

Tutoriatul 04

Tupluri, set-uri, dicționare, string-uri,
fișiere

Înapoi la [colecții](#)

Reamintim: În curs, veți întâlni următoarele tipuri de colecții:

1. Liste
2. Tupluri
3. Muștimi (set-uri, frozenset-uri)
4. Dicționare

Dintre ele, în acest tutoriat vom vorbi despre tupluri, muștimi și dicționare.

Tupluri - introducere*

Exemplu de tuplu: t = (1, -3.5, 'Hello', 7 == 9, None, ["text", (20, 30, 40), True])

Tuplurile sunt un fel de liste **îmmutabile**, dar în practică sunt folosite în alte scopuri decât listele. Iată un exemplu de utilizare:

Reprezentarea coordonatelor unui punct în spațiu:

```
punct = (5, 0, -3)
```

Tuplurile pot fi folosite drept chei pentru dicționare (foarte folositor!!)

*pt. mai multe detalii consultați cursul

Set-uri - introducere*

Set-urile din Python sunt implementarea mulțimilor din matematică, iar frozenset-urile sunt mulțimi imutabile (frozen).

Exemplu de set:

```
s = {1, (2, 3), "Python e super!"}
print(s)
```

```
{(2, 3), 1, 'Python e super!'}
```

Obs: - set-urile sunt **neordonate** (nu mai putem accesa s[1]).

- există operațiile de reuniune, intersecție etc.

*pt. mai multe detalii consultați cursul

Ce este ăla un dicționar în Python?



Ce este ăla un dicționar în Python?

- structură de date (colecție) neordonată, **mutabilă** și alcătuită din perechi **cheie-valoare**
- Cheia trebuie să fie **unică**, **imutabilă** și **hashable** (imutabil este de multe ori hashable)

Ce înseamnă de fapt asta:

Ce este ăla un dicționar în Python?

- structură de date (colecție) neordonată, **mutabilă** și alcătuită din perechi **cheie-valoare**
- Cheia trebuie să fie **unică**, **imutabilă** și **hashable** (imutabil este de multe ori hashable)

Ce înseamnă de fapt asta:

Listă: 1 4 5 2 65 3 6

Indici: 0 1 2 3 4 5 6

Ce este ăla un dicționar în Python?

- structură de date (colecție) neordonată, **mutabilă** și alcătuită din perechi **cheie-valoare**
- Cheia trebuie să fie **unică**, **imutabilă** și **hashable** (imutabil este de multe ori hashable)

Ce înseamnă de fapt asta:

Listă: 1 4 5 2 65 3 6

Indici: 0 1 2 3 4 5 6

Dicționare: 1 4 5 2 65 3 6

Ce este ăla un dicționar în Python?

- structură de date (colecție) neordonată, **mutabilă** și alcătuită din perechi **cheie-valoare**
- Cheia trebuie să fie **unică**, **imutabilă** și **hashable** (imutabil este de multe ori hashable)

Ce înseamnă de fapt asta:

Listă: 1 4 5 2 65 3 6

Indici: 0 1 2 3 4 5 6

Dicționare: 1 4 5 2 65 3 6

~~Indici~~ Chei:

Ce este ăla un dicționar în Python?

- structură de date (colecție) neordonată, **mutabilă** și alcătuită din perechi **cheie-valoare**
- Cheia trebuie să fie **unică**, **imutabilă** și **hashable** (imutabil este de multe ori hashable)

Ce înseamnă de fapt asta:

Listă: 1 4 5 2 65 3 6

Indici: 0 1 2 3 4 5 6

Dicționare: 1 4 5 2 65 3 6

Indicei Chei: 0 "unu" ("gigel", "militaru") etc.

La ce ne folosește?

1. Organizarea eficientă a datelor. Avem următoarele situații, sub formă de tabel:

- i) studenții la un optional: nume, note, nr. Matricol
- ii) user-ii unui site: username, mail, activitate recentă
- iii) candidați pentru șefia grupei: obiective, vârstă, nume

Î: Ce am putea alege drept cheie pentru fiecare situație?

