ADVANCED PYTHON HOMEWORK 1

strings and doubles

The deadline of the homework as well as its submission is handled centrally via SKOS. Please make sure you are submitting through that.

Every exercise is worth 2 points. The Polish homework also states that "na pracowni do oceny należy przedstawić trzy zadania". Once I know the correct translation (and interpretation) of this sentence, I will update this sheet.

This homework sheet is equivalent to the Polish one; if you do not understand anything, feel free to ask your lab teacher or consult the Polish sheet. Note that it is possible that an exercise may be intentionally vague or discuss "reasonable" amount of support; in that case, it may be part of the task to find the right interpretation.

Exercise 1. In Poland the VAT may be counted in two different ways: in the case of invoices, you sum up the total net cost and multiply it by 23%, but in the case of receipts, one adds the 23% separately for each purchased item's net cost and then sums up the total cost. Your task is to program two functions in Python, `vat_invoice(list)` (for the invoices) and `vat_receipt(list)` (for the receipts). In both cases, the `list` is a list of the `float` net costs for each item in the bill. The return value should be one `float`.

Will these functions return the same value on all inputs? Form a hypothesis and provide some experimental results for your theory, testing e.g. the following:

```
sales = [0.2, 0.5, 4.59, 6]
print(vat_invoice(sales) == vat_receipt(sales))
```

Finally, in the documentation/comments, discuss if replacing `float` by `Decimal` changes the outcome of your hypothesis.

Exercise 2. Implement a function `is_palindrome(text)` which returns `True` if the argument is a palindromic string or sentence. Note that for this exercise we ignore spaces, capital letters and punctuation marks, so a palindrome can be a single word such as *rotor*, but also a longer sentence such as *Kobyła ma mały bok.*

Make sure that your function supports multiple languages (to a reasonable extent):

```
is_palindrome("Eine güldne, gute Tugend: Lüge nie!")
```

Exercise 3. Oct 1, 2021 was the World Multiplication Table Day. *(Translator's note: It is celebrated mainly within Polish elementary schools and promoted by a Polish company. You may want to Google it.)*

Implement a function `multi_table(x1, x2, y1, y2)`, which prints to the standard output the multiplication tables of numbers in the interval $[x_1, \dots, x_2] \times [y_1, \dots, y_2]$. For example, the command `multi_table(3,5,2,4)` should print out

```
   3  4  5
2  6  8 10
3  9 12 15
4 12 16 20
```

Make sure that the table looks nice, the columns are all right-aligned and spacing is chosen based on the digit length of the integers within.

Exercise 4. Implement an experiment about throwing coins. In this experiment, we throw a fair coin until we get the same face of a coin three times in a row. The experiment should take as a parameter the number of runs (repetitions) and print out an estimate of the expected number of throws until a single run stops. The experiment should be implemented so that we can also adjust the number of the same face until the experiment stops.

Exercise 5. Implement a function which, given a list of strings `list_of_words`, finds the longest common prefix shared by at least three strings inside the list. For example, the following call:

```
common_prefix(["Cyprian", "cyberotoman", "cynik", "ceniąc", "czule"])
```

should return `"cy"`.

Again, you can see that non-ASCII characters are to be supported, and we also ignore upper-case of characters.

Translator's note: The input strings are a Polish reference, again possibly worth Googling.