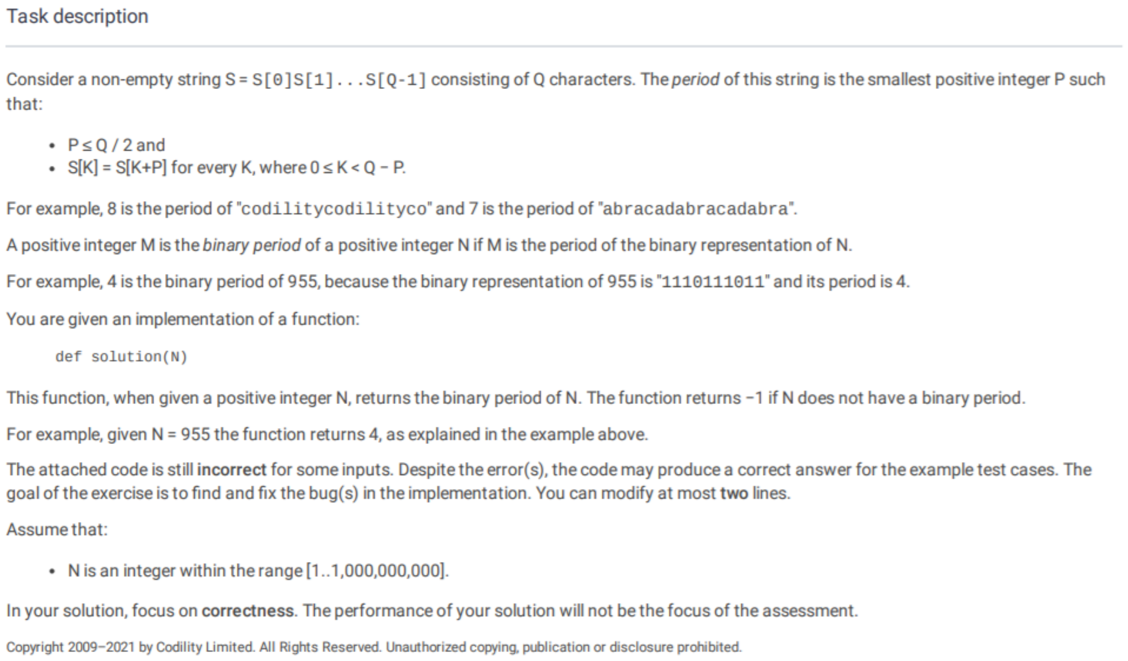
1.-BugxingBinaryPeriod

Encontrar el periodo de un string (Eduardo Gómez,)



Task description

Consider a non-empty string S = `S[0]S[1]...S[Q-1]` consisting of Q characters. The *period* of this string is the smallest positive integer P such that:

- P ≤ Q / 2 and
- S[K] = S[K+P] for every K, where 0 ≤ K < Q − P.

For example, 8 is the period of "`codilitycodilityco`" and 7 is the period of "`abracadabracadabra`".

A positive integer M is the *binary period* of a positive integer N if M is the period of the binary representation of N.

For example, 4 is the binary period of 955, because the binary representation of 955 is "`1110111011`" and its period is 4.

You are given an implementation of a function:

```
def solution(N)
```

This function, when given a positive integer N, returns the binary period of N. The function returns −1 if N does not have a binary period.

For example, given N = 955 the function returns 4, as explained in the example above.

