A. Bachgold Problem

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Bachgold problem is very easy to formulate. Given a positive integer n represent it as a sum of **maximum possible** number of prime numbers. One can prove that such representation exists for any integer greater than 1.

Recall that integer k is called *prime* if it is greater than 1 and has exactly two positive integer divisors — 1 and k.

Input

The only line of the input contains a single integer n ($2 \le n \le 100\,000$).

Output

The first line of the output contains a single integer k — maximum possible number of primes in representation.

The second line should contain k primes with their sum equal to n. You can print them in any order. If there are several optimal solution, print any of them.

Examples

input

```
5
```

output

```
2
2 3
```

input

```
6
```

output

```
3
2 2 2
```

B. Parallelogram is Back

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Long time ago Alex created an interesting problem about parallelogram. The input data for this problem contained four integer points on the Cartesian plane, that defined the set of vertices of some non-degenerate (positive area) parallelogram. Points not necessary were given in the order of clockwise or counterclockwise traversal.

Alex had very nice test for this problem, but is somehow happened that the last line of the input was lost and now he has only three out of four points of the original parallelogram. He remembers that test was so good that he asks you to restore it given only these three points.

Input

The input consists of three lines, each containing a pair of integer coordinates x_i and y_i ($-1000 \le x_i, y_i \le 1000$). It's guaranteed that these three points do not lie on the same line and no two of them coincide.

Output

First print integer k — the number of ways to add one new integer point such that the obtained set defines some parallelogram of positive area. There is no requirement for the points to be arranged in any special order (like traversal), they just define the set of vertices.

Then print k lines, each containing a pair of integer — possible coordinates of the fourth point.

Example

input

```
0 0
1 0
0 1
```

output

```
3
1 -1
-1 1
1 1
```

Note

If you need clarification of what parallelogram is, please check Wikipedia page:

https://en.wikipedia.org/wiki/Parallelogram

C. Voting

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n employees in Alternative Cake Manufacturing (ACM). They are now voting on some very important question and the leading world media are trying to predict the outcome of the vote.

Each of the employees belongs to one of two fractions: depublicans or remocrats, and these two fractions have opposite opinions on what should be the outcome of the vote. The voting procedure is rather complicated:

1. Each of n employees makes a statement. They make statements one by one starting from employees 1 and finishing with employee n. If at the moment when it's time for the i-th employee to make a statement he no longer has the right to vote, he just skips his turn (and no longer takes part in this voting).
2. When employee makes a statement, he can do nothing or declare that one of the other employees no longer has a right to vote. It's allowed to deny from voting people who already made the statement or people who are only waiting to do so. If someone is denied from voting he no longer participates in the voting till the very end.
3. When all employees are done with their statements, the procedure repeats: again, each employees starting from 1 and finishing with n who are still eligible to vote make their statements.
4. The process repeats until there is only one employee eligible to vote remaining and he determines the outcome of the whole voting. Of course, he votes for the decision suitable for his fraction.

You know the order employees are going to vote and that they behave optimal (and they also know the order and who belongs to which fraction). Predict the outcome of the vote.

Input

The first line of the input contains a single integer n ($1 \leq n \leq 200\,000$) — the number of employees.

The next line contains n characters. The i-th character is 'D' if the i-th employee is from depublicans fraction or 'R' if he is from remocrats.

Output

Print 'D' if the outcome of the vote will be suitable for depublicans and 'R' if remocrats will win.

Examples

input

```
5
DDRRR
```

output

```
D
```

input

```
6
DDRRRR
```

output

```
R
```

Note

Consider one of the voting scenarios for the first sample:

1. Employee 1 denies employee 5 to vote.
2. Employee 2 denies employee 3 to vote.
3. Employee 3 has no right to vote and skips his turn (he was denied by employee 2).
4. Employee 4 denies employee 2 to vote.
5. Employee 5 has no right to vote and skips his turn (he was denied by employee 1).
6. Employee 1 denies employee 4.
7. Only employee 1 now has the right to vote so the voting ends with the victory of depublicans.

D. Leaving Auction

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

There are n people taking part in auction today. The rules of auction are classical. There were n bids made, though it's not guaranteed they were from different people. It might happen that some people made no bids at all.

