

Assignment 3 - Conditions, Loops and Functions

- 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 3.1 *Categorization of characters*

(1 point)

Presence assignment, due by 18:30 h today

Write a program, where you can enter a character from the keyboard. Then determine whether the character is a lowercase alphabetic character or not and print a corresponding message in both cases on the screen.

You can assume that the input will be valid.

Problem 3.2 *Infinite loop by bad coding*

(1 point)

Presence assignment, due by 18:30 h today

The program below is supposed to print the numbers 0 to 9, but it surprisingly just prints

```
n is 0
n is 0
...
```

until you press `Ctrl-C`. Fix this program such that it prints the numbers 0 to 9 without changing the `while` loop into a `for` loop. Explain your fixes within comments.

```
n = 0
while n < 11:
    print("n is", n)
    n += 1
print("That's it.")
```

Problem 3.3 *Seconds and minutes I*

(1 point)

Presence assignment, due by 18:30 h today

Write a program where you can enter from the keyboard an integer `n`. Then a conversion table for 1 to `n` minutes is printed on the screen as in the example below. Use a `while` loop in your solution.

```
1 minute = 60 seconds
2 minutes = 120 seconds
3 minutes = 180 seconds
...
```

Print the table exactly (concerning the use of singular and plural) as given here in the example. You can assume that the input will be valid.

Problem 3.4 *Seconds and minutes II*

(1 point)

In your previous program replace the `while` loop with a `for` loop such that the program will have the same functionality.

Problem 3.5 *Writing characters I*

(1 point)

Write a program where you first enter a character `ch` and then an integer `n`, both from the keyboard. Print the character `ch` `n` times on the screen.

You can assume that the input will be valid.

Problem 3.6 Writing characters II

(1 point)

Write a program where you first enter from the keyboard an uppercase character `ch` and then an integer `n`. Print on the screen the characters from `ch` until `n` characters after `ch` in the ASCII table order. For example, if `n = 3` and `ch = 'A'` then `'A', 'B', 'C', 'D'` should be printed.

You can assume that `ch` is uppercase and `n` will be a valid integer for the context of the problem.

Bonus Problem 3.7 Writing characters III

(1 point)

Add checks to your previous program such that it will repeat to read the input until it is valid. Therefore make sure that `ch` is actually an uppercase character, otherwise repeatedly read `ch`. Also make sure that `n` is greater or equal than 0 and less or equal than 32, otherwise repeatedly read `n`.

Problem 3.8 Computing sum and average

(1 point)

Write a program where you can enter from the keyboard up to 10 integers. If the number entered is equal to `-9`, stop reading numbers from the keyboard and compute the sum and average of all values (excluding `-9`). Print the average value on the screen.

Make sure you consider all the cases: less than 10 or exactly 10 numbers can be entered. After the numbers have been entered you need to make sure that the average is computed.

You can assume that the input will be valid and `-9` will not be used as one of the values.

Problem 3.9 Printing a rectangle

(1 point)

Write a program which reads from the keyboard two integers `n` and `m`, and a character `c` also from the keyboard. This program should define and call a function:

```
print_rectangle(n, m, c)
```

which prints a rectangle of size `n×m` consisting of the character `c`.

You can assume that the input will be valid.

Bonus Problem 3.10 Printing a frame

(1 point)

Write a program which reads from the keyboard two integers `n` and `m`, and a character `c` also from the keyboard. This program should define and call a function:

```
print_frame(n, m, c)
```

which prints a frame of size `n×m` consisting of the character `c` as in the following testcase.

You can assume that the input will be valid.

Testcase 3.10: input

```
4
7
$
```

Testcase 3.10: output

```
$$$$$$$
$      $
$      $
$$$$$$$
```

How to submit your solutions

Name the programs `a3-px.py`.

Each program **must** include a comment on the top like the following:

```
# JTSK-350111
# a3-pl.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 Sunday, January 28th, 10:00 h