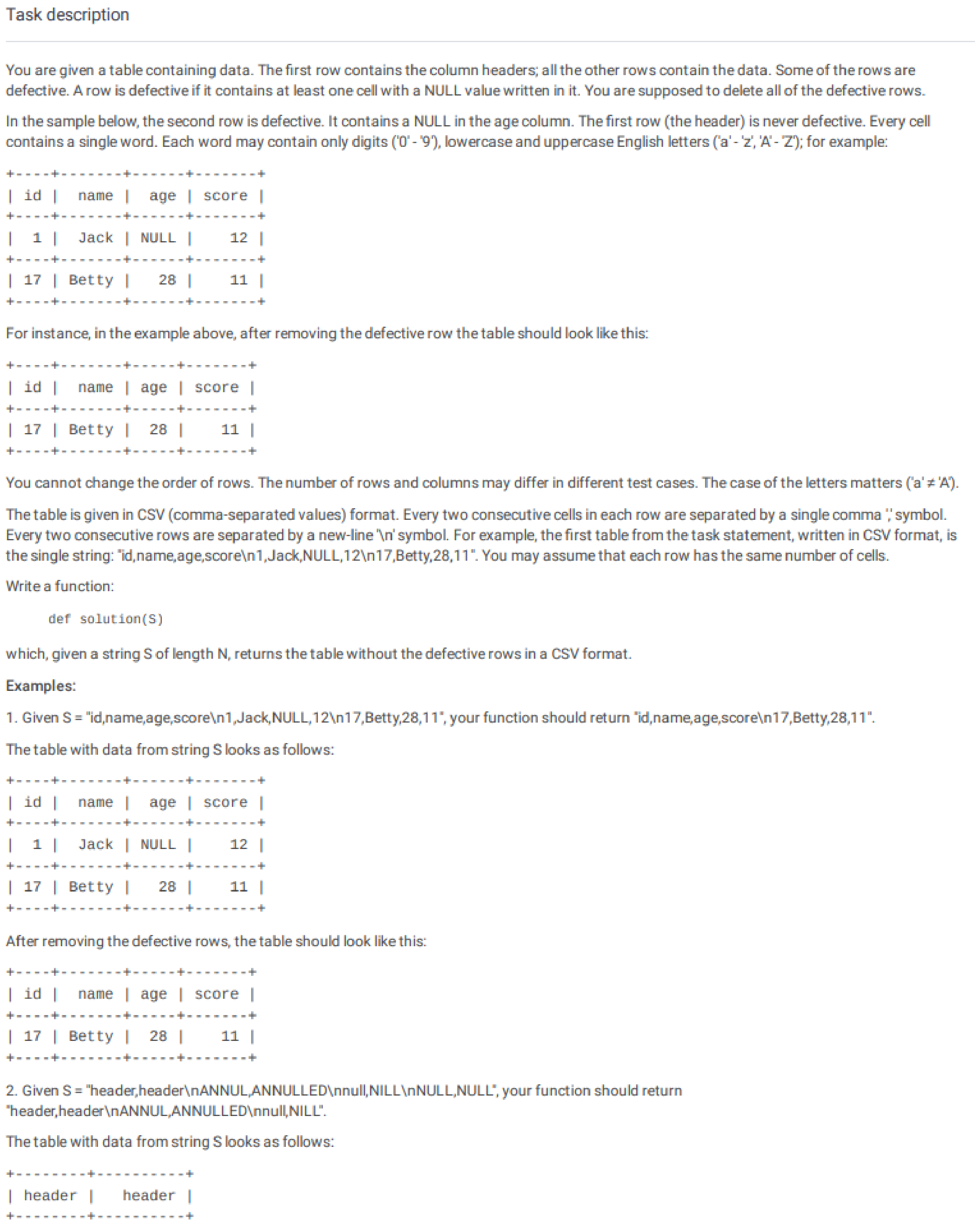
The attached code is still **incorrect** for some inputs. Despite the error(s), the code may produce a correct answer for the example test cases. The goal of the exercise is to find and fix the bug(s) in the implementation. You can modify at most **two** lines.

Assume that:

- N is an integer within the range [1..1,000,000,000].

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

2.-DeleteDefectiveRows
(Rafael)



Task description

You are given a table containing data. The first row contains the column headers; all the other rows contain the data. Some of the rows are defective. A row is defective if it contains at least one cell with a NULL value written in it. You are supposed to delete all of the defective rows.

In the sample below, the second row is defective. It contains a NULL in the age column. The first row (the header) is never defective. Every cell contains a single word. Each word may contain only digits ('0' - '9'), lowercase and uppercase English letters ('a' - 'z', 'A' - 'Z'); for example:

```
+----+-------+------+-------+
| id |  name |  age | score |
+----+-------+------+-------+
|  1 |  Jack | NULL |    12 |
+----+-------+------+-------+
| 17 | Betty |   28 |    11 |
+----+-------+------+-------+
```

For instance, in the example above, after removing the defective row the table should look like this:

```
+----+-------+-----+-------+
| id |  name | age | score |
+----+-------+-----+-------+
| 17 | Betty |  28 |    11 |
+----+-------+-----+-------+
```

You cannot change the order of rows. The number of rows and columns may differ in different test cases. The case of the letters matters ('a' ≠ 'A').

