**1. List comprehensions:**

1) Sa se genereze si afiseze lista numerelor impare intre 0 si 10

2) Sa se genereze lista literelor in lowercase de la a la z

3) Sa se scrie o functie care primeste ca parametru un numar n. Sa se genereze lista de forma 1,-2,3,-4,... pana la n(cu semnul corespunzator).

4) Sa se scrie o functie care primeste ca parametru o lista de numere. Sa se returneze lista cu elementele impare.

5) Sa se scrie o functie care primeste ca parametru o lista de numere. Sa se returneze lista cu elementele de pe pozitiile impare.

6) Sa se scrie o functie care calculeaza pentru o lista de numere data, lista continand elementele care au aceeasi paritate cu pozitia pe care se afla. De exemplu, pentru lista [2,4,1,7,5,1,8,10], lista calculata va contine elementele: 2, 7, 1, 8.

7) Sa se scrie o functie care primeste o lista si returneaza lista cu perechiile (tupluri) de elementele de pe pozitii vecine. De exemplu pentru lista [1,2,3,4] lista rezultata ar fi [(1,2),(2,3),(3,4)]

8) Sa se scrie o functie care primeste ca parametru un numar n si genereaza cu ajutorul list comprehension o lista de n liste. In fiecare lista element vom avea siruri de forma "x\*y=rez". Elementul x va avea drept valoare indicele listei-element iar y va varia intre 1 si n (sirurile fiind ordonate crescator). Practic se genereaza tabla inmultirii numerelor de la 1 