La ce ne folosește?

1. Organizarea eficientă a datelor. Avem următoarele situații, sub formă de tabel:

i) studenții la un optional: nume, note, nr. Matricol

Nr. matricol (singurul unic)

ii) user-ii unui site: username, mail, activitate recentă

iii) candidați pentru şefia grupei: obiective, vârstă, nume

La ce ne folosește?

1. Organizarea eficientă a datelor. Avem următoarele situații, sub formă de tabel:

i) studenții la un optional: nume, note, nr. Matricol

Nr. matricol (singurul unic)

ii) user-ii unui site: username, mail, activitate recentă

Mail sau username dacă este atribuit în mod unic

iii) candidați pentru şefia grupei: obiective, vârstă, nume

La ce ne folosește?

1. Organizarea eficientă a datelor. Avem următoarele situații, sub formă de tabel:

i) studenții la un optional: nume, note, nr. Matricol

Nr. matricol (singurul unic)

ii) user-ii unui site: username, mail, activitate recentă

Mail sau username dacă este atribuit în mod unic

iii) candidați pentru şefia grupei: obiective, vârstă, nume

Nume

La ce ne folosește?

2. Probleme de frecvență a datelor. Avem următoarele situații:

- i) Se dă un sir finit de cifre. Să se afișeze de câte ori apare fiecare cifră în sir.

Î: Cum facem?

La ce ne folosește?

2. Probleme de frecvență a datelor. Avem următoarele situații:

- i) Se dă un sir finit de cifre. Să se afișeze de câte ori apare fiecare cifră în sir.

Atribuim direct

SAU(folosind comprehensiunea)

```
d = {0 : 0, 1 : 0, 2 : 0, 3 : 0, ...  
for element in lista:  
    d[element] = d[element] + 1  
print(d)
```

La ce ne folosește?

2. Probleme de frecvență a datelor. Avem următoarele situații:

- i) Se dă un sir finit de cifre. Să se afișeze de câte ori apare fiecare cîfră în sir.

Atribuim direct

```
d = {0 : 0, 1 : 0, 2 : 0, 3 : 0, ...  
for element in lista:  
    d[element] = d[element] + 1  
print(d)
```

SAU(folosind comprehensiunea)

```
d = {x : 0 for x in range(10)}  
for element in lista:  
    d[element] = d[element] + 1  
print(d)
```

Output: {0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 0}

La ce ne folosește?

2. Probleme de frecvență a datelor. Avem următoarele situații:

ii) Se dă o frază. Să se determine care caracter apare de cele mai multe ori.

Î: Cum facem?

Hint: ‘d[cheie] = valoare’ este o

La ce ne folosește?

2. Probleme de frecvență a datelor. Avem următoarele situații:

ii) Se dă o frază. Să se determine care caracter apare de cele mai multe ori.

Î: Cum facem?

Hint: ‘d[cheie] = valoare’ este o **atribuire corectă în orice caz!**

La ce ne folosește?

2. Probleme de frecvență a datelor. Avem următoarele situații:

ii) Se dă o frază. Să se determine care caracter apare de cele mai multe ori.

Populare succesivă.

```
sir = "Ana are mere si alune frate"
d = {}
for litera in sir:
    if litera in d: #verific daca este cheie deja
        d[litera] = d[litera] + 1
    else:
        d[litera] = 1
print(d)
```

Output:

La ce ne folosește?

2. Probleme de frecvență a datelor. Avem următoarele situații:

ii) Se dă o frază. Să se determine care caracter apare de cele mai multe ori.