The table is given in CSV (comma-separated values) format. Every two consecutive cells in each row are separated by a single comma ',' symbol. Every two consecutive rows are separated by a new-line '\n' symbol. For example, the first table from the task statement, written in CSV format, is the single string: "id,name,age,score\n1,Jack,NULL,12\n17,Betty,28,11". You may assume that each row has the same number of cells.

Write a function:

```
    def solution(S)
```

which, given a string S of length N, returns the table without the defective rows in a CSV format.

Examples:

1. Given S = "id,name,age,score\n1,Jack,NULL,12\n17,Betty,28,11", your function should return "id,name,age,score\n17,Betty,28,11".

The table with data from string S looks as follows:

```
+----+-------+------+-------+
| id |  name |  age | score |
+----+-------+------+-------+
|  1 |  Jack | NULL |    12 |
+----+-------+------+-------+
| 17 | Betty |   28 |    11 |
+----+-------+------+-------+
```

After removing the defective rows, the table should look like this:

```
+----+-------+-----+-------+
| id |  name | age | score |
+----+-------+-----+-------+
| 17 | Betty |  28 |    11 |
+----+-------+-----+-------+
```

2. Given S = "header,header\nANNUL,ANNULLED\nnull,NILL\nNULL,NULL", your function should return "header,header\nANNUL,ANNULLED\nnull,NILL".

The table with data from string S looks as follows:

```
+--------+----------+
| header |   header |
+--------+----------+
|  ANNUL | ANNULLED |
+--------+----------+
|   null |     NILL |
+--------+----------+
|   NULL |     NULL |
+--------+----------+
```

After removing the defective rows, the table should look like this:

```
+--------+----------+
| header |   header |
+--------+----------+
|  ANNUL | ANNULLED |
+--------+----------+
|   null |     NILL |
+--------+----------+
```

The second row is not defective, because the "null" is written in lowercase.

3. Given S = "country,population,area\nUK,67m,242000km2", your function should return "country,population,area\nUK,67m,242000km2".

The table with data from string S looks as follows:

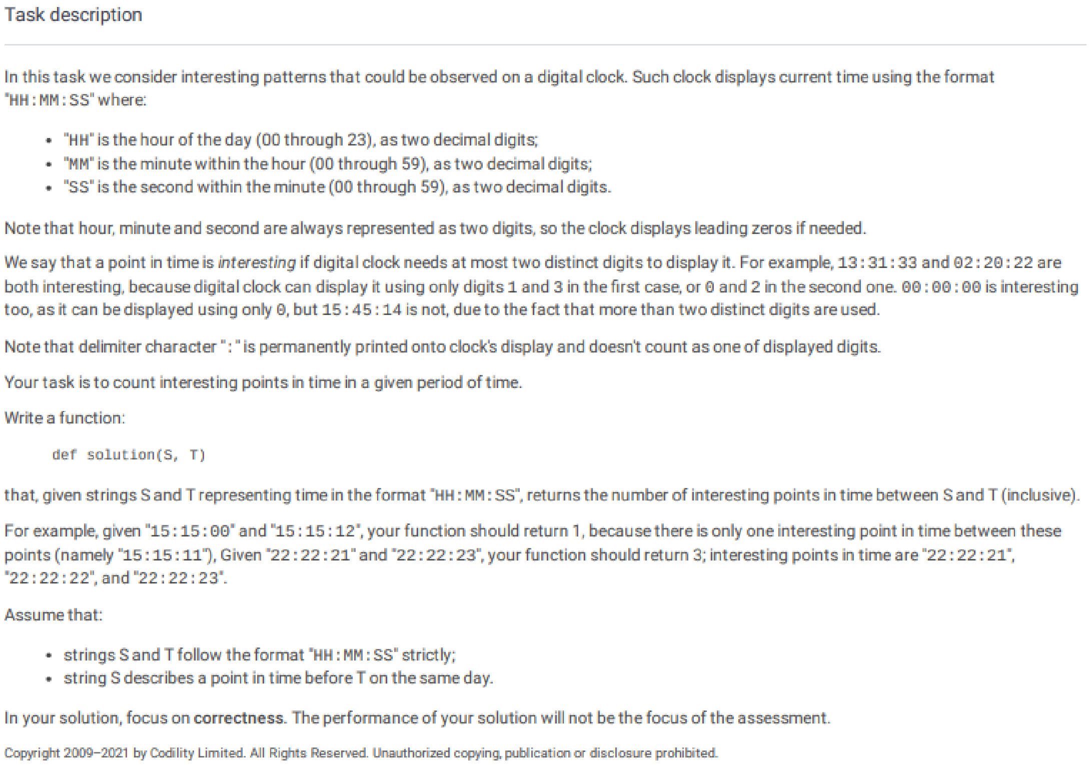
```
+---------+------------+-----------+
| country | population |      area |
+---------+------------+-----------+
|      UK |        67m | 242000km2 |
+---------+------------+-----------+
```

There are no defective rows in this table, hence after removing defective rows the table looks the same.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [1..200,000];
- S is a string of length N in CSV format;
- there is at least one row;
- each row has the same, positive number of cells;
- each cell consists only of letters and/or digits;
- the first row does not contain a NULL cell.

3.-TwoDigitHours
Jose Luis



Task description

In this task we consider interesting patterns that could be observed on a digital clock. Such clock displays current time using the format "`HH:MM:SS`" where:

- "`HH`" is the hour of the day (00 through 23), as two decimal digits;
- "`MM`" is the minute within the hour (00 through 59), as two decimal digits;
- "`SS`" is the second within the minute (00 through 59), as two decimal digits.

Note that hour, minute and second are always represented as two digits, so the clock displays leading zeros if needed.

We say that a point in time is *interesting* if digital clock needs at most two distinct digits to display it. For example, `13:31:33` and `02:20:22` are both interesting, because digital clock can display it using only digits `1` and `3` in the first case, or `0` and `2` in the second one. `00:00:00` is interesting too, as it can be displayed using only `0`, but `15:45:14` is not, due to the fact that more than two distinct digits are used.

Note that delimiter character "`:`" is permanently printed onto clock's display and doesn't count as one of displayed digits.

Your task is to count interesting points in time in a given period of time.

Write a function:

```
    def solution(S, T)
```

that, given strings S and T representing time in the format "`HH:MM:SS`", returns the number of interesting points in time between S and T (inclusive).

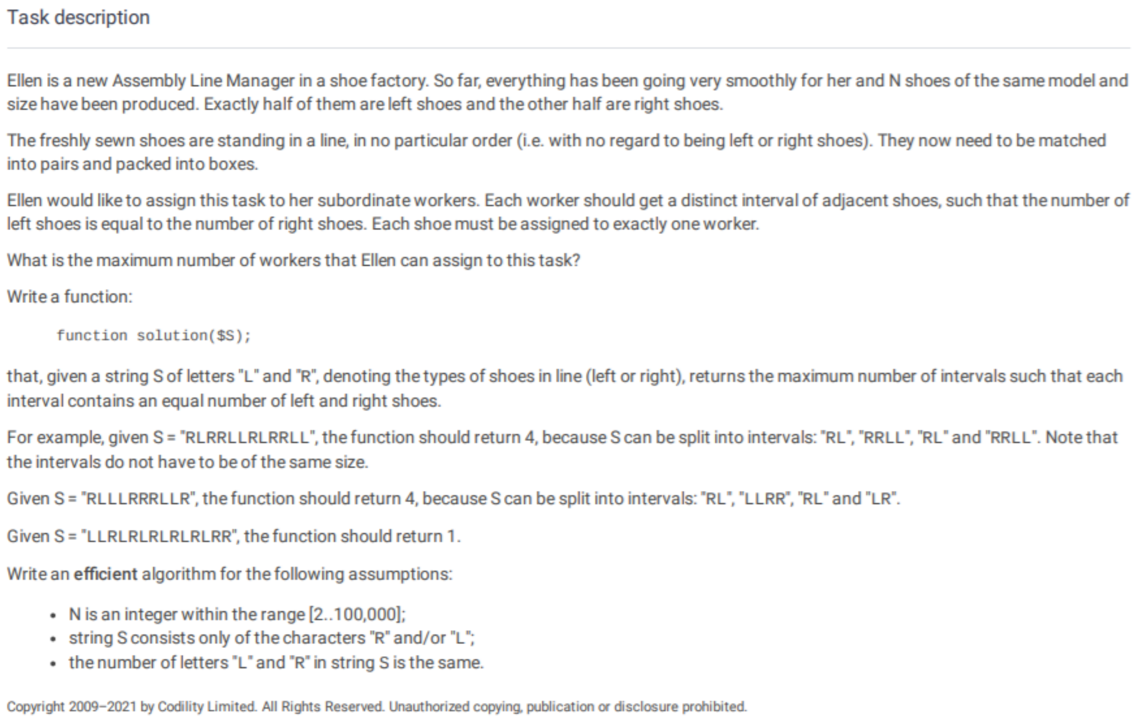
For example, given "`15:15:00`" and "`15:15:12`", your function should return 1, because there is only one interesting point in time between these points (namely "`15:15:11`"), Given "`22:22:21`" and "`22:22:23`", your function should return 3; interesting points in time are "`22:22:21`", "`22:22:22`", and "`22:22:23`".

