Session 7 worksheet

Select a group

In Moodle, please register to the group with the same label as is written on your card. Make sure that you and your partner are in the same group.

1. Username from URL

Explain your solution of the username homework exercise \((home2.py)\) to your partner. You probably had slightly different approaches to finding the username. In what aspects do your approaches differ?

What do you think, which of your versions is better?

Explain, why?

Based on your comparison or programs and general knowledge, in your opinion, what would be the best method to extract the username from an URL?

2. String operations

Look at your homework as whole (programs \(home1.py\) and \(home2.py\) for both of you). Write down all string-related operations you find in your programs.

Please discuss with your partner and find one good example of a programming problem that can be solved with extensive help of strings.
3. Euros and cents

Write a program that prompts for a number of cents and prints the same amount in euros and cents or only in euros or only in cents. If the amount of euros or cents is zero, then the program should not print that part of the answer. The program should use correct word forms in singular or plural.

Enter amount: 207
2 euros and 7 cents
Enter amount: 101
1 euro and 1 cent
Enter amount: 95
95 cents
Enter amount: 100
1 euro

*Hint*: integer division is //, remainder is %.

4. Balanced brackets

The user enters a string that can contain any number of brackets ‘(‘ and ‘)’. Write a program that checks whether the brackets are balanced in the string.

Enter string: large (in fact, huge) waves
Brackets are balanced
Enter string: 1+(5-(3+4)*3)**4
Brackets are not balanced
Enter string: print("Result is " + calc(calc(x, x+y), calc(x+y, y))) + " km."
Brackets are not balanced
Enter string: (((())))((()))
Brackets are not balanced

*Hint*: Define a variable that keeps track of current nesting depth. For each opening bracket increase its value by one, for each closing bracket decrease by one.

Next, extend the program to also accept the brackets ‘{‘, ‘}’ and ‘[‘, ‘]’. All brackets can appear inside each other and closing bracket should always be at the correct nesting level.

Enter string: {{[]}}
Brackets are not balanced
Enter string: [()]{}{[()()]()}
Brackets are balanced

5. Rhymes

While writing poetry, often the need arises to find a good word that rhymes with the word at the end of the previous line. Write a program that allows the user to test different rhyme-words of a given word and select the best. The words can be in any language.

*Easier version*. Let the rhyme score of two words be the maximum number of consecutive letters, starting from the end, that are the same in both words. The user first enters a word and then starts to enter words
that rhyme with the first word. For each rhyme-word the program prints its rhyme score with respect to the first word. If the user enters 'done', then the program prints out the word that had the largest rhyme score.

Enter word: program
Enter rhyme-word: tram
Rhyme score is 3
Enter rhyme-word: kilogram
Rhyme score is 5
Enter rhyme-word: wolfram
Rhyme score is 3
Enter rhyme-word: telegram
Rhyme score is 4
Enter rhyme-word: done
The word with the highest rhyme score was 'kilogram', its score was 5.

Harder version. A vowel group is a set of one or more consecutive vowels that are surrounded by consonants or word boundary. Let the rhyme score of two words be the number of consecutive vowel groups, starting from the end, that coincide in both words. The user enters a word and then starts entering different rhyme-words; the program prints the rhyme score for each rhyme-word and, after entering 'done', prints out the word with the largest rhyme score.

Enter word: behave
Enter rhyme-word: caramel
Rhyme score is 2
Enter rhyme-word: tetrahedral
Rhyme score is 0
Enter rhyme-word: teasers
Rhyme score is 1
Enter rhyme-word: debate
Rhyme score is 3
Enter rhyme-word: reggae
Rhyme score is 0
Enter rhyme-word: done
The word with the highest rhyme score was 'debate', its score was 3.