A - Yay!

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 150 points

Problem Statement

You are given a string S consisting of lowercase English letters. The length of S is between 3 and 100, inclusive.

All characters but one of S are the same.

Find x such that the x-th character of S differs from all other characters.

Constraints

- S is a string of length between 3 and 100, inclusive, consisting of two different lowercase English letters.
- All characters but one of S are the same.

Input

The input is given from Standard Input in the following format:

S

Output

Print the answer.

Sample Input 1

```
yay
```

Sample Output 1

```
2
```

The second character of `yay` differs from the first and third characters.

Sample Input 2

```
egg
```

Sample Output 2

```
1
```

Sample Input 3

```
zzzzzwz
```

Sample Output 3

```
6
```

B - Which is ahead?

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 200 points

Problem Statement

There are N people standing in a line. The person standing at the i-th position from the front is person P_i.

Process Q queries. The i-th query is as follows:

- You are given integers A_i and B_i. Between person A_i and person B_i, print the person number of the person standing further to the front.

Constraints

- All inputs are integers.
- $1 \leq N \leq 100$
- $1 \leq P_i \leq N$
- $P_i \neq P_j \ (i \neq j)$
- $1 \leq Q \leq 100$
- $1 \leq A_i < B_i \leq N$

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
P_1 \ \dots \ P_N \\
Q \\
A_1 \ B_1 \\
\vdots \\
A_Q \ B_Q
\end{array}
$$

Output

Print Q lines. The i-th line should contain the response for the i-th query.

Sample Input 1

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

Sample Output 1

```
2
2
1
```

In the first query, person 2 is at the first position from the front, and person 3 is at the third position, so person 2 is further to the front.

In the second query, person 1 is at the second position from the front, and person 2 is at the first position, so person 2 is further to the front.

In the third query, person 1 is at the second position from the front, and person 3 is at the third position, so person 1 is further to the front.

Sample Input 2

```
7
3 7 2 1 6 5 4
13
2 3
1 2
1 3
3 6
3 7
2 4
3 7
1 3
4 7
1 6
2 4
1 3
1 3
```

Sample Output 2

```
3
2
3
3
3
2
3
3
7
1
2
3
3
```

C - Many Replacement

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 350 points

Problem Statement

You are given a string S of length N consisting of lowercase English letters.

You will perform an operation Q times on the string S. The i-th operation $(1 \leq i \leq Q)$ is represented by a pair of characters (c_i, d_i), which corresponds to the following operation:

- Replace all occurrences of the character c_i in S with the character d_i.

Print the string S after all operations are completed.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- S is a string of length N consisting of lowercase English letters.
- $1 \leq Q \leq 2 \times 10^5$
- c_i and d_i are lowercase English letters $(1 \leq i \leq Q)$.
- N and Q are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
S \\
Q \\
c_1 \quad d_1 \\
c_2 \quad d_2 \\
\vdots \\
c_Q \quad d_Q
\end{array}
$$

Output

Print the string S after all operations are completed.

Sample Input 1

```
7
atcoder
4
r a
t e
d v
a r
```

Sample Output 1

```
recover
```

S changes as follows: `atcoder` → `atcodea` → `aecodea` → `aecovea` → `recover`. For example, in the fourth operation, all occurrences of `a` in $S =$ `aecovea` (the first and seventh characters) are replaced with `r`, resulting in $S =$ `recover`.

After all operations are completed, $S =$ `recover`, so print `recover`.

Sample Input 2

```
3
abc
4
a a
s k
n n
z b
```

Sample Output 2

```
abc
```

There may be operations where $c_i = d_i$ or S does not contain c_i.

Sample Input 3

```
34
supercalifragilisticexpialidocious
20
g c
l g
g m
c m
r o
s e
a a
o f
f s
e t
t l
d v
p k
v h
x i
h n
n j
i r
s i
u a
```

Sample Output 3

```
laklimamriiamrmrllrmlrkramrjimrial
```

D - Square Pair

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 400 points

Problem Statement

You are given a sequence of non-negative integers $A = (A_1, \ldots, A_N)$ of length N. Find the number of pairs of integers (i, j) that satisfy both of the following conditions:

- $1 \leq i < j \leq N$
- $A_i A_j$ is a square number.

