Exam in Introduction to programming

2020-01-08

Instructions. Answer the questions below. They are roughly ordered by difficulty, but independent. Some questions are broken down into subquestions, *e.g*, question 4 might consist of subquestions 4-1, 4-2, and 4-3; the subquestions are supposed to be be treated in that order.

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, *e. g.*, `answers-jegp`. Hand in the renamed `answers-xxxx` directory as a single zip file called `answers-xxxx.zip`.

Some questions require extra input data. Such data is provided in a directory of the same name as the question. For instance, data for question four would be in the directory called `4`. In the questions below, the file name is given relative to the program you are supposed to write, so the data file `names.txt` for question four would be referred to as `../4/names.txt`, assuming you are standing in the `answers-xxxx` directory.

Any free text, such as program comments or explanations, can be written in Danish or English, but English is preferred.

At most 100 points can be earned in this exam.

Question 1 (6 Points)

For each of the following expressions, give their result.

```
3 - 3 + 2
"A" + "3" + "2"
3 * 12 - 1
3 ** 4
(True and False) or True
12 == 4
```

Question 2 (6 Points)

What is printed by the following 6 print statements?

```
a = ['Copenhagen', 'Helsinki', 'Oslo', 'Stockholm']
b = ['Denmark', 'Sweden']
s = 'Wait, is that a python over there?'
print(len(a))
print(a[2])
print(s[:5])
```

```
print(s[16:23])
print(a[0] + " " + b[0])
print(a + b)
```

Question 3 (8 Points)

What is the purpose of the following code?

```
s = "this is the introduction to programming exam"

balance = 0
for c in s:
    if c in 'aeiou':
        balance += 1
    else:
        balance -= 1
```

Pick one (and exactly one) of the following:

a. Decide whether there are more vowels (i.e., characters a, e, i, o, u) than consonants in the string `s`.
b. Count the number of vowels in the number `s`.
c. Make the balance of the string `s` positive.
d. Count the number of lower-case characters in the string `s`.
e. Decide if the string `"c"` is in the string `s`.

Question 4 (10 Points)

Claus came up with the following function that is supposed to double elements in a list.

```
def double_list_elements(other_list):
    i = 0
    while i < len(other_list):
    other_list[i] = other_list * 2
    i = i - 1

my_list = [1, 3, 5]
double_list_elements(my_list)
print(my_list) # should print [2, 6, 10]
```

4-1 (6 Points)

The function doesn't work. It doesn't even run! Correct the function, but still use the while loop. You are only allowed to change the code inside the function `double_list_elements`.

4-2 (4 Points)

Rewrite your solution from 4-1 to use a for loop instead of a while loop.

Question 5 (12 Points)

ITU's IT department has problems with its user management. Their old system was written in the ancient program language ALGO60. Unfortunately, the person that wrote the program left ITU 15 years ago and no one knows what the program is doing any longer. You are hired to help ITU out. Your task is to write a program that creates an email address for a given person.

Write a function `create_itu_address` that takes two arguments `given_name` and `family_name` and returns a string that represents that person's email address at ITU. The email address consists of the first two characters of `given_name` followed by the first two characters of `family_name` (all lower-case), ending with `@itu.dk`.

You can assume that both `given_name` and `family_name` are at least two characters long, and you can ignore the problem that two people with different names might get the same email address.

Test the correctness of your function by **providing two assert statements** that test that the function produces the correct output.

Examples: `create_itu_address("Martin", "Aumüller")` should return the string `"maau@itu.dk"`, and `create_itu_address("Helge", "Pfeiffer")` should return `"hepf@itu.dk"`.

Question 6 (12 Points)

An anagram is a word formed by rearranging the letters of a different word using all the original letters exactly once. For example, "listen" is an anagram of "silent".

Write a function `is_anagram(word1, word2)` that returns `True` if `word1` is an anagram of `word2`, and `False` otherwise. You can assume that the two words do not contain blanks. For full points, your program must handle the case that words may contain individual letters more than once, such as in "railsafety" and "fairytales".

