

1. Pana acum am tot facut copy/paste la diverse functii dintr-un program în altul. Acum stim sa creem module, gata cu copy/paste. :-) Refacem codul problemelor 1-4 de la tema anterioara dar grupam functiile într-un singur modul. Modulul acesta îl importam în toate celelalte 4 programe. Fie folositi codul vostru fie va inspirati sau folositi complet ca sursa solutiile pe care le-am urcat pe site. Ideal ar fi sa folositi codul vostru. Puteti face orice fel de modificari mai considerati. Ideea e sa avem un modul care sa contina functii iar în programele care importa acel modul sa avem cod cat mai putin. Ramane la latitudinea voastra cum numiti modulul, cum îl importati (pe tot sau anumite functii doar), etc..

2. La problema anterioara, inlocuim type hinting-ul și mini-documentatia functiilor cu niște docstring-uri explicite. Atentie la sintaxa și la organizarea lor. Verificati sa va atentioneze pycharm daca apelati functiile cu valori cu tipuri care nu corespund docstring-ului. Daca va atentioneaza atunci ati creat corect docstring-ul, daca nu nu.

3. Haideti sa ne antrenam în ce privește folosirea parametrului optional **key** de la **sorted**.

a. avem doua liste de string-uri de felul urmator

```
ls = ['5', '123', '-7', '33']
```

```
li = [5, 123, -7, 33]
```

Observati diferentele dintre `sorted(li)` și `sorted(ls)`. De ce sunt aceste diferente? Incercati sa folositi parametrul `key` al lui `sorted` ca sa îl sortati pe `ls` în functie de conversia la `int` a fiecarui string component al listei.

b. Daca avem o lista de tuple ca aceasta: `l = [(5, 21, 8), (6, 11, -5), (0, 25, 3), (-6, 6, 1)]`

Incercati sa sortati lista dar criteriul de sortare sa fie suma celor doua elemente din tupla. Adica la inceput o sa fie tuplele cu suma elementelor mai mica iar spre final alea cu suma tot mai mare.

La punctele a și b de mai sus puteti folosi pentru `key` fie vreo functie creata de voi sau daca gasiti deja ceva existent în Python e și mai bine. Important e sa rezolvati problemele.

4. Incercati sa va instalati în PyCharm package-ul `XlsxWriter`. Familiarizati-va cu acest package citind descrierea lui de pe pypi.org sau site-ul sau dedicat (gasiti link-ul tot în pagina package-ului de pe pypi.org) Incercati sa rulati acel program exemplu de pe pagina de prezentare de pe pypi.org. Ar trebui sa va genereze un fisier `xlsx`. Visionati-l cu Excel sau cu alt program care e capabil sa deschida fisiere `xlsx` (de exemplu LibreOffice daca nu aveti Microsoft Excel <https://www.libreoffice.org/>)

5. După ce v-ați familiarizat cu XlsxWriter, generați un fișier xlsx care să arate în felul următor

	A	B	
1	Elev	Media	
2	Nume prenume 1	8.50	
3	Nume prenume 2	7.25	
4	Nume prenume 3	6.75	
5			
6			

Elev și Media să fie cu bold. Numele elevilor să fie în ordine alfabetică și aliniate la stânga. Mediile să fie aliniate la dreapta. Studiați documentația lui XlsxWriter ca să vedeți cum se poate genera un astfel de fișier. Folosiți catalogul definit la temele anterioare, fie textul fie dicționarul, care vă e mai ușor. Ce mă interesează este fișierul xlsx generat.