

You should use keyboard input and console output in all cases but parameter passing to functions are also acceptable. The input and output cases shown are for testing purposes only, and are not an accurate representation of the test data.

Balanced Brackets

Given a string consisting entirely of the characters `()[]{}` , determine if it is "balanced". That is, every opening bracket must have a closing bracket of the same type following it, and the string in between the pair must also be balanced. For the purposes of the problem, an empty string should be considered balanced.

	Input	Output
1	<code>()[]{}(([])){[()][]}</code>	balanced
2	<code>()[]{} </code>	not balanced
3	<code>[(])</code>	not balanced

Anagram Detection

You are given two strings, a 'parent' string and a 'query' string respectively. Your task is to determine how many times the query string – or an anagram of the query string – appears in the parent string.

NOTE: There are a range of solutions to this problem. With a little thought, you can massively improve the efficiency of your solution. The optimal solution runs almost instantly even for extremely large (1 million+ characters) parent and query strings.

	Input	Output
1	<code>AdnBndAndBdaBn dAn</code>	4
2	<code>AbrAcadAbRa cAda</code>	2
3	<code>[(])</code>	not balanced

Explanation for input-1

The substrings are highlighted below.

AdnBndAndBdaBn

AdnBndAndBdaBn

AdnBndAndBdaBn

AdnBndAndBdaBn

CSV Parsing

The input to this problem consists of a string of n comma-separated values, each value being an integer **or a string**. The required output is n consecutive lines, where line i contains the i th value of the input.

NOTICE – string may contain commas (See **Input 2 and 3** below).

	Input	Output
1	2,6,3,2,5	2 6 3 2 5
2	"pears","apples","walnuts","grapes","cheese,cake"	"pears" "apples" "walnuts" "grapes" "cheese,cake"
3	1,"Que?","Kay?",2,"Si.,""Sea? Kay, sea?","No, no, no. Que... 'what'."",234,"Kay Watt?","Si, que 'what'."", "C.K. Watt?",3,"Yes!","comma,comma, comma , :)"	1 "Que?" "Kay?" 2 "Si." "Sea? Kay, sea?" "No, no, no. Que... 'what'." 234 "Kay Watt?" "Si, que 'what'."\ "C.K. Watt?" 3 "Yes!" "comma,comma, comma , :)"

Spiral

You are given the dimension of a $h \times w$ grid filled with consecutive integers from left to right, top to bottom, starting with 1.

You are also given a starting position r c . The output should be the ordered list of integers obtained by spiralling outward in an anti-clockwise direction from row r column c , starting upward.

Input 1

5 5 3 3

Output 1

13 8 7 12 17 18 19 14 9 4 3 2 1 6 11 16 21 22 23 24 25 20 15 10 5

Explanation 1

The following graphics show the grid in question and the spiral generated, beginning at cell (3,3).

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Input 2

2 4 1 2

Output 2

2 1 5 6 7 3 8 4

Explanation 2

1	2	3	4
5	6	7	8

1	2	3	4
5	6	7	8

Time Confusion

There are three watches, each giving a different time. One watch is x minutes behind the actual time. One watch is x minutes ahead of the actual time. From the time displayed on each watch, determine the actual time. If it is not possible, print "Look at the sun".

Input:

The input begins with an integer T indicating the number of cases.

Each of the following T lines contains one test case, made up of three readings, separated by single space characters: $H1:M1$ $H2:M2$ $H3:M3$

In each reading $H1, H2, H3$ represent the hours displayed ($0 < H1, H2, H3 < 13$), and $M1, M2, M3$ represent the minutes displayed ($0 < M1, M2, M3 < 60$). If the number of minutes is less than 10, a leading 0 is prepended.

Input 1:

```
3
5:00 12:00 10:00
11:59 12:30 1:01
12:00 4:00 8:00
```

Output 1:

```
The correct time is 5:00.
The correct time is 12:30.
Look at the sun.
```