Scientific Programming 2; Practice Exam

Name:

……

- This is a digital exam consisting of 3 assignments where you will write short Python programs.
- You will complete the exam using the online editor linked on the course website.
- Use a single file for all solutions called sp2_exam_practice.py. This file is already open in the online exam editor.
- You may use the course website (sp.proglab.nl) as a recource. Keep in mind that:
  - you cannot use any other website,
  - you cannot use rely on any existing code you've written before this exam,
  - you cannot get assistance with the programming during the exam.
- You will be **evaluated solely on the correctness** of your solutions. Code design, comments, and style are not important, so you do not need to worry about them.
- Do not use external modules such as numpy, csv, or others unless the assignment specifically says you can.

- **Submitting the exam**. When you have completed all assignments:
  - use the **submit button** on the website to ensure your final edits are committed,
  - **go to the teacher** with your **student card or ID** and this exam sheet,
  - have the teacher mark your work as completed,
  - hand in this exam sheet.

Assignment 1: After is

Write a Python function called `after_is(text)` that takes a text string as input and returns keeps track which word directly follows the word 'is' or 'was'. The function returns a dictionary where the keys are the words that directly occur after 'is' or 'was' and the values count how often they occur in that position.

The function should consider words case-insensitively and should ignore common punctuation marks such as commas, periods, parentheses, exclamation marks, and question marks. You can use the following `tokenize()` function to convert the text to a list of words:

```
def tokenize(text):
    return [word.strip(" ,\n.);(!?)'").lower() for word in text.split(' ')]
```

Example usage:

```
text = """It was a place where up is down, and down is up; where nothing is
quite what it seems. Why is a raven like a writing desk?"""
print(after_is(text))
```

Expected output:

```
{'a': 2, 'down': 1, 'up': 1}
```

Assignment 2: Translations

Imagine we have two python dictionaries for translations. For instance an English-French and a French-Spanish dictionary. Write a function `combine_dicts(dict1, dict2)`, that combines them to create an English-Spanish dictionary.

Example usage:

```
english_to_french = {'banana': 'banane', 'apple': 'pomme', 'almond': 'amande',
                     'cat': 'chat', 'fine': 'amande'}
french_to_spanish = {'pomme': 'manzana', 'car': 'coche', 'banane': 'plátano',
                      'amande': 'almendra'}

english_to_spanish = combine_dicts(english_to_french, french_to_spanish)
print(english_to_spanish)
```

Expected output:

```
{'banana': 'plátano', 'apple': 'manzana', 'almond': 'almendra', 'fine': 'almendra'}
```

Assignment 3: Ingrid Bergman

The file `films.csv` contains data of over a 1000 films from before the 2000s. In this exercise, you will use `films.csv` to answer a question about the data. The contents of the file look as follows:

```
Year,Length,Title,Subject,Actor,Actress,Director,Popularity,Awards
1991,113.0,High Heels,Comedy,"Bos, Miguel","Abril, Victoria","Almodvar, Pedro",68.0,No
1988,108.0,Miles from Home,Drama,"Anderson, Kevin","Anderson, Jo","Sinise, Gary",53.0,No
1989,88.0,Final Notice,Mystery,"Gerard, Gil","Anderson, Melody","Stern, Steven Hilliard",88.0,No
1979,110.0,Quintet,Drama,"Newman, Paul","Andersson, Bibi","Altman, Robert",19.0,No
1960,90.0,"Devil's Eye, The",Drama,"Kulle, Jarl","Andersson, Bibi","Bergman, Ingmar",20.0,No
1957,91.0,Wild Strawberries,Drama,"Sjstrm, Victor","Andersson, Bibi","Bergman, Ingmar",42.0,Yes
....
```

The films have no particular order, and for each film the file contains the following information:

- Year: the year in which the film was released
- Length: the duration of the film in minutes
- Title: the title of the film
- Subject: the genre of the film
- Actor: the name of the main actor in the movie (contains only one name)
- Actress: the name of the main actress in the movie (contains only one name)
- Director: the name of the director of the movie (contains only one name)
- Popularity: the popularity of the movie on a scale from 0 to 100
- Awards: whether the movie has gotten an award (Yes/No)

Write a function `most_frequent_actress(filename)` that finds the actress that has the most appearances in this data file. Have the function also return a list of all the movies the actress appears in. Using `pandas` for this assignment is optional.

Example usage:

```
actress, movies = most_frequent_actress('films.csv')
print(actress)
print(movies)
```

Expected output:

```
Bergman, Ingrid
['Count of Old Town, The', 'Autumn Sonata', 'Gaslight', 'Indiscreet', 'Walpurgis Night',
'Joan of Arc', 'A Woman Called Golda', 'A Walk in the Spring Rain', 'Under Capricorn',
'Notorious', 'June Night', 'Goodbye Again', 'Anastasia', "Bells of St. Mary's, The",
'Intermezzo', "A Woman's Face", 'Swedenhielms', 'Only One Night']
```