Populare succesivă.

```
sir = "Ana are mere si alune frate"
d = {}
for litera in sir:
    if litera in d: #verific daca este cheie deja
        d[litera] = d[litera] + 1
    else:
        d[litera] = 1
print(d)
```

Output:

```
{'A': 1, 'n': 2, 'a': 4, ' ': 5, 'r': 3, 'e': 5, 'm': 1, 's': 1, 'i': 1, 'l': 1, 'u': 1, 'f': 1, 't': 1}
```

Accesare, modificare și adăugarea elementelor

Dacă vrem să accesăm strict **cheile**:

`dict.keys()` -> cheile lui dict în "listă" (echivalent cu 'for i in dict')

Dacă vrem să accesăm strict **elementele**:

`dict.values()` -> valorile lui dict în "listă"

Dacă vrem să accesăm perechile **cheile - elementele**:

`dict.items()` -> "listă" cu tupluri formate din cheie + valoare

```
d = {'A': 1, 'n': 2, 'a': 4, ' ': 5, 'r': 3, 'e': 5, 'm': 1, 's': 1, 'i': 1, 'l': 1, 'u': 1, 'f': 1, 't': 1}
print(d.values())
print(d.keys())
print(d.items())
print(d)

dict_values([1, 2, 4, 5, 3, 5, 1, 1, 1, 1, 1, 1])
dict_keys(['A', 'n', 'a', ' ', 'r', 'e', 'm', 's', 'i', 'l', 'u', 'f', 't'])
dict_items([('A', 1), ('n', 2), ('a', 4), (' ', 5), ('r', 3), ('e', 5), ('m', 1), ('s', 1), ('i', 1), ('l', 1), ('u', 1), ('f', 1), ('t', 1)])
{'A': 1, 'n': 2, 'a': 4, ' ': 5, 'r': 3, 'e': 5, 'm': 1, 's': 1, 'i': 1, 'l': 1, 'u': 1, 'f': 1, 't': 1}
```

Probleme de rezolvat

Pentru a aprofunda noile cunoștințe vom rezolva o serie de probleme legate de dicționare

Problema 1

Se dă un dicționar unde valorile sunt comparabile între ele. Să se afișeze valoarea cea mai mică. (considerați că dicționarul este salvat în variabila d)

Î: Cum facem?

Problema 1

Se dă un dicționar unde valorile sunt comparabile între ele. Să se afișeze valoarea cea mai mică. (considerați că dicționarul este salvat în variabila d)

I: Cum facem?

R: Trebuie, în primul rând, să accesăm valorile dicționarului

Problema 1

Se dă un dicționar unde valorile sunt comparabile între ele. Să se afișeze valoarea cea mai mică. (considerați că dicționarul este salvat în variabila d)

I: Cum facem?

R: Trebuie, în primul rând, să accesăm valorile dicționarului `ls = d.values()`

Problema 1

Se dă un dicționar unde valorile sunt comparabile între ele. Să se afișeze valoarea cea mai mică. (considerați că dicționarul este salvat în variabila d)

I: Cum facem?

R: Trebuie, în primul rând, să accesăm valorile dicționarului. `ls = d.values()`

Trebuie să găsim minimul.

Problema 1

Se dă un dicționar unde valorile sunt comparabile între ele. Să se afișeze valoarea cea mai mică. (considerați că dicționarul este salvat în variabila d)

I: Cum facem?

R: Trebuie, în primul rând, să accesăm valorile dicționarului.

Trebuie să găsim minimul.

```
ls = d.values()  
minimul = min(ls)
```

Problema 1

Se dă un dicționar unde valorile sunt comparabile între ele. Să se afișeze valoarea cea mai mică. (considerați că dicționarul este salvat în variabila d)

I: Cum facem?

R: Trebuie, în primul rând, să accesăm valorile dicționarului.

Trebuie să găsim minimul.

Trebuie să îl afișăm.

```
ls = d.values()  
minimul = min(ls)
```

Problema 1

Se dă un dicționar unde valorile sunt comparabile între ele. Să se afișeze valoarea cea mai mică. (considerați că dicționarul este salvat în variabila d)

I: Cum facem?

R: Trebuie, în primul rând, să accesăm valorile dicționarului.

