Codeforces Round #294 (Div. 2)

A. A and B and Chess

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

A and B are preparing themselves for programming contests.

To train their logical thinking and solve problems better, A and B decided to play chess. During the game A wondered whose position is now stronger.

For each chess piece we know its weight:

- the queen's weight is 9,
- the rook's weight is 5,
- the bishop's weight is 3,
- the knight's weight is 3,
- the pawn's weight is 1,
- the king's weight isn't considered in evaluating position.

The player's weight equals to the sum of weights of all his pieces on the board.

As A doesn't like counting, he asked you to help him determine which player has the larger position weight.

Input

The input contains eight lines, eight characters each — the board's description.

The white pieces on the board are marked with uppercase letters, the black pieces are marked with lowercase letters.

The white pieces are denoted as follows: the queen is represented is 'Q', the rook — as 'R', the bishop — as'B', the knight — as 'N', the pawn — as 'P', the king — as 'K'.

The black pieces are denoted as 'q', 'r', 'b', 'n', 'p', 'k', respectively.

An empty square of the board is marked as '.' (a dot).

It is **not guaranteed** that the given chess position can be achieved in a real game. Specifically, there can be an arbitrary (possibly zero) number pieces of each type, the king may be under attack and so on.

Output

Print "White" (without quotes) if the weight of the position of the white pieces is more than the weight of the position of the black pieces, print "Black" if the weight of the black pieces is more than the weight of the white pieces and print "Draw" if the weights of the white and black pieces are equal.

Examples

input
...QK... ....rk...
output
White

input
rnbqkbnr pppppppp .... PPPPPPPP RNBQKBNR

output

```
Draw
```

input

```
rppppppr
...k....
........
........
........
........
K...Q...
........
```

output

```
Black
```

Note

In the first test sample the weight of the position of the white pieces equals to 9, the weight of the position of the black pieces equals 5.

In the second test sample the weights of the positions of the black and the white pieces are equal to 39.

In the third test sample the weight of the position of the white pieces equals to 9, the weight of the position of the black pieces equals to 16.

B. A and B and Compilation Errors

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

A and B are preparing themselves for programming contests.

B loves to debug his code. But before he runs the solution and starts debugging, he has to first compile the code.

Initially, the compiler displayed n compilation errors, each of them is represented as a positive integer. After some effort, B managed to fix some mistake and then another one mistake.

However, despite the fact that B is sure that he corrected the two errors, he can not understand exactly what compilation errors disappeared — the compiler of the language which B uses shows errors in the new order every time! B is sure that unlike many other programming languages, compilation errors for his programming language do not depend on each other, that is, if you correct one error, the set of other error does not change.

Can you help B find out exactly what two errors he corrected?

Input

The first line of the input contains integer n ($3 \le n \le 10^5$) — the initial number of compilation errors.

The second line contains n space-separated integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the errors the compiler displayed for the first time.

The third line contains $n - 1$ space-separated integers $b_1, b_2, \ldots, b_{n-1}$ — the errors displayed at the second compilation. It is guaranteed that the sequence in the third line contains all numbers of the second string except for exactly one.

The fourth line contains $n - 2$ space-separated integers $c_1, c_2, \ldots, c_{n-2}$ — the errors displayed at the third compilation. It is guaranteed that the sequence in the fourth line contains all numbers of the third line except for exactly one.

Output

Print two numbers on a single line: the numbers of the compilation errors that disappeared after B made the first and the second correction, respectively.

Examples

input
5 1 5 8 123 7 123 7 5 1 5 1 7
output
8 123

input
6 1 4 3 3 5 7 3 7 5 4 3 4 3 7 5
output
1 3

Note

In the first test sample B first corrects the error number 8, then the error number 123.

In the second test sample B first corrects the error number 1, then the error number 3. Note that if there are multiple errors with the same number, B can correct only one of them in one step.

C. A and B and Team Training

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

A and B are preparing themselves for programming contests.

An important part of preparing for a competition is sharing programming knowledge from the experienced members to those who are just beginning to deal with the contests. Therefore, during the next team training A decided to make teams so that newbies are solving problems together with experienced participants.

A believes that the optimal team of three people should consist of one experienced participant and two newbies. Thus, each experienced participant can share the experience with a large number of people.

