Introduction to Programming 2022, Mandatory Assignment 2

Dates:

- Hand-out date: September 29
- Hand-in date: October 12, 23:59

Rules:

- You hand in the solution before the hand-in date mentioned above.
- You hand in the solution individually.
- The hand-in is approved if at least 3 point are achieved.
- You can work on the assignment within your group. The hand-in must state all students that you have worked with.

Hand-in Format:

- Provide your answers by editing the files provided in the `answers-xxxx` directory.
  - You are not supposed to create any new files, nor change the names of the provided files in that directory.
  - Before handing in, replace the xxxx in the `answers-xxxx` directory with your ITU student ID.
    - For example, Helge's ITU login is `ropf` (as in the email address `ropf@itu.dk`)
    - So his assignment hand-in file would be `answers-ropf.zip`.
    - Similarly, Martin's file would be called `answers-maau.zip`.
  - That is, you **do not** hand-in plain text files only, no PDF files, no Word files, etc.

List of people that have been working together with for this assignment

Edit the file called `collaborators.txt` in the template and add the names of the students with whom you have been working.

Problem 1 (2 Points)

Assume that we have defined:

```
values = [2, 3, 9, 10]

def f(val):
    return 2 * val
```

What do the following expressions evaluate to? Write one value per line.

- `values[0]`
- `values[:2]`
- `values[::2]`
- `f(10)`
- `f("gbi")`

The file containing your solution is called `1.txt` and contains exactly one value per line. That is, the evaluation result of each of the expressions.

Problem 2 (2 Points)

You are working for a company that prints t-shirts and you need to add functionality to their web store. Your task is as follows:

Define a function `make_shirt` that has two parameters: the `size` of the shirt, which should be a string “S”, “M”, “L”, or “XL”, and the `message` to be printed on the shirt. The function **returns the string** “I will print a t-shirt in size YYY that says ‘ZZZ’ ”, where YYY is the `size` of the shirt, and ZZZ is the `message` to be printed on the shirt. Additionally, if `size` does not have a value “S”, “M”, “L”, or “XL”, the string to be returned is “This t-shirt size is unavailable”.

Hand-in your Python code in a file called `2.py`. Make sure to return the string from the function, i.e., no `print` statement is used in the function.

Examples:

- `make_shirt("S", "KBH")` returns the string `"I will print a t-shirt in size S that says 'KBH'"`.
- `make_shirt("L", "I love ITU")` returns the string `"I will print a t-shirt in size L that says 'I love ITU'"`.
- `make_shirt("XXS", "Amager4Ever")` returns the string `"This t-shirt size is unavailable"`, because “XXS” is not a valid t-shirt size according to our specification.

Beware: For `'I love ITU'` to be in the string, you have to use `"` quotation marks for the rest of the string, i.e., don’t do `variable = 'that says 'I love ITU''`, but use something like `variable = "that says 'I love ITU'"`.

```python
# The following has to work

print(make_shirt("S", "KBH"))
# Should print:  I will print a t-shirt in size S that says 'KBH'
print(make_shirt("XXS", "Amager"))
# Should print: This t-shirt size is unavailable
```

Problem 3 (2 Points)

Write a function `shortest_word(word_list)` that **returns** the shortest word in the list `word_list`. If more than one word is a shortest word, the function shall report the word that occurs first in the list, from left-to-right. If `word_list` is empty, i.e., contains no values, `shortest_word(word_list)` shall return `None`.

For example, `shortest_word(["denmark", "sweden", "germany"])` should return `"sweden"`. Likewise, `shortest_word(["long", "longer", "longest"])` should return `"long"`, since it has the fewest number of characters in the list. Also, `shortest_word(["one", "two", "six"])` should return `"one"`, since `"one"` is the shortest word that is first in the list.

Hand-in your Python code in a file called `3.py`.

Step-by-step approach

You can approach this problem in the following three steps:

1. Check if the list `word_list` contains no elements, and return `None` in that case.
2. Figure out a way to compare the length of two words in the list `word_list`. Is there a built-in function that could be helpful? (Hint: yes!).
3. Loop through the list `word_list`, comparing the length of each word to the length of the currently shortest word. So, you need a variable, say `shortest`, that contains the currently shortest word. You have to set `shortest` in the beginning to a suitable choice: Which one out of `word_list[0]` (the first word in the list), `word_list[1]`, ..., `word_list[-1]` do you take? (Hint: There is only one choice that will fulfill all the requirements in the problem statement.)

```python
# The following has to work:
word_list = ["denmark", "sweden", "germany"]
print(shortest_word(word_list)) # Should print: "sweden"
```