Here, a non-negative integer a is called a square number when it can be expressed as $a = d^2$ using some non-negative integer d.

Constraints

- All inputs are integers.
- $2 \leq N \leq 2 \times 10^5$
- $0 \leq A_i \leq 2 \times 10^5$

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
A_1 \ \ldots \ A_N
\end{array}
$$

Output

Print the answer.

Sample Input 1

```
5
0 3 2 8 12
```

Sample Output 1

```
6
```

Six pairs of integers, $(i,j) = (1,2), (1,3), (1,4), (1,5), (2,5), (3,4)$, satisfy the conditions.

For example, $A_2 A_5 = 36$, and 36 is a square number, so the pair $(i,j) = (2,5)$ satisfies the conditions.

Sample Input 2

```
8
2 2 4 6 3 100 100 25
```

Sample Output 2

```
7
```

E - Last Train

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 450 points

Problem Statement

In the country of AtCoder, there are N stations: station 1, station 2, $\ldots$, station N.

You are given M pieces of information about trains in the country. The i-th piece of information $(1 \leq i \leq M)$ is represented by a tuple of six positive integers $(l_i, d_i, k_i, c_i, A_i, B_i)$, which corresponds to the following information:

- For each $t = l_i, l_i + d_i, l_i + 2d_i, \ldots, l_i + (k_i - 1)d_i$, there is a train as follows:
  - The train departs from station A_i at time t and arrives at station B_i at time $t + c_i$.

No trains exist other than those described by this information, and it is impossible to move from one station to another by any means other than by train.
Also, assume that the time required for transfers is negligible.

Let $f(S)$ be the latest time at which one can arrive at station N from station S.
More precisely, $f(S)$ is defined as the maximum value of t for which there is a sequence of tuples of four integers $\big((t_i, c_i, A_i, B_i)\big)_{i=1,2,\ldots,k}$ that satisfies all of the following conditions:

- $t \leq t_1$
- $A_1 = S, B_k = N$
- $B_i = A_{i+1}$ for all $1 \leq i < k$,
- For all $1 \leq i \leq k$, there is a train that departs from station A_i at time t_i and arrives at station B_i at time $t_i + c_i$.
- $t_i + c_i \leq t_{i+1}$ for all $1 \leq i < k$.

If no such t exists, set $f(S) = -\infty$.

Find $f(1), f(2), \ldots, f(N-1)$.

Constraints

- $2 \leq N \leq 2 \times 10^5$
- $1 \leq M \leq 2 \times 10^5$
- $1 \leq l_i, d_i, k_i, c_i \leq 10^9 \ (1 \leq i \leq M)$
- $1 \leq A_i, B_i \leq N \ (1 \leq i \leq M)$
- $A_i \neq B_i \ (1 \leq i \leq M)$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{llllll}
N & M & & & & \\
l_1 & d_1 & k_1 & c_1 & A_1 & B_1 \\
l_2 & d_2 & k_2 & c_2 & A_2 & B_2 \\
\vdots & & & & & \\
l_M & d_M & k_M & c_M & A_M & B_M
\end{array}
$$

Output

Print $N-1$ lines. The k-th line should contain $f(k)$ if $f(k) \neq -\infty$, and `Unreachable` if $f(k) = -\infty$.

