October 8, 2024

Python Advanced Course

List 1.

Each task is worth 2 points. Three tasks must be presented for assessment in the lab.

Task 1.

In Poland, the goods and services tax (VAT) is calculated in two ways: in the case of invoices, the net values are summed up and multiplied by 23%, and in the case of cash registers and receipts, the VAT 23% is calculated for each item separately and added up at the end. Program two functions in Python that return the VAT for a given shopping list

- vat_invoice(list)
- vat_receipt(list)

where list is a list of numbers representing the net price. We usually expect the following program to print True

```
print(vat_invoice(shopping) == vat_receipt(shopping))
```

where purchases is a list of float numbers.

Find a shopping list for which the above program prints False and place this list in the source file. Do an experiment by changing the float numbers in the shopping list to their Decimal class equivalents and checking if the program still prints False in the frame.

Task 2.

Write a function is_palindrome(text) that returns True if the argument is a palindrome. We assume that text can be either a single word (e.g. rotor or eye), or a longer expression: "The mare has a small side."; in such a case we ignore punctuation marks, spaces, and case.

Check if the function works correctly for foreign language texts:

```
is_palindrom("Eine güldne, gute Tugend: Lüge nie!") is_palindrom("Míý
omoýím.")
```

Task 3.

November 15th will be World Multiplication Table Day. Program the function table(x1, x2, y1, y2, d), which will print to the screen the multiplication table for the numbers [x1, x1 + d, x1 + 2 ÿ d, . . . , x2] × [y1, y1 + d, y1 + 2 ÿ d, . . . , y2], where x1, x2, y1, y2 and d are float numbers.

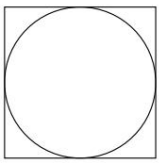
For example, array(3.0, 5.0, 2.0, 4.0, 1.0) should print

```
      3.0 4.0 5.0
2.0 6.0 8.0 10.0
3.0 9.0 12.0 15.0
4.0 12.0 16.0 20.0
```

Make sure that the column widths are the same and appropriate to the number of digits in the numbers. We assume that x1, x2, y1, y2 can also be negative numbers.

Task 4.

The number ÿ (or rather its successive approximations) can be calculated in many ways.
One of them is to throw a dart multiple times at a square target with a circle inscribed in it:



and counting the number of hits inside the circle (ltwo) and the total number of hits on the target (cltwt). These two numbers will allow us to calculate an approximation of ÿ:

$$\ddot{y}\,\ddot{y} \quad \frac{4\ \ddot{y}\ \text{ltwo}}{\text{cltwt}}$$

Program a simulation of throwing a dart at a target by drawing the coordinates of a point on the target. The program should print the obtained approximations of ÿ after each throw. The program can end after a specified number of draws or when the difference between the obtained approximation and the value of math.pi is less than a specified value.

Task 5.

Program a function that for a given list of strings list_slow returns the longest common prefix for at least three elements of list_slow. For example1

```
common_prefix(["Cyprian", "cyber-ottoman", "cynic", "appreciating", "tenderly"])
```

should return

```
"cy"
```

The size of letters does not matter to us.

Marcin Mÿotkowski

[1]Inspiration: Cyberiad, Stanisÿaw Lem