

Tema1

October 5, 2022

1 Tema 1

Termen de predare: miercuri, 12 octombrie, ora 20

Notebook-ul va fi rulat de sus în jos, complet, toate celulele.

În caz de eroare, se punctează sarcinile/celulele rezolvate și rulate fără eroare, până în punctul întreruperii execuției.

1. (1p) Se dă o listă de numere. Să se retrurneze lista numerelor pare, în ordine inversă a aparițiilor.

Exemplu: [23, 45, 4, 658, 8, 9, -4] => [-4, 8, 658, 4]

```
[ ]: lista1 = [23, 45, 4, 658, 8, 9, -4]
      # variabila rezultat va fi rezultat_lista_pare

      assert rezultat_lista_pare == [-4, 8, 658, 4], "Rezultat asteptat: [-4, 8, 658, ↵
      ↵4]"
```

2. (1p) Se dă o listă de numere. Să se afișeze: suma dintre maximul numerelor pare și minumul numerelor impare.

```
[ ]: lista2 = [4, 5, 6, 78, 43, 57, 68]

      assert rezultat == 83, "Rezultat asteptat: 83"
```

3. (1p) Fiind data o lista de cuvinte, să se returneze lista care contine cvintele care NU incep cu litera L, sortate in ordine alfabetica.

```
[ ]: lista_cuvinte=["lamaie", "maxim", "alfabet","carte", "inima", "barca", ↵
      ↵"lacusta", "are"]

      assert cuvinte_fara_L_alfabetic == ['are', 'barca', 'carte', 'inima', 'maxim'], ↵
      ↵"Rezultat asteptat: ['are', 'barca', 'carte', 'inima', 'maxim']"
```

4. (2p) Se da o listă de dicționare în care cheile sunt mâncăruri iar valorile sunt ingrediente. Să se returneze:
 1. Felurile de mâncare care nu conțin un ingredient dat.
 2. Felurile de mâncare care conțin (dar nu coincid cu) o lista de ingrediente data.

3. Felul de mâncare cu cele mai multe ingrediente.
4. Toate perechile de mâncăruri A, B pentru care lista de ingrediente a lui A este inclusă în (dar nu coincide cu) lista de ingrediente a lui B.

Indicație: folosiți tipul de date *set*.

```
[ ]: mancaruri = {
    "tocana": ["cartofi", "carne", "rosii", "ceapa", "ulei"],
    "cartofi_prajiti": ["cartofi", "ulei"],
    "ciorba_vacuta": ["ulei", "ceapa", "ulei", "carne", "rosii", "cartofi",
↳ "morcovi", "telina"],
    "savarina": ["oua", "faina", "zahar", "frisca"],
    "musaca": ["carne", "oua", "branza", "ceapa", "smantana"]
}
# 1. Felurile de mancare care nu contin un ingredient dat.
ingredient = "carne"
mancare_fara_ingredient = ...

assert mancare_fara_ingredient == ['cartofi_prajiti', 'savarina'], "Rezultat_
↳ asteptat: ['cartofi_prajiti', 'savarina'] "

# 2. Felurile de mancare care contin (dar nu coincid cu) o lista de ingrediente_
↳ data.

ingrediente_cerute=["carne", "oua"]
# mancaruri-cu_ingredient_cerute = ['musaca']

# 3. Felul de mâncare cu cele mai multe ingrediente.
# mancare_cu_ingredient_max = ['ciorba_vacuta']

# 4. Toate perechile de mâncăruri A, B pentru care lista de ingrediente a lui A_
↳ este inclusă în (dar nu coincide cu)
# lista de ingrediente a lui B.
# raspuns: (tocana, ciorba_vacuta), (cartofi_prajiti, tocana),_
↳ (cartofi_prajiti, ciorba_vacuta)
```

5. (2p) Fiind dată o listă de cuvinte:
 1. Înlocuiți vocalele cu '*'.
 2. Determinați cuvintele cu cele mai multe vocale.
 3. Înlocuiți consoanele cu prima literă a cuvântului.
 4. Determinați cuvintele ce conțin cel puțin 3 litere comune cu un cuvânt dat.

```
[ ]: cuvinte = ['ana', 'mere', 'elena']
# 1. Înlocuiți vocalele cu '*'.
# răspuns: ['*n*', 'm*r*', '*l*n*']

# 2. Determinați cuvintele cu cele mai multe vocale.
# răspuns: elena
```

```
# 3. Înlocuiți consoanele cu prima literă a cuvântului.
# răspuns: ['aaa', 'meme', 'eeeea']
```

```
# 4. Determinați cuvintele ce conțin cel puțin 3 litere comune cu un cuvânt dat.
# cuvânt_dat = 'lenes' => răspuns: 'elena'
# cuvânt_dat = 'camere' => răspuns: 'mere'
```

6. (1p) Fiind dat un dicționar în care cheia este un cuvânt, iar valoarea este partea de propoziție (funcții sintactice), returnați toate combinațiile care pot deveni propoziții, de forma : S+P+C sau S+A+P+C sau S+P sau S+A+P unde
- S -> subiect
 - A -> atribut
 - P -> predicat
 - C -> complement

```
[ ]: propozitii={
    'Ana' : 'S',
    'are' : 'P',
    'mere' : 'C'
    'rosii' : 'A',
    'sunt' : 'P',
    'acolo': 'C',
    'copiii': 'S'
}
# răspuns:
# 'Ana' + 'are' + 'mere'
# 'Ana' + 'are' + 'acolo'
# 'copiii' + 'sunt' + 'acolo'
# 'copiii' + 'sunt'
# 'copiii' + 'rosii' + 'sunt' + 'acolo'
# etc.
```

7. (1p) Să se verifice dacă o propoziție dată este palindrom sau nu (fără a lua în considerare semnele de punctuație sau diacriticele).

Exemple:

1. ELE FAC CAFELE.
2. ALA E SERGIU, NU-I GRESEALA.
3. AI RAMAS ACASA, MARIA?

```
[ ]: 
```