However, B believes that the optimal team should have two experienced members plus one newbie. Thus, each newbie can gain more knowledge and experience.

As a result, A and B have decided that all the teams during the training session should belong to one of the two types described above. Furthermore, they agree that the total number of teams should be as much as possible.

There are n experienced members and m newbies on the training session. Can you calculate what maximum number of teams can be formed?

Input

The first line contains two integers n and m ($0 \le n, m \le 5 \cdot 10^5$) — the number of experienced participants and newbies that are present at the training session.

Output

Print the maximum number of teams that can be formed.

Examples

input
2 6
output
2

input
4 5
output
3

Note

Let's represent the experienced players as XP and newbies as NB.

In the first test the teams look as follows: (XP, NB, NB), (XP, NB, NB).

In the second test sample the teams look as follows: (XP, NB, NB), (XP, NB, NB), (XP, XP, NB).

D. A and B and Interesting Substrings

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

A and B are preparing themselves for programming contests.

After several years of doing sports programming and solving many problems that require calculating all sorts of abstract objects, A and B also developed rather peculiar tastes.

A likes lowercase letters of the Latin alphabet. He has assigned to each letter a number that shows how much he likes that letter (he has assigned negative numbers to the letters he dislikes).

B likes substrings. He especially likes the ones that start and end with the same letter (their length must exceed one).

Also, A and B have a string s. Now they are trying to find out how many substrings t of a string s are interesting to B (that is, t starts and ends with the same letter and its length is larger than one), and also the sum of values of all letters (assigned by A), *except for the first and the last one* is equal to zero.

Naturally, A and B have quickly found the number of substrings t that are interesting to them. Can you do it?

Input

The first line contains 26 integers $x_a, x_b, \ldots, x_z$ ($-10^5 \le x_i \le 10^5$) — the value assigned to letters $a, b, c, \ldots, z$ respectively.

The second line contains string s of length between 1 and 10^5 characters, consisting of Lating lowercase letters— the string for which you need to calculate the answer.

Output

Print the answer to the problem.

Examples

input
1 1 -1 1 1 1 1 1 1 1 1 1 1 1 1 7 1 1 1 8 1 1 1 1 1 1 xabcab
output
2

input
1 1 -1 1 1 1 1 1 1 1 1 1 1 1 1 7 1 1 1 8 1 1 1 1 1 1 aaa
output
2

Note

In the first sample test strings satisfying the condition above are $abca$ and $bcab$.

In the second sample test strings satisfying the condition above are two occurences of aa.

E. A and B and Lecture Rooms

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

A and B are preparing themselves for programming contests.

The University where A and B study is a set of rooms connected by corridors. Overall, the University has n rooms connected by $n - 1$ corridors so that you can get from any room to any other one by moving along the corridors. The rooms are numbered from 1 to n.

Every day A and B write contests in some rooms of their university, and after each contest they gather together in the same room and discuss problems. A and B want the distance from the rooms where problems are discussed to the rooms where contests are written to be equal. The distance between two rooms is the number of edges on the shortest path between them.

As they write contests in new rooms every day, they asked you to help them find the number of possible rooms to discuss problems for each of the following m days.

Input

The first line contains integer n ($1 \le n \le 10^5$) — the number of rooms in the University.

The next $n - 1$ lines describe the corridors. The i-th of these lines ($1 \le i \le n - 1$) contains two integers a_i and b_i ($1 \le a_i, b_i \le n$), showing that the i-th corridor connects rooms a_i and b_i.

The next line contains integer m ($1 \le m \le 10^5$) — the number of queries.

Next m lines describe the queries. The j-th of these lines ($1 \le j \le m$) contains two integers x_j and y_j ($1 \le x_j, y_j \le n$) that means that on the j-th day A will write the contest in the room x_j, B will write in the room y_j.

Output

In the i-th ($1 \le i \le m$) line print the number of rooms that are equidistant from the rooms where A and B write contest on the i-th day.

Examples

input
4 1 2 1 3 2 4 1 2 3
output
1

input
4 1 2 2 3 2 4 2 1 2 1 3
output
0 2

Note

in the first sample there is only one room at the same distance from rooms number 2 and 3 — room number 1.