previous up next

Tartalomjegyzék

Exercises

Shopping

In the `hazi` system, exercise: **`shopping`**

Enciclopedia

In the `hazi` system, exercise: **`enciclopedia`**

Grading

After the midterm, you want to program a query system, where the students can ask their points with their Neptun code. They enter their Neptun code and the grade should be returned.

Write a python function called `get_grade` that

- has two parameters:
  - `percents` a dict where the keys are the NEPTUN codes and the values are the result of that student in percents.
  - `neptun` the Neptun code of the student in question
- return the grade of this student

The grading system is as this:

- below 40%: 1
- from 40% and above: 2
- from 55% and above: 3
- from 70% and above: 4
- from 85% and above: 5

Top grades

After evaluating the midterm results, you want to select the top performing students to congratulate them.

You have to write a similar function called `topgrades` that

- has one parameter: `percents`, a dict like before. The keys are the neptun codes, the values are the results in percent.
- return the list of neptun codes of those who got the top grade (5)

Hint

You can use the previous function.

Goals

Louis and his friends play soccer every weekend and they want to know who scored the most goals at the end of the season. Louis asked you to write a python function for them to track their scores.

The function should be names `goals` with three parameters:

- `results` the results so far, a dictionary where the keys are the players and the values are their total scores up to this point in time.
- `player` is a string, the name of a player
- `goal` an integer, how much goals did he/she scored during the last game.

The function should update the `results` by adding the scores of the person to the previous scores. Mind that if the player scored for the first time in this season then the dictionary won't contain his/her name but you have to add it to the dictionary.

Return the updated `results` dictionary.

Inventory update

A warehouse keeps its inventory in a python list where every item has an integer ID. During an inventory update they want to change the IDs of certain items.

Write a python function called `inventoryupdate` that

- has two paremeters:
    - `inventory` a list of the items represented by their IDs. The same ID can appear multiple times if they have more than one of that item.
    - `newid` a dictionary representing the changes. The keys are the old IDs and the values are the new IDs. If an item doesn't change its ID then it is not even included in this dictionary.
- return the updated inventory (a list of new IDs) where every item is changed according to the updates dictionary, or has the same ID if it was not changed. Don't change the order of the inventory.

previous up next