Each bid is define by two integers (a_i, b_i), where a_i is the index of the person, who made this bid and b_i is its size. Bids are given in chronological order, meaning $b_i < b_{i+1}$ for all $i < n$. Moreover, participant never makes two bids in a row (no one updates his own bid), i.e. $a_i \neq a_{i+1}$ for all $i < n$.

Now you are curious with the following question: who (and which bid) will win the auction if some participants were absent? Consider that if someone was absent, all his bids are just removed and no new bids are added.

Note, that if during this imaginary exclusion of some participants it happens that some of the remaining participants makes a bid twice (or more times) in a row, only first of these bids is counted. For better understanding take a look at the samples.

You have several questions in your mind, compute the answer for each of them.

Input

The first line of the input contains an integer n $(1 \leq n \leq 200\,000)$ — the number of participants and bids.

Each of the following n lines contains two integers a_i and b_i $(1 \leq a_i \leq n, 1 \leq b_i \leq 10^9, b_i < b_{i+1})$ — the number of participant who made the i-th bid and the size of this bid.

Next line contains an integer q $(1 \leq q \leq 200\,000)$ — the number of question you have in mind.

Each of next q lines contains an integer k $(1 \leq k \leq n)$, followed by k integers l_j $(1 \leq l_j \leq n)$ — the number of people who are not coming in this question and their indices. It is guarenteed that l_j values are different for a single question.

It's guaranteed that the sum of k over all question won't exceed $200\,000$.

Output

For each question print two integer — the index of the winner and the size of the winning bid. If there is no winner (there are no remaining bids at all), print two zeroes.

Examples

input

```
6
1 10
2 100
3 1000
1 10000
2 100000
3 1000000
3
1 3
2 2 3
2 1 2
```

output

```
2 100000
1 10
3 1000
```

input

```
3
1 10
2 100
1 1000
2
2 1 2
2 2 3
```

output

```
0 0
1 10
```

Note

Consider the first sample:

- In the first question participant number 3 is absent so the sequence of bids looks as follows:
  1. $1\ 10$
  2. $2\ 100$
  3. $1\ 10\,000$

4. $2\ 100\ 000$

   Participant number 2 wins with the bid $100\ 000$.

- In the second question participants 2 and 3 are absent, so the sequence of bids looks:

  1. $1\ 10$
  2. $1\ 10\ 000$

  The winner is, of course, participant number 1 but the winning bid is 10 instead of $10\ 000$ as no one will ever increase his own bid (in this problem).

- In the third question participants 1 and 2 are absent and the sequence is:

  1. $3\ 1\ 000$
  2. $3\ 1\ 000\ 000$

  The winner is participant 3 with the bid $1\ 000$.

E. Inversions After Shuffle

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given a permutation of integers from 1 to n. Exactly once you apply the following operation to this permutation: pick a random segment and shuffle its elements. Formally:

1. Pick a random segment (continuous subsequence) from l to r. All segments are equiprobable.
2. Let $k = r - l + 1$, i.e. the length of the chosen segment. Pick a random permutation of integers from 1 to k, $p_1, p_2, \ldots, p_k$. All $k!$ permutation are equiprobable.
3. This permutation is applied to elements of the chosen segment, i.e. permutation $a_1, a_2, \ldots, a_{l-1}, a_l, a_{l+1}, \ldots, a_{r-1}, a_r, a_{r+1}, \ldots, a_n$ is transformed to $a_1, a_2, \ldots, a_{l-1}, a_{l-1+p_1}, a_{l-1+p_2}, \ldots, a_{l-1+p_{k-1}}, a_{l-1+p_k}, a_{r+1}, \ldots, a_n$.

Inversion if a pair of elements (not necessary neighbouring) with the wrong relative order. In other words, the number of inversion is equal to the number of pairs (i, j) such that $i < j$ and $a_i > a_j$. Find the expected number of inversions after we apply exactly one operation mentioned above.

Input

The first line contains a single integer n $(1 \le n \le 100\,000)$ — the length of the permutation.

The second line contains n distinct integers from 1 to n — elements of the permutation.

Output

Print one real value — the expected number of inversions. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-9}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if .

Example

input

```
3
2 3 1
```

output

```
1.916666666666666666666666666667
```