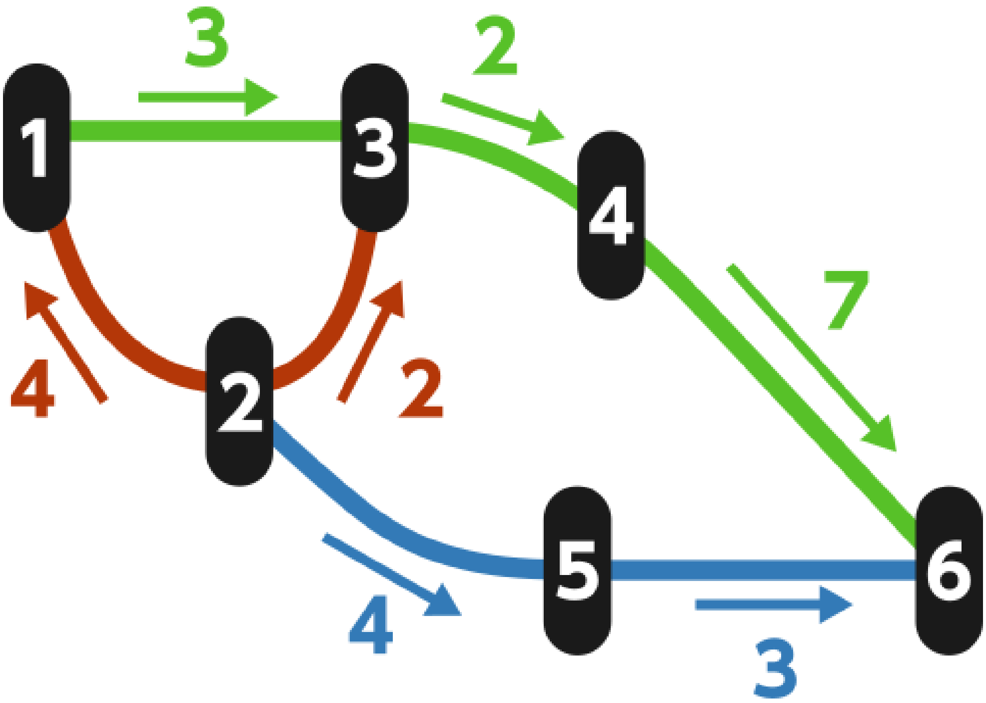
Sample Input 1

```
6 7
10 5 10 3 1 3
13 5 10 2 3 4
15 5 10 7 4 6
3 10 2 4 2 5
7 10 2 3 5 6
5 3 18 2 2 3
6 3 20 4 2 1
```

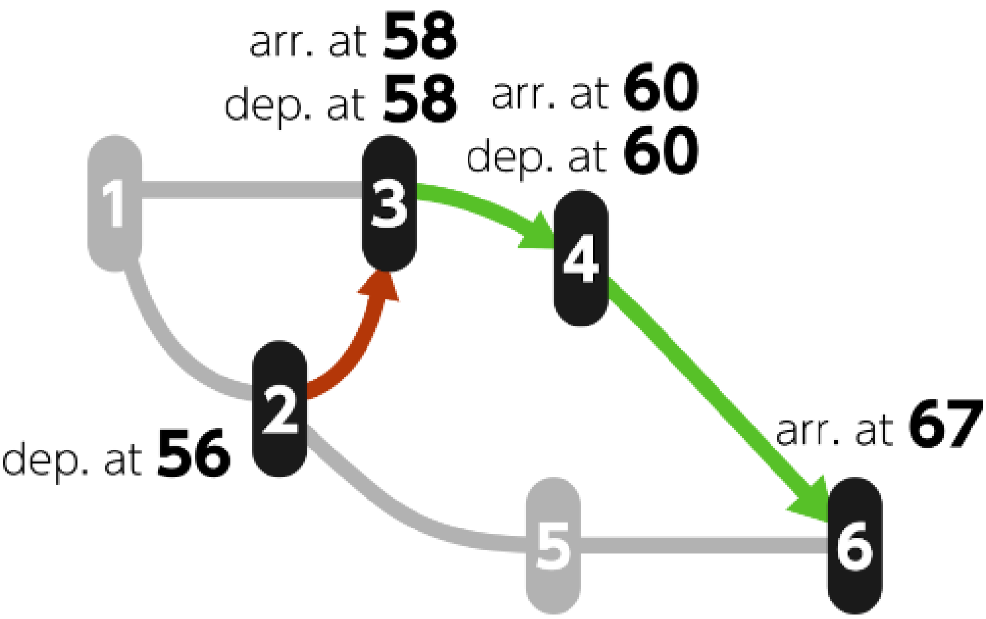
Sample Output 1

```
55
56
58
60
17
```

The following diagram shows the trains running in the country (information about arrival and departure times is omitted).



Consider the latest time at which one can arrive at station 6 from station 2. As shown in the following diagram, one can arrive at station 6 by departing from station 2 at time 56 and moving as station $2 \rightarrow$ station $3 \rightarrow$ station $4 \rightarrow$ station 6.



It is impossible to depart from station 2 after time 56 and arrive at station 6, so $f(2) = 56$.

Sample Input 2

```
5 5
1000000000 1000000000 1000000000 1000000000 1 5
5 9 2 6 2 3
10 4 1 6 2 3
1 1 1 1 3 5
3 1 4 1 5 1
```

Sample Output 2

```
1000000000000000000
Unreachable
1
Unreachable
```

There is a train that departs from station 1 at time 10^{18} and arrives at station 5 at time $10^{18} + 10^9$. There are no trains departing from station 1 after that time, so $f(1) = 10^{18}$. As seen here, the answer may not fit within a 32 bit integer.

Also, both the second and third pieces of information guarantee that there is a train that departs from station 2 at time 14 and arrives at station 3 at time 20. As seen here, some trains may appear in multiple pieces of information.

Sample Input 3

```
16 20
4018 9698 2850 3026 8 11
2310 7571 7732 1862 13 14
2440 2121 20 1849 11 16
2560 5115 190 3655 5 16
1936 6664 39 8822 4 16
7597 8325 20 7576 12 5
5396 1088 540 7765 15 1
3226 88 6988 2504 13 5
1838 7490 63 4098 8 3
1456 5042 4 2815 14 7
3762 6803 5054 6994 10 9
9526 6001 61 8025 7 8
5176 6747 107 3403 1 5
2014 5533 2031 8127 8 11
8102 5878 58 9548 9 10
3788 174 3088 5950 3 13
7778 5389 100 9003 10 15
556 9425 9458 109 3 11
5725 7937 10 3282 2 9
6951 7211 8590 1994 15 12
```

Sample Output 3

```
720358
77158
540926
255168
969295
Unreachable
369586
466218
343148
541289
42739
165772
618082
16582
591828
```

F - Black Jack

Time Limit: 2 sec / Memory Limit: 1024 MB

Score: 550 points

Problem Statement

You will play a game against a dealer. The game uses a D-sided die (dice) that shows an integer from 1 to D with equal probability, and two variables x and y initialized to 0. The game proceeds as follows:

- You may perform the following operation any number of times: roll the die and add the result to x. You can choose whether to continue rolling or not after each roll.
- Then, the dealer will repeat the following operation as long as $y < L$: roll the die and add the result to y.
- If $x > N$, you lose. Otherwise, you win if $y > N$ or $x > y$ and lose if neither is satisfied.

Determine the probability of your winning when you act in a way that maximizes the probability of winning.

Constraints

- All inputs are integers.
- $1 \leq L \leq N \leq 2 \times 10^5$
- $1 \leq D \leq N$

Input

The input is given from Standard Input in the following format:

```
N L D
```

Output

Print the answer. Your output will be considered correct if its absolute or relative error from the true value is at most 10^{-6}.

Sample Input 1

```
3 2 2
```

Sample Output 1

```
0.468750000000000
```

It can be proved that the optimal strategy is to continue rolling as long as x is not greater than 2.

Sample Input 2

```
200000 200000 200000
```

Sample Output 2

```
0.999986408692793
```

G - Retroactive Range Chmax

Time Limit: 5 sec / Memory Limit: 1024 MB

Score: 625 points

Problem Statement

You are given an integer sequence $A = (A_1, A_2, \dots, A_N)$ of length N.