Assume that:

- strings S and T follow the format "`HH:MM:SS`" strictly;
- string S describes a point in time before T on the same day.

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

4.-Shoe Factory
(Erick David)



Task description

Ellen is a new Assembly Line Manager in a shoe factory. So far, everything has been going very smoothly for her and N shoes of the same model and size have been produced. Exactly half of them are left shoes and the other half are right shoes.

The freshly sewn shoes are standing in a line, in no particular order (i.e. with no regard to being left or right shoes). They now need to be matched into pairs and packed into boxes.

Ellen would like to assign this task to her subordinate workers. Each worker should get a distinct interval of adjacent shoes, such that the number of left shoes is equal to the number of right shoes. Each shoe must be assigned to exactly one worker.

What is the maximum number of workers that Ellen can assign to this task?

Write a function:

```
    function solution($S);
```

that, given a string S of letters "L" and "R", denoting the types of shoes in line (left or right), returns the maximum number of intervals such that each interval contains an equal number of left and right shoes.

For example, given S = "RLRRLLRLRRLL", the function should return 4, because S can be split into intervals: "RL", "RRLL", "RL" and "RRLL". Note that the intervals do not have to be of the same size.

Given S = "RLLLRRRLLR", the function should return 4, because S can be split into intervals: "RL", "LLRR", "RL" and "LR".

