The one with the Smart Home sensors

Due Tuesday by 23:59 **Points** 1 **Submitting** an external tool
Available until 1 Nov at 23:59

Successfully solving this problem will award the student **1 point** towards Examination 1

Requirements

- The script must pass all the automatic tests
- The script must pass a manual inspection
- **The student must write the script on his or her own**, any suspicion about cheating, cooperation, and/or plagiarism will be reported to the Discipline Board (see the **Study Guide** **(https://hkr.instructure.com/courses/5190/files/1027789/download?download_frd=1)** for more information). The students script will be checked both manually and using anti-plagiarism software

Instructions

1. Analyse the problem below using Polya's Problem solving Technique (https://math.berkeley.edu/~gmelvin/polya.pdf)
2. Create a script named **Lab3_1.py** that solves the problem according to the given instructions
3. Pay special attention when formatting the output as it must match the specified format **exactly** (character by character) to pass the tests
4. Upload the script using the link below

Task description

Last summer you installed a Smart Home system at your parents' place, but this move cost you a lot of headache. One of the sensors malfunction which resulted in that the indoor temperature was all over the place. One day it could be hot like in a sauna indoors and the next day you would need to wear mittens. In the end you found, and replaced, the malfunctioning sensor but it took a long time to find out what the problem was.

This year when taking a course in embedded systems you remembered the ordeal from last summer and wanted to come up with a solution. The end goal is to make the Smart Home a little bit smarter, to have it report that one of its sensors is malfunctioning. In your prototype you have installed not one, but two, of each sensor (two sensors measuring the temperature in the kitchen instead of just one). The idea is to create a script that can check if the measurements from the two sensors deviate too much, if they do, there is most likely a problem with one of the sensors.

Luckily the Smart Home have an export function allowing all the measurements from the primary sensors to be stored in one file and all the measurements from the secondary sensors (the validation sensors) to be stored in another file. Each file will contain a pickled dictionary on the following format:

```
{‘Kitchen’: ‘24°’, ’Master bedroom’: ’23°’}
```

What the script needs to do is to take the path to two of these measurement files as command line arguments and then figure out if the values from the two files deviate too much. Currently the threshold for what is too much shall be set to two degrees. The script needs to implement the following functions:

read_file(filename)
Shall use pickle to load the dictionary stored in the file referenced by the parameter *filename*. The function shall return the read dictionary, or raise a FileNotFoundError if the filename is not valid.

map_to_int(measurements)
The parameter *measurements* is a dictionary on the format described earlier, a tag describing where the sensor is located, and a temperature value. Note that both are strings and that the temperature value is followed by a degree sign. This function shall create a new dictionary that contains the same tags as the *measurements* dictionary but where all the values are integers. An example: {'Kitchen': '20°'} shall become {'Kitchen': 20}. The new dictionary shall be returned from the function.

find_faulty(primary, secondary, threshold)
This function takes two dictionaries on the format {'Kitchen': 20} together with an integer value called *threshold* as parameters. The function shall compare each key-value pair in the *primary* dictionary to the same pair in the *secondary* dictionary and if the difference between the two values is greater than the *threshold* value, the key shall be added to a

set. After having compared all the key-value pairs the set shall be returned from the function.

display_warnings(faulty_sensors)
The *faulty_sensors* parameter is a set containing all the sensor tags (as strings) where the difference was too big between the primary and secondary sensors. The function shall display a message containing all the sensor tags on the following format:

```
Analyzis of the provided files is complete.
-------------------------------------------

The following sensors differ more than 2°:
Kitchen
```

main()
This function shall use the command line arguments to read the primary and secondary sensor values from the specified files and then continue to prepare the dictionaries so that they meet the requirements needed to call the function that finds the faulty sensors. Finally the function shall find and print the faulty sensors. The main function shall use the other functions to accomplish its tasks and it shall have error handling that prints the message below if the files given as command line arguments are not valid:

```
Error: The files given as arguments are not valid.
```

To avoid problems with the testing framework, and to build a good habit, call the main function using the following construct:

```
if __name__ == '__main__':
    main()
```

Comments and additional resources

- Students are highly recommended to first solve all the tasks from the corresponding lectures and exercises before attempting to solve this problem. Having a solid plan before starting to code will also help
- The student can upload as many times as he or she likes to before the deadline
- These two files (**primary** **(https://hkr.instructure.com/courses/5190/files/1017888/download?download_frd=1)** & **secondary** **(https://hkr.instructure.com/courses/5190/files/1017889/download?download_frd=1)**) can be used as an example of how the files with sensor readings might look like
- Watch this (https://youtu.be/IOeIDvyRUQs) video if you want to learn more about why we call main the way we do

- This video (https://youtu.be/o1Rxrtiqml8) covering how to use command line arguments might be useful
- This video (https://youtu.be/Qf9t75riuFk) recorded while running the finished script showcases how the script shall look and work when finished

This tool needs to be loaded in a new browser window

Load The one with the Smart Home sensors in a new window