Trebuie să găsim minimul.

Trebuie să îl afișăm.

```
ls = d.values()  
minimul = min(ls)  
print(minimul)
```

Problema 1

Se dă un dicționar unde valorile sunt comparabile între ele. Să se afișeze valoarea cea mai mică. (considerați că dicționarul este salvat în variabila d)

I: Cum facem?

R: Trebuie, în primul rând, să accesăm valorile dicționarului.

Trebuie să găsim minimul.

Trebuie să îl afișăm.

```
ls = d.values()  
minimul = min(ls)  
print(minimul)
```

Tot ce este mai sus poate fi scris într-o singură linie de cod.

Problema 1

Se dă un dicționar unde valorile sunt comparabile între ele. Să se afișeze valoarea cea mai mică. (considerați că dicționarul este salvat în variabila d)

I: Cum facem?

R: Trebuie, în primul rând, să accesăm valorile dicționarului.

Trebuie să găsim minimul.

Trebuie să îl afișăm.

```
ls = d.values()  
minimul = min(ls)  
print(minimul)
```

Tot ce este mai sus poate fi scris într-o singură linie de cod.

```
print(min(d.values()))
```

Problema 2

Se dă un dicționar. Să se afișeze (sub orice formă convenabilă) valorile corespunzătoare cheilor care sunt siruri de caractere de lungime cel puțin 4.

Î: Cum facem?

Problema 2

Se dă un dicționar. Să se afișeze (sub orice formă convenabilă) valorile corespunzătoare cheilor care sunt siruri de caractere de lungime cel puțin 4.

Î: Cum facem?

R: Ne uităm printre cheile dicționarului

Problema 2

Se dă un dicționar. Să se afișeze (sub orice formă convenabilă) valorile corespunzătoare cheilor care sunt siruri de caractere de lungime cel puțin 4.

Î: Cum facem?

R: Ne uităm printre cheile dicționarului `ls = d.keys()`

Problema 2

Se dă un dicționar. Să se afișeze (sub orice formă convenabilă) valorile corespunzătoare cheilor care sunt siruri de caractere de lungime cel puțin 4.

Î: Cum facem?

R: Ne uităm printre cheile dicționarului `ls = d.keys()`

Verificăm pe fiecare în parte dacă
verifică condiția

Problema 2

Se dă un dicționar. Să se afișeze (sub orice formă convenabilă) valorile corespunzătoare cheilor care sunt siruri de caractere de lungime cel puțin 4.

I: Cum facem?

R: Ne uităm printre cheile dicționarului `ls = d.keys()`

Verificăm pe fiecare în parte dacă `for element in ls:`
verifică condiția

Problema 2

Se dă un dicționar. Să se afișeze (sub orice formă convenabilă) valorile corespunzătoare cheilor care sunt siruri de caractere de lungime cel puțin 4.

I: Cum facem?

R: Ne uităm printre cheile dicționarului

```
ls = d.keys()
```

Verificăm pe fiecare în parte dacă
verifică condiția

```
for element in ls:  
    if type(element) is str and len(element) >= 4:
```

Problema 2

Se dă un dicționar. Să se afișeze (sub orice formă convenabilă) valorile corespunzătoare cheilor care sunt siruri de caractere de lungime cel puțin 4.

I: Cum facem?

R: Ne uităm printre cheile dicționarului

```
ls = d.keys()
```

Verificăm pe fiecare în parte dacă
verifică condiția

```
for element in ls:  
    if type(element) is str and len(element) >= 4:  
        print(d[element])
```

Problema 2

Se dă un dicționar. Să se afișeze (sub orice formă convenabilă) valorile corespunzătoare cheilor care sunt siruri de caractere de lungime cel puțin 4.

I: Cum facem?

R: Ne uităm printre cheile dicționarului

```
ls = d.keys()
```

Verificăm pe fiecare în parte dacă
verifică condiția

```
for element in ls:  
    if type(element) is str and len(element) >= 4:  
        print(d[element])
```

Tot ce este mai sus poate fi scris
într-o singură linie de cod*.

