL town, in UNAL town the wheels move in counterclockwise direction, also as this wheel has n cabins, everyone and everything can play. In particular, numbers come to this attraction. However, a complete number doesn't fit in one cabin, in fact, each cabin have space only for one bit. For this reason a number splits into its binary from and ride the wheel each bit per cabin. As numbers do not want to mess it up, the bits enter into the attraction in the same order the number is formed. Nonetheless, the wheel moves k times (entering and exiting from the attraction don't count as moves), then when the bits have to get out of the wheel they probably don't do it in the same order they entered, therefore making a different number.

For example, when the number 13 enter in a intense 8-bit it will look like the left side of the image below, after 5 moves the wheel will look like the right side of the image, in this case the number exiting the wheel will be 161.



You as the chief of the numbers in the UNAL town want to know how the wheel can affect the numbers that ride this attraction.

Input

First line of input contains 2 numbers n ($1 \leq n \leq 50$) and m ($1 \leq m \leq 1000$) - the quantity of cabins in the wheel and the quantity of numbers that ride in the wheel, respectively.

Next m lines of input contains each 2 integers num ($0 \leq num < 2^n$) and k ($1 \leq k \leq 10^{18}$) - the number that ride the wheel and the quantity of times the wheel move, respectively.

Output

For each number that ride the wheel print the resulting number after leaving the wheel. Output this number in its decimal form.

input

```
8 3
1 1
13 5
17 12
```

output

```
2
161
17
```

Note that all numbers that ride into the wheel have exactly n bits, for example the number 13 in a 8-cabin wheel is not 1101 but 00001101 (see example).