Question 7 (16 Points)

7-1 (8 Points)

Create a class `University`. Each university has three attributes `name` (the name of the university), `location` (a description of the location of the university), and `students` (the number of students enrolled at the university) that are given when creating a university. Add a method `print_description` that prints a message such as "IT University of Copenhagen is in Copenhagen and has 2355 enrolled students", including information about all three attributes. Add explanatory docstrings as documentation to the class.

Instantiate the following three universities and call the `print_description` method on each of them. (The order doesn't matter.)

- IT University of Copenhagen, Copenhagen, 2355
- University of Copenhagen, Copenhagen, 38615
- University of Aarhus, Aarhus, 44500

7-2 (4 Points)

(Copy and paste your solution from the previous question.)

Add a method `update_student_numbers` to the class `University` that takes two arguments: the number of newly enrolled students and the number of students that left the university (for example because they graduated). Update the attribute containing the number of students according to these numbers. Add an explanatory docstring as documentation to this method.

Call this method for the IT University of Copenhagen with 1100 newly enrolled students and 800 students that left the university. Print its description afterwards. (Make sure that the number of students is updated.)

7-3 (4 Points)

(Copy and paste your solution from the previous question.)

As it often happens in administration, there might be a zero too many at the end of a number which would have catastrophic results. (For example, think about 1100 newly enrolled students and 8000 students leaving ITU.) Update the method `update_student_numbers` such that it prints a message "I don't believe these numbers are correct" if `update_student_numbers` is called with two numbers which would result in a negative number of students. Don't update the number of students in such a case. Demonstrate that your program works as intended by calling `update_student_numbers` and printing the university description before and after the call.

Instead of printing such a message you can also raise an `AssertionError` by providing a proper `assert` statement, or raise a custom error. (All correct solutions result in full points.)

Question 8 (18 Points)

You are managing the grade system of a school. The students have to pass five tests "A", "B", "C", "D", and "E" to complete the school, and get a grade according to the Danish grade system for each test.

The file `../8/grades.txt` contains data about the tests taken by students and the achieved grade. Each line in the file contains the name of the student, the test the student took, and the achieved grade. You can assume that no two students share the same name. Furthermore, a student cannot retake the test if a passing grade was achieved.

8-1 (6 Points)

Write a program that asks the user to input a name. It then displays all the test results for that name that can be found in the grade file. (You can just print out those lines in the file.)

For example, if the user input is the name *Magnus*, your program should produce an output that roughly looks as follows:

```
Magnus, A, 7
Magnus, D, 0
Magnus, D, 0
Magnus, D, 12
Magnus, B, -2
Magnus, C, 12
Magnus, B, -2
Magnus, B, 4
Magnus, E, 4
```

8-2 (6 Points)

Write a function `print_finished` that takes as argument a filename and prints the names of all students that have finished school. A student has finished school if all tests have been taken and in each test a grade of at least 2 has been achieved.

8-3 (6 Points)

Write a function `print_best_student` that takes as argument a filename and prints the student that finished school with the highest average grade. Print both the name and the average grade achieved. You can assume that grade averages are distinct and that you do not have to worry about multiple students having the same average grade.

Question 9 (12 Points)

Here is a simple program to find the elements that two lists `list_1` and `list_2` have in common.

```python
def find_common(list_1, list_2):
    res = []
    for x in list_1:
        for y in list_2:
            if x == y:
                res.append(x)
    return res

print(find_common([1, 2, 5], [2, 3, 4]))
# prints [2]
```

(To make our life easier, let's assume that each list has only distinct elements and that both lists contain the same number of elements.)

9-1 (4 Points)

Run the program on some larger lists. What is the length of the largest lists that the program can handle in a few seconds?

a. 1000

b. 10000

c. 100000

d. 1000000

9-2 (4 Points)

What is the running time of the function?

a. constant

b. logarithmic in the length of the lists

c. linear in the length of the lists

d. quadratic in the length of the lists

9-3 (4 Points)

Write a program that finds the common elements of two lists much faster. Your program must be able to find the common elements of lists of length 10000000 within an hour.