Problema 2

Se dă un dicționar. Să se afișeze (sub orice formă convenabilă) valorile corespunzătoare cheilor care sunt siruri de caractere de lungime cel puțin 4.

I: Cum facem?

R: Ne uităm printre cheile dicționarului `ls = d.keys()`

Verificăm pe fiecare în parte dacă
verifică condiția

```
for element in ls:  
    if type(element) is str and len(element) >= 4:  
        print(d[element])
```

Tot ce este mai sus poate fi scris
într-o singură linie de cod*.

```
print([d[x] for x in d.keys() if type(x) is str and len(x) >= 4])
```

Problema 2

Se dă un dicționar. Să se afișeze (sub orice formă convenabilă) valorile corespunzătoare cheilor care sunt siruri de caractere de lungime cel puțin 4.

R: Cum facem?

R: Ne uităm printre cheile dicționarului

```
ls = d.keys()
```

Verificăm pe fiecare în parte dacă verifică condiția

```
for element in ls:  
    if type(element) is str and len(element) >= 4:  
        print(d[element])
```

*care este diferența între cele două programe?

Tot ce este mai sus poate fi scris într-o singură linie de cod*.

```
print([d[x] for x in d.keys() if type(x) is str and len(x) >= 4])
```

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile listează 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

Î: Cum facem?

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile listează 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

I: Cum facem?

R: Accesăm valorile dicționarului

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile listează 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

Î: Cum facem?

R: Accesăm valorile dicționarului `ls = d.values()`

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile listează 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

I: Cum facem?

R: Accesăm valorile dicționarului `ls = d.values()`

Determinăm maximul **orientativ**

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile listează 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

Î: Cum facem?

R: Accesăm valorile dicționarului

```
ls = d.values()  
mx = 0
```

Determinăm maximul **orientativ**

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile listează 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

I: Cum facem?

R: Accesăm valorile dicționarului

```
ls = d.values()  
mx = 0  
for i in d.values():
```

Determinăm maximul **orientativ**

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile liste de 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

I: Cum facem?

R: Accesăm valorile dicționarului

Determinăm maximul **orientativ**

```
ls = d.values()
mx = 0
for i in d.values():
    if sum(i) > mx:
        mx = sum(i)
```

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile liste de 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

Î: Cum facem?

R: Accesăm valorile dicționarului

Determinăm maximul **orientativ**

```
ls = d.values()
mx = 0
for i in d.values():
    if sum(i) > mx:
        mx = sum(i)
print(mx / 5)
```

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile liste de 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

I: Cum facem?

R: Accesăm valorile dicționarului

Determinăm maximul **orientativ**

Tradiția spune că putem scrie totul
mai eficient :))

```
ls = d.values()
mx = 0
for i in d.values():
    if sum(i) > mx:
        mx = sum(i)
print(mx / 5)
```

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile listează 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

Î: Cum facem?

R: Accesăm valorile dicționarului

Determinăm maximul **orientativ**

Tradiția spune că putem scrie totul mai eficient :))

Î: Ce ar fi să folosim funcția max și comprehensiunea?

```
ls = d.values()
mx = 0
for i in d.values():
    if sum(i) > mx:
        mx = sum(i)
print(mx / 5)
```

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile listează 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

Î: Cum facem?

R: Accesăm valorile dicționarului

Determinăm maximul **orientativ**

Tradiția spune că putem scrie totul mai eficient :))

Î: Ce ar fi să folosim funcția max și comprehensiunea?

```
ls = d.values()
mx = 0
for i in d.values():
    if sum(i) > mx:
        mx = sum(i)
print(mx / 5)
```

```
[sum(x) for x in d.values()]
```

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile listează 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

Î: Cum facem?