Process Q operations in order. There are three types of operations:

- A type-1 operation is represented by a triple of integers (l, r, x), which corresponds to replacing A_i with $\max\{A_i, x\}$ for each $i = l, l+1, \dots, r$.
- A type-2 operation is represented by an integer i, which corresponds to canceling the i-th operation (it is guaranteed that the i-th operation is of type 1 and has not already been canceled). The new sequence A can be obtained by performing all type-1 operations that have **not been canceled**, starting from the initial state.
- A type-3 operation is represented by an integer i, which corresponds to printing the current value of A_i.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq A_i \leq 10^9$ $(1 \leq i \leq N)$
- $1 \leq Q \leq 2 \times 10^5$
- In a type-1 operation, $1 \leq l \leq r \leq N$ and $1 \leq x \leq 10^9$.
- In a type-2 operation, i is not greater than the number of operations given before, and $1 \leq i$.
- In a type-2 operation, the i-th operation is of type 1.
- In type-2 operations, the same i does not appear multiple times.
- In a type-3 operation, $1 \leq i \leq N$.
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
A_1 \quad A_2 \quad \ldots \quad A_N \\
Q \\
\mathrm{query}_1 \\
\mathrm{query}_2 \\
\vdots \\
\mathrm{query}_Q
\end{array}
$$

Here, query_k $(1 \leq k \leq Q)$ represents the k-th operation, and depending on the type of the k-th operation, one of the following is given.

For a type-1 operation, the following is given, meaning that the k-th operation is a type-1 operation represented by (l, r, x):

$$1 \quad l \quad r \quad x$$

For a type-2 operation, the following is given, meaning that the k-th operation is a type-2 operation represented by i:

$$2 \quad i$$

For a type-3 operation, the following is given, meaning that the k-th operation is a type-3 operation represented by i:

$$3 \quad i$$

Output

Let q be the number of type-3 operations. Print q lines. The i-th line $(1 \leq i \leq q)$ should contain the value that should be printed for the i-th given type-3 operation.

Sample Input 1

```
6
2 7 1 8 2 8
15
3 1
3 3
3 4
1 1 5 4
3 1
3 3
3 4
1 3 6 9
3 1
3 3
3 4
2 4
3 1
3 3
3 4
```

Sample Output 1

```
2
1
8
4
4
8
4
9
9
2
9
9
```

Initially, the sequence A is $(2, 7, 1, 8, 2, 8)$.

For the 1-st, 2-nd, 3-rd operations, print the values of A_1, A_3, A_4, which are $2, 1, 8$, respectively.

The 4-th operation replaces the values of A_1, A_2, A_3, A_4, A_5 with $\max\{A_i, 4\}$. Just after this operation, A is $(4, 7, 4, 8, 4, 8)$.

For the 5-th, 6-th, 7-th operations, print the values of A_1, A_3, A_4 at this point, which are $4, 4, 8$, respectively.

The 8-th operation replaces the values of A_3, A_4, A_5, A_6 with $\max\{A_i, 9\}$. Just after this operation, A is $(4, 7, 9, 9, 9, 9)$.

For the 9-th, 10-th, 11-th operations, print the values of A_1, A_3, A_4 at this point, which are $4, 9, 9$, respectively.

The 12-th operation cancels the 4-th operation. Just after this operation, A is $(2, 7, 9, 9, 9, 9)$.

For the 13-th, 14-th, 15-th operations, print the values of A_1, A_3, A_4 at this point, which are $2, 9, 9$, respectively.

Sample Input 2

```
24
721 78 541 256 970 478 370 467 344 542 43 166 619 17 592 222 983 729 338 747 62 452 815 838
35
3 10
3 8
3 8
3 13
3 9
1 1 17 251
3 3
3 19
3 13
3 22
3 1
3 15
3 18
3 10
3 15
1 16 19 883
1 8 23 212
3 5
3 13
2 6
3 15
1 5 18 914
2 17
3 20
1 23 23 56
3 13
2 25
3 13
3 13
3 10
2 16
1 17 22 308
3 19
3 17
3 7
```

Sample Output 2

```
542
467
467
619
344
541
338
619
452
721
592
729
542
592
970
619
592
747
914
914
914
914
338
983
914
```