Programming in Python I, Intersession
Jacobs University Bremen
Dr. Kinga Lipskoch

Course: JTSK-350111
January 31st, 2018

Assignment 6 - Graphics, Dictionaries and Tuples

- The problems of this assignment must be solved in Python.
- The TAs are grading solutions to the problems according to the following criteria: https://grader.eecs.jacobs-university.de/courses/350111/2017_2is/Grading-Criteria-Python.pdf

Problem 6.1 *A square* (1 point)

Presence assignment, due by 18:30 h today

Write a program where you enter from the keyboard an integer value representing the side of a square. Then the square with that side is drawn to the middle of the screen. Use the following method call to set the size of the screen to 600 pixels $\times$ 600 pixels:

```
t.screen.setup(width=600, height=600).
```

You can assume that the input will be valid and the screen will not be resized.

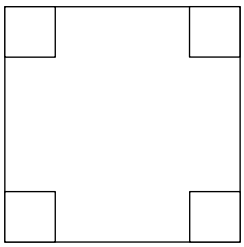
Problem 6.2 *Squares* (1 point)

Presence assignment, due by 18:30 h today

Write a program where you enter from the keyboard an integer value representing the side of a square. Then one large square of some fixed size (you are free to select the exact side size) and four green squares with the side entered from the keyboard are drawn to each corner of the fixed square. Use a function to draw one square and call it five times. Use the following method call to set the size of the screen to 700 pixels $\times$ 700 pixels:

```
t.screen.setup(width=700, height=700).
```

The result should approximately look like in the following figure (using green as color):



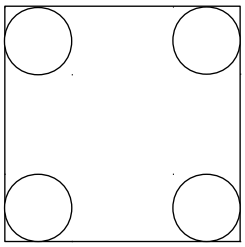
You can assume that the input will be valid and the screen will not be resized.

Problem 6.3 *Circles* (1 point)

Write a program where you enter from the keyboard the radius of a circle. Then one large square of fixed size (you are free to select the exact side size) and four orange circles with that radius are drawn to each corner of the fixed square. Use the function `circle()` to draw one circle. Use the following method call to set the size of the screen to 700 pixels $\times$ 700 pixels:

```
t.screen.setup(width=700, height=700).
```

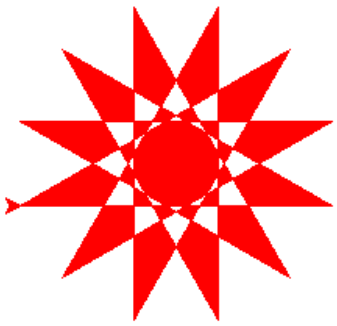
The result should approximately look like in the following figure (using orange as color):



You can assume that the input will be valid and the screen will not be resized.

Problem 6.4 *A star* (1 point)

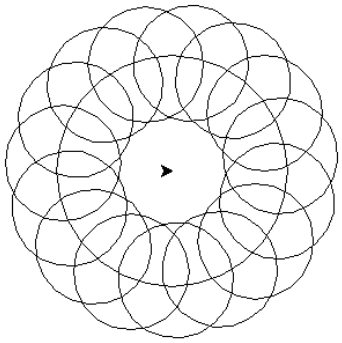
Write a program where you draw a star "similar" to the one below. You can freely decide about the color, size, position and exact shape. Use the following method call to set the size of the screen to 600 pixels × 600 pixels:
`t.screen.setup(width=600, height=600).`



You can use for example the following functions: `color()`,`begin_fill()`,`end_fill()`,`forward()`, `left()`,`position()`.

Bonus Problem 6.5 *Flower of circles* (1 point)

Write a program where you draw a figure "similar" to the one below composed of circles. You can freely decide about the color, size and position. Use the following method call to set the size of the screen to 700 pixels × 700 pixels:
`t.screen.setup(width=700, height=700).`



Problem 6.6 *Stars* (1 point)

Write a program where you enter from the keyboard an integer value `n`. Then `n` stars are drawn onto the screen, circling around (not moving, but positioned like the small circles from the previous problem) the center of your screen. You can safely assume that $1 < n < 20$. Use the following method call to set the size of the screen to 700 pixels × 700 pixels:
`t.screen.setup(width=700, height=700).`
You can assume that the input will be valid and the screen will not be resized.

Problem 6.7 *Programmable turtle* (1 point)

Write a program that implements a programmable turtle. Your program should repeatedly read a string `s` which controls the turtle. If you enter the empty string then the program should stop. An `F` moves the turtle forward by distance 10, while an `L` turns the turtle left by 15 degrees and a `R` turns the turtle right by 15 degrees. A `U` moves up the pen, while a `D` moves the pen down. Use a jump table to implement the solution to this problem. Use the following method call to set the size of the screen to 600 pixels × 600 pixels:
`t.screen.setup(width=600, height=600).`
For example, the string `DFFFFLLLLLLFFFFLLLLLLFFFFLLLLLLFFFF` should draw a rectangle.
You can assume that the input will be valid.

Problem 6.8 *Translate Christmas greeting* (1 point)

Represent a small bilingual lexicon as a Python dictionary in the following fashion `dict = {"merry":"god", "christmas":"jul", "and":"och", "happy":"gott", "new":"nytt", "year":"år"}` and use it to translate your Christmas greeting (e.g., Merry Christmas and Happy New Year) from English into Swedish. That is, write and call a function `translate(list_en)` that takes a list of English words as parameter and returns a list of Swedish words. Print on the screen the translation of the greeting in Swedish. Note, that for printing 'å' you can use the unicode character '`\u00E5`'.
You can assume that the input will be valid.

Problem 6.9 *Computing a scalar product* (2 points)

Write a program that enters from the keyboard an integer `n` and two tuples `v` and `w` of real numbers with `n` components each. You can safely assume that the two inputs will have the same length. Write a function that computes the scalar product of `v` and `w`. The scalar product is defined as

$$v \cdot w = \sum_{i=1}^{n} v_i \cdot w_i$$

Use the function to compute the scalar product of the two vectors represented as tuples. Additionally, determine the smallest and largest components of the tuples, and the position in the tuple where they occur. Print these results on the screen.
You can assume that the input will be valid.

How to submit your solutions

Name the programs a6_p*x*.py.
Each program **must** include a comment on the top like the following:

```
# JTSK-350111
# a6_p1.py
# Firstname Lastname
# myemail@jacobs-university.de
```

You have to submit your solutions via *Grader* at
`https://grader.eecs.jacobs-university.de`.

If there are problems (but only then) you can submit the programs by sending mail to `k.lipskoch@jacobs-university.de` **with a subject line that starts with `JTSK-350111`**.

Please note, that after the deadline it will not be possible to submit solutions. It is useless to send solutions then by mail, because they will not be accepted.

Your code must run without any errors or warnings under python3.x.

This assignment is due by Monday, February 5th, 10:00 h