R: Accesăm valorile dicționarului

Determinăm maximul **orientativ**

Tradiția spune că putem scrie totul mai eficient :))

Î: Ce ar fi să folosim funcția max și comprehensiunea?

```
ls = d.values()
mx = 0
for i in d.values():
    if sum(i) > mx:
        mx = sum(i)
print(mx / 5)
```

```
max([sum(x) for x in d.values()])
```

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile listează 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

Î: Cum facem?

R: Accesăm valorile dicționarului

Determinăm maximul **orientativ**

Tradiția spune că putem scrie totul mai eficient :))

Î: Ce ar fi să folosim funcția max și comprehensiunea?

```
ls = d.values()
mx = 0
for i in d.values():
    if sum(i) > mx:
        mx = sum(i)
print(mx / 5)
```

```
max([sum(x) for x in d.values()]) / 5
```

Problema 3

Se dă un dicționar care are drept chei numărul matricol al unui student din grupa 102, iar valorile listează 5 elemente, care reprezintă notele studentului respectiv. Să se afișeze media maximă din grupă.

I: Cum facem?

R: Accesăm valorile dicționarului

Determinăm maximul **orientativ**

Tradiția spune că putem scrie totul mai eficient :))

```
ls = d.values()
mx = 0
for i in d.values():
    if sum(i) > mx:
        mx = sum(i)
print(mx / 5)
```

```
print(max([sum(x) for x in d.values()])/5)
```

Problema 4

Se dă un dicționar care reprezintă tabelul de frecvență al unor caractere într-o frază. Să se afișeze caracterele împreună cu frecvența lor în ordine invers lexicografică.

Î: Cum facem?

Problema 4

Se dă un dicționar care reprezintă tabelul de frecvență al unor caractere într-o frază. Să se afișeze caracterele împreună cu frecvența lor în ordine invers lexicografică.

Î: Cum facem?

Idee:

Problema 4

Se dă un dicționar care reprezintă tabelul de frecvență al unor caractere într-o frază. Să se afișeze caracterele împreună cu frecvența lor în ordine invers lexicografică.

Î: Cum facem?

Idee: funcția sorted

Î: Ce trebuie să sortăm de fapt?

Problema 4

Se dă un dicționar care reprezintă tabelul de frecvență al unor caractere într-o frază. Să se afișeze caracterele împreună cu frecvența lor în ordine invers lexicografică.

I: Cum facem?

Idee: funcția sorted

I: Ce trebuie să sortăm de fapt?

R: `d.items !`

Problema 4

Se dă un dicționar care reprezintă tabelul de frecvență al unor caractere într-o frază. Să se afișeze caracterele împreună cu frecvența lor în ordine invers lexicografică.

I: Cum facem?

Idee: funcția sorted

I: Ce trebuie să sortăm de fapt?

R: `d.items !`

`sorted(d.items(),`

Problema 4

Se dă un dicționar care reprezintă tabelul de frecvență al unor caractere într-o frază. Să se afișeze caracterele împreună cu frecvența lor în ordine invers lexicografică.

I: Cum facem?

Idee: funcția sorted

I: Ce trebuie să sortăm de fapt?

R: `d.items !`

```
sorted(d.items(), reverse = True)
```

Problema 4

Se dă un dicționar care reprezintă tabelul de frecvență al unor caractere într-o frază. Să se afișeze caracterele împreună cu frecvența lor în ordine invers lexicografică.

I: Cum facem?

Idee: funcția sorted

I: Ce trebuie să sortăm de fapt?

R: `d.items` !

```
print(sorted(d.items(), reverse = True))
```

Dicționare - Problema #2 testul v1

Fie fișierul *date1.txt* (atașat) ce conține, pe câte o linie, informații despre elevii unei școli: CNP, nume, prenume, clasa, media de admitere (pentru elevii de clasa a 9-a) sau media anului anterior (pentru cei din clasele 10-12), profil.

- a) Să se memoreze datele într-un dicționar ce are drept chei profilurile, iar valoarea asociată fiecărei chei este un dicționar ce are drept chei clasele din respectivul profil. Valoarea asociată fiecărei chei ce identifică o clasă trebuie să conțină toți elevii respectivei clase, împreună cu toate datele lor (CNP, nume, prenume, clasa, medie). Afipați dicționarul obținut. (1,5p)

Dicționare - Problema #2 testul v1 a)

Î: Cum propuneți să structurăm datele? (stabiliti ce colecții folosim și în ce scop)

Dicționare - Problema #2 testul v1 a)

I: Cum propuneți să structurăm datele? (stabiliti ce colecții folosim și în ce scop)

R: next slide

licen = {

"matematică - informatică": { "11 C": [(cnp, nume, ...), (...)] }
"12 B": [(), (), ()...]
⋮
⋮
⋮

"științe - ale - naturii": { "9A": [(), (), ...] }
⋮
⋮

"științe - sociologice": { ... }

Dicționare - Problema #2 testul v1 a)

Pașii principali pt. rezolvarea subpunctului a):

1. Deschidem fișierul cu datele de intrare
2. Citim datele fiecărui elev
3. Le introducem în structura noastră de date
4. Printăm întregul dicționar

Dictionary - Problema #2 testul v1 a)

```
import pprint

liceu = {profil : dict() for profil in ["matematica-informatica", "stiente-ale-naturii", "stiente-sociologice"]}

with open("date1.txt", "r") as file:
    for line in file:
        line = line.strip().split()
        cnp = line[0]
        nume = line[1]
        prenume = line[2]
        clasa = line[3]
        media = float(line[4])
        profilul = line[5]
        tuplu_elev = (cnp, nume, prenume, clasa, media, profilul)
        if clasa not in liceu[profilul]:
            liceu[profilul][clasa] = [tuplu_elev]
        else:
            liceu[profilul][clasa].append(tuplu_elev)
pprint.pprint(liceu)
```

String-uri în Python - introducere

String-urile sunt secvențe de caractere (= bucăți de text). Exemplu:

```
mystr = "Gigel Militaru"
```

String-urile seamănă mult cu liste. Atenție, însă: string-urile sunt **îmmutabile**.

-> Explicați următorul program:

```
mystr = "Gigel Militaru"  
for litera in mystr:  
    if litera.lower() in "aeiou":  
        print(litera, end=" ")
```

Fișiere text - cum lucrăm cu ele

Într-un program Python, putem deschide un fișier folosind funcția built-in open:

```
f = open("fisier.txt", "r")
```

Citirea întregului conținut:

```
tot = f.read()
```

Citirea fiecărei linii:

```
f = open("fisier.txt", "r")  
  
for line in f:  
    print(line)
```

Fișiere text - cum lucrăm cu ele

Scriere + with .. as:

```
with open("fisier.out", 'w') as f:  
    f.write("Hello, World\n")
```

Blocul de cod with .. as e închide automat fișierul f.

Nu mai e nevoie să scriem f.close().

Problema bonus

Se citește de la tastatură un sir de numere reale pozitive v_1, v_2, \dots, v_n (numerele sunt date pe o linie, separate prin spațiu).

Să se determine media aritmetică a numerelor din sir și să se afișeze un mesaj de forma: $(v_1 + \dots + v_n) / n = \text{media}$ unde media este media numerelor din sir afișată cu o zecimală.

Exemplu:

input: 40 15 1.25 1.4

output: $(40 + 15 + 1.25 + 1.4) / 4 = 14.4$

Problema bonus

Soluție (nu vă impacientați; am scris-o aşa special ca să sporim confuzia)

```
lista_cu_nr = [float(x) for x in input("introdu numerele: ").split()]

suma = 0
for nr in lista_cu_nr:
    suma += nr

print("", end="")
print(" + ".join(str(int(nr)) if nr.is_integer() else str(nr) for nr in lista_cu_nr), end="")
print(f" / {len(lista_cu_nr)} = {suma / len(lista_cu_nr):.1f}")
```