

1 Context

Formati echipe de cate 2 persoane. Fiecare echipa va face un proiect *privat* pe github, in care voi fi invitat (github id: lmsasu).

Tema de mai jos este obligatorie si face parte in nota finala. Tema se va incarca pe github, in repo-ul privat si pe elearning in lucrarea deschisa, pana in 1 martie ora 23.00.

2 Tema 1

1. (1p) Se da o lista de numere. Sa se afiseze diferenta dintre: suma celor mai mari 3 numere si produsul celor mai mici 2 numere.

lista = [10, 3, 5, 20, -4, 9, 1] -> 43

lista = [0, -1, -2, -3, -4] -> -15

2. (1p) Se da o lista de cuvinte. Sa se determine cuvintele unice (fara diferenta intre litere mici si mari) si de cate ori apare fiecare.

Exemple:

['mouse', 'and', 'branza', 'vieux', 'vieux', 'vieux', 'me', 'slow', 'mouse', 'This', 'mouse', 'out', 'in', 'want', 'branza', 'song', 'barza', 'barza', 'barza', 'makes', 'to', 'in', 'branza', 'run', 'motion'] -> {'mouse': 3, 'and': 1, 'branza': 3, 'vieux': 3, 'me': 1, 'slow': 1, 'This': 1, 'out': 1, 'in': 2, 'want': 1, 'song': 1, 'barza': 3, 'makes': 1, 'to': 1, 'run': 1, 'motion': 1}

['doi', 'doi', 'trei', 'Trei', 'un'] -> {'doi': 2, 'trei': 2, 'un': 1}

3. (1p) Se da o lista de cel putin 10 numere. Care 3 numere din lista, inmultite, dau produsul maxim?

Exemple:

[10, 1, 2, 9, 7, 8, 3, 6, 4, 5] -> p_max=720, solutie=(10, 9, 8)

[1, -2, -3, -4, -5, 0] -> p_max=20, solutie=(1, -4, -5)

4. (1p) Dandu-se o lista de numere si un numar n natural, $n \geq 1$, sa se determine toate tripletele a, b, c cu $a < b$ din lista, pentru care $a^n + b^n = c^n$.

Exemple:

lista = [12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2], n=2 -> 3 4 5; 6 8 10

lista = [90, 10, 20, 30, 40, 50], n=3 -> se afiseaza 'Nu exista solutie'

5. (1p) Dandu-se o propozitie, sa se determine transcrierea ei in "pasareasca" (o vocala se transforma in vocala + 'P' + vocala



Sa se scrie un program care face traducerea inversa (daca stringul dat provine din pasareasca; daca nu, semnaleaza acest lucru).

Exemplu de traducere in pasareasca: acasa-apacapasapa, prietena-pripitepenapa.

6. (2p) Se da un dictionar in care cheile sunt nume de studenti, iar valorile sunt liste de carti preferate. Sa se determine:
- Care sunt studentii care au in lista de preferinte o carte specificata?
 - Care sunt perechile de studenti care au aceleasi preferinte de carti (ordinea in lista de preferinte nu e relevanta)
 - Pentru un student dat, care sunt studentii cu care are cele mai multe carti comune in liste de preferinte; daca sunt mai multi astfel de studenti cei mai apropiati, se vor enumera toti.
 - Pentru doi studenti, A si B, care e lista de stergeri si adaugari prin care lista lui A devine identica cu lista lui B (ordinea in lista nefiind importanta)
 - Toate perechile de studenti A, B pentru care lista de preferinte a lui A este inclusa in (dar nu coincide cu) lista de preferinte a lui B.

Indicatie: considerati tipul de date set.

Exemplu:

```
preferinte = {
    'Popescu': ['carte2', 'carte1', 'carte3', 'carte4'],
    'Ionescu': ['carte1', 'carte2', 'carte3', 'carte7', 'alta carte'],
    'Georgescu': ['carte1', 'alta carte'],
    'Xulescu': ['carte2', 'carte5', 'carte6', 'carte7'],
    'Dragomir': ['carte4', 'carte1', 'carte2', 'carte3']
}
```

- carte = 'alta carte' -> ['Ionescu', 'Georgescu']
- Popescu si Dragomir au aceleasi preferinte: ['carte2', 'carte1', 'carte3', 'carte4']. Se remarca faptul ca nu se afiseaza si perechea echivalenta: Dragomir si Popescu.
- Xulescu -> Suprapunere maxima de preferinte: 2 cu colegii: ['Ionescu']
- A='Popescu', B='Ionescu' -> Se vor adauga cartile: {'carte7', 'alta carte'}, se vor scoate cartile: {'carte4'}
- Preferintele lui Georgescu sunt subset propriu al preferintelor lui Ionescu

7. (1p) Se da un dictionar cu persoane, si pentru fiecare persoana o lista de intervale de timp de disponibilitate. O disponibilitate incepe la o anumita ora fixa (ex: 17) sau ora fixa+30 de minute si dureaza exact 30 de minute. Exemplu:

```
doodle = {
    "Popescu": ["13", "13.30", "16", "17"], # Popescu este disponibil intre: 13 si 13.30, 13.30-14, 16-16.30, 17-17.30
    "Ionescu": ["13", "13.30", "16", "17", "17.30", "11", "15"],
    "Maximilian": ["14", "14.30", "15", "15.30", "16", "16.30", "19", "19.30"]
}
```

Sa se determine toate intervalele in care toate persoanele sunt disponibile. Daca nu exista niciun astfel de interval, atunci sa se determine care sunt intervalele de timp in care numarul maxim de persoane poate participa. Daca sunt mai multe intervale pentru acest numar maxim de persoane, se vor da toate solutiile. In fiecare caz se da lista de persoane participante si intervalele orare.

Nota: se cauta intervale de timp cu cat mai multi participanti de 30 de minute.

8. (2p) Se da o lista de dictionare de forma:

```
[  
  {"nume": "Popescu",  
   "gen": feminin,  
   "varsta": 30,  
   "salariu": 2000,  
   "animal preferat": "pisica",  
  },  
  {"nume": "Irimia",  
   "varsta": 40,  
   "gen": "masculin",  
   "salariu": 3000,  
   "animal preferat": "caine"  
  },  
  # etc  
]
```

Sa se determine:

- A. salariul mediu, minim si maxim
- B. cei care au varsta intr-un anumit interval dat [a, b]
- C. care este animalul cu cele mai multe preferinte. Daca sunt mai multe astfel e animale, atunci se vor da toate
- D. cate persoane de gen feminin si respectiv masculin sunt in lista

Nota: Toate dictionarele au aceleasi chei: nume etc.

In []: