# Assignment - 2

Deadline: **23.12.25**

- Each task in its own .py file **(1 points)**
- All completed task files must be pushed to the designated GitHub repository. **(1 points)**
- Assessment**:** Successful submission requires both the **GitHub push** and a subsequent **oral defense** of the work.

**Task 1 (1points)**

You are given a sequence consisting only of the characters '>', '<', and '-'. Your goal is to find the number of arrows hidden in this sequence. Arrows are substrings of the form '>>-->' and '<--<<'.

Input Data contains a string consisting of the characters '>', '<', and '-' (without spaces). The string must be no more than 250 characters long.

Output Data

The desired number of arrows must be printed on a single line

| Input | Output |
|---|---|
| <<<<>>--><--<<--<<>>>-- ><<<<< | 4 |

**Task 2 (2 points)**

A cyclic shift of a string s is the string $s_k s_{k+1} s_{k+2} \dots s_{|s|} s_1 s_2 \dots s_{k-1}$ for some k, where |s| is the length of s. A substring of s is the string $s_i s_{i+1} \dots s_{j-1} s_j$ for some i and j. You are given two strings a and b. Print the number of substrings of a that are cyclic shifts of b.

Input Data

The first line of the input contains the string a ($1 \le |a| \le 1000$). The second line of the input filcontains the string b ($1 \le |b| \le \min(100,|a|)$). Both strings consist only of English alphabetic characters and digits.

Output Data

Print an integer—the answer to the problem

| Input | Output |
|---|---|
| abcabc<br>abc | 4 |
| abcabc<br>acb | 0 |
| aaaaaaa<br>aa | 6 |
| aAaa8aaAa<br>aAa | 4 |

**Task 3(1 points)**

The equation for fifth-graders is a string five characters long. The second character is either a '+' (plus) or '-' (minus), and the fourth character is an '=' (equal) sign. Of the first, third, and fifth characters, exactly two are digits from 0 to 9, and one is the letter x, representing the unknown.

You are required to write a program that will solve this equation for x.

| Input | Output |
|---|---|
| **x+5=7** | **2** |
| **3-x=9** | **-6** |

**Task 4 (2 points)**

At a very famous university, a very famous professor delivered his lectures so quickly that it was impossible to understand anything. Recently, a student named

Willy decided to conduct a study of the professor's vocabulary. To this end, he even attended one lecture and recorded everything said on a tape recorder. Then, by playing the recording back at home at ten times the speed of the tape, Willy was able to transcribe everything the professor said.

But here's the problem: the professor spoke so quickly that even listening to the slowed-down recording, it was impossible to tell exactly where he paused between words. So, Willy has a text consisting of n lowercase English letters he lecture the professor delivered. Now Willy wants to know how many different words of length m the professor could have used in his lecture.

Input Data

The first line contains two numbers, n and m ($1 \leq m \leq n \leq 100$), representing the lecture length and the word length. The second line contains n English characters the text of the professor's lecture.

Output Data

In the output print a single number the number of words of length m that the professor could have used in their lecture.

| Input | Output |
|---|---|
| 3 1<br>abc | **3** |
| 10 3<br>bbaabbbabb | **6** |

**Task 5 (2 points)**

One day, Vasya spent a long time waiting at a bus stop before his bus arrived. To pass the time, he decided to write down the license plates of passing buses on a piece of paper, taking a route different from the one he needed. He only wrote down the main license plate number, ignoring regional affiliation. Ultimately, Vasya managed to write down N such license plates.

The main part of a license plate number consists of six characters: three letters and three numbers. A letter begins, then three numbers, and two more letters complete

the entry. Any digit from 0 to 9 can be used as numbers, and only uppercase letters whose symbols are present in both the English and Russian alphabets can be used as letters, i.e., only the following characters: A, B, C, E, H, K, M, O, P, T, X, and Y. For example, "P204BT" is a valid license plate number, while "X182YZ" and "ABC216" are not.

Your task is to verify the accuracy of Vasya's work. Specifically, it is necessary to determine which numbers comply with the accepted standard and which do not.

Input Data

The first line of the input contains a single natural number N—the number of license plates Vasya recorded ($N \leq 50$). This is followed by N lines of bus number entries. Line lengths range from 1 to 300 and contain only characters with ASCII codes from 33 to 127 (excluding spaces, special characters, and Russian characters).

Output Data

Print N lines. The i-th line should contain "Yes" if the corresponding i-th license plate entry is correct and "No" otherwise.

| Input | Output |
|---|---|
| 5<br>P204BT<br>X182YZ<br>a216bc<br>A216BC<br>ABC216 | Yes<br>No<br>No<br>Yes<br>No |

**Task 6 (1 points)**

Condition
At the input we have a list of strings of different lengths.
It is necessary to write the all_eq(list) function, which will return a new list of strings of the same length.
The length of the final line is determined based on the largest of them.
If a particular line is shorter than the longest, add underscores from the right edge to the required number of characters.

Do not change the location of the elements of the initial list.
First you need to determine the length of each row in the list and find the maximum. Next, we add the characters "_" to the strings whose length is less.

**Task 7**: **(3 points)**

Shopping List Analysis (Collections)

A store recorded a list of items purchased by customers during the day.
Each purchase is represented by a string (the item name).

You need to:

1. Count how many times each item was purchased
2. Find the most frequently purchased item
3. Output all items that were purchased exactly once
4. Sort the items by descending purchase count

Input
apple banana apple orange banana apple milk

Output
Purchase frequency:
apple: 3
banana: 2
orange: 1
milk: 1

Most popular item: apple

Purchased once: orange milk

Sorted by frequency:
apple 3
banana 2
orange 1
milk 1

**Task 8 (1 points)**

String S1 is called an anagram of string S2 if it is obtained from S2 by rearranging its characters. Given strings S1 and S2, write a program that checks whether S1 is an anagram of S2.

Input
The first line contains the string S1, and the second line contains S2. Both lines consist only of uppercase English letters. The lines are not empty and have a length of no more than 100,000 characters.

Output
In the output fprint YES if S1 is an anagram of S2, and NO otherwise.

| Input | Output |
|---|---|
| ABAA<br>ABBA | NO |
| ABBA<br>BABA | YES |