Control 1 - Model A - Finding Partners

A new public service is helping people to find partners for creating start-up companies. People go to this service and wait in a line to be registered. While waiting in the line, if two people work on the same field and are next to each other in the line, then they agree to collaborate as partners and leave the line.

Implement a program in C++ that, given a line of people of certain fields indicated as numbers representing their fields, determines the number of pairs of partners that find each other while waiting in the line.

For example, imagine a list of people with the following working fields represent as numbers:

1 2 2 6 7 9 9 7 1 4 4

Then, there the result would we four pairs. The pairs of field 2 find each other immediately, and the same occurs with the pair of field 9. You can see that the two people working in 7 field meet each other when the pair of 9 leaves. The pair of working 4 also finds each other.

However, even if there are two people of field 1, they don't meet as the person working in 6 is in the middle and they never match each other, even while all the pairs leave.

Input

The first line will indicate the number of cases. Each case will be defined with a list of positive numbers (representing the work field of the people in the line) separated by spaces. Each line represents the identifier of the field of each partner waiting for the line. The end of each line will be represented with "-1" number.

Output

The output of each case should be printed in one line. The output will be number of pairs that find each other while waiting in the line.

Implementation Details

In order to fasten the resolution of exercise, the virtual campus provides a template for solving exercises for cases that receive input from a list of numbers and provides a number as output. In this template, you would need to define just one function (called generically as "solveCase") avoiding to program anything concerning from reading or writing the numbers. You could also define auxiliary functions if you find it appropriate. The input is a list of integer values and the output is just one integer. This function is called generically as the same template will be used for both models of the first control. Using this template is optional, so students can either use it or not as they prefer. The linear data structures are also included in this template.

Example of input

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

Example of output

```
4
9
0
7
```