

Estimated time

15-30 minutes

Level of difficulty

Easy

Objectives

- improving the student's skills in operating with strings;
- encouraging the student to look for non-obvious solutions.

Scenario

Do you know what a palindrome is?

It's a word which look the same when read forward and backward. For example, "kayak" is a palindrome, while "loyal" is not.

Your task is to write a program which:

- asks the user for some text;
- checks whether the entered text is a palindrome, and prints result.

Note:

- assume that an empty string isn't a palindrome;
- treat upper- and lower-case letters as equal;
- spaces are not taken into account during the check - treat them as non-existent;
- there are more than a few correct solutions - try to find more than one.

Test your code using the data we've provided.

Test data

Sample input:

```
Ten animals I slam in a net
```

Sample output:

```
It's a palindrome
```

Sample input:

```
Eleven animals I slam in a net
```

Sample output:

```
It's not a palindrome
```

Estimated time

30-60 minutes

Level of difficulty

Easy

Objectives

- improving the student's skills in operating with strings;
- converting strings into lists, and vice versa.

Scenario

An anagram is a new word formed by rearranging the letters of a word, using all the original letters exactly once. For example, the phrases "rail safety" and "fairy tales" are anagrams, while "I am" and "You are" are not.

Your task is to write a program which:

- asks the user for two separate texts;
- checks whether, the entered texts are anagrams and prints the result.

Note:

- assume that two empty strings are not anagrams;
- treat upper- and lower-case letters as equal;
- spaces are not taken into account during the check - treat them as non-existent

Test your code using the data we've provided.

Test data

Sample input:

Listen
Silent

Sample output:

Anagrams

Sample input:

modern
norman

Sample output:

Not anagrams

Estimated time

30-45 minutes

Level of difficulty

Medium

Objectives

- improving the student's skills in operating with strings;
- using the `find()` method for searching strings.

Scenario

Let's play a game. We will give you two strings: one being a word (e.g., "dog") and the second being a combination of any characters.

Your task is to write a program which answers the following question: **are the characters comprising the first string hidden inside the second string?**

For example:

- if the second string is given as "vcxzxduybfdsobywuefgas", the answer is `yes`;
- if the second string is "vcxzxdcybfdstbywuefsas", the answer is `no` (as there are neither the letters "d", "o", or "g", in this order)

Hints:

- you should use the two-argument variants of the `pos()` functions inside your code;
- don't worry about case sensitivity.

Test your code using the data we've provided.

Test data

Sample input:

`donor`

`Nabucodonosor`

Sample output:

Yes

Sample input:

donut

Nabucodonosor

Sample output:

No