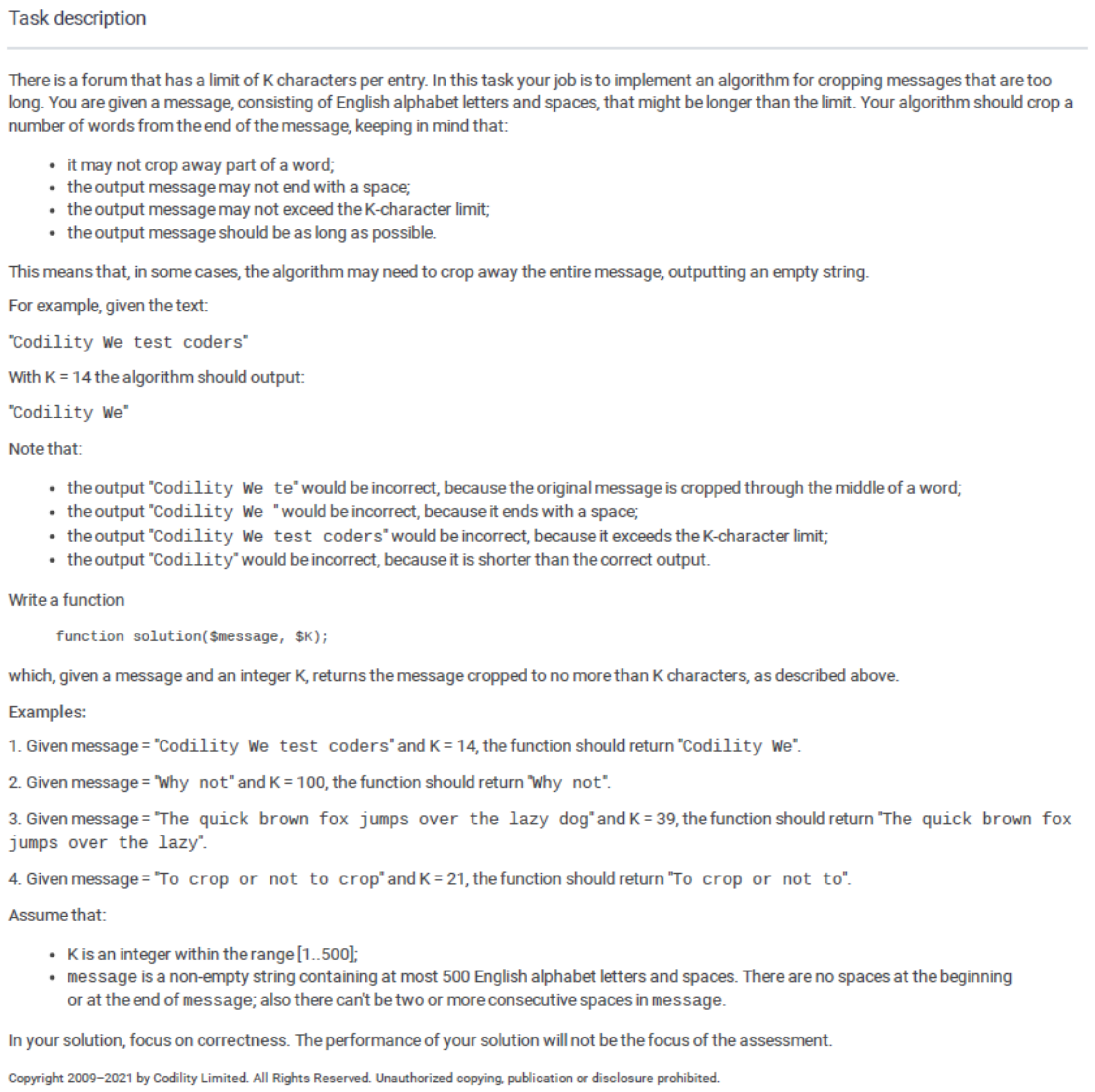
Given S = "LLRLRLRLRLRLRR", the function should return 1.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [2..100,000];
- string S consists only of the characters "R" and/or "L";
- the number of letters "L" and "R" in string S is the same.

5.-Crop the message

Omar



Task description

There is a forum that has a limit of K characters per entry. In this task your job is to implement an algorithm for cropping messages that are too long. You are given a message, consisting of English alphabet letters and spaces, that might be longer than the limit. Your algorithm should crop a number of words from the end of the message, keeping in mind that:

- it may not crop away part of a word;
- the output message may not end with a space;
- the output message may not exceed the K-character limit;
- the output message should be as long as possible.

This means that, in some cases, the algorithm may need to crop away the entire message, outputting an empty string.

For example, given the text:

"`Codility We test coders`"

With K = 14 the algorithm should output:

"`Codility We`"

Note that:

- the output "`Codility We te`" would be incorrect, because the original message is cropped through the middle of a word;
- the output "`Codility We `" would be incorrect, because it ends with a space;
- the output "`Codility We test coders`" would be incorrect, because it exceeds the K-character limit;
- the output "`Codility`" would be incorrect, because it is shorter than the correct output.

Write a function

```
    function solution($message, $K);
```

which, given a message and an integer K, returns the message cropped to no more than K characters, as described above.

Examples:

1. Given message = "`Codility We test coders`" and K = 14, the function should return "`Codility We`".

2. Given message = "`Why not`" and K = 100, the function should return "`Why not`".

3. Given message = "`The quick brown fox jumps over the lazy dog`" and K = 39, the function should return "`The quick brown fox jumps over the lazy`".

4. Given message = "`To crop or not to crop`" and K = 21, the function should return "`To crop or not to`".

Assume that:

- K is an integer within the range [1..500];
- `message` is a non-empty string containing at most 500 English alphabet letters and spaces. There are no spaces at the beginning or at the end of `message`; also there can't be two or more consecutive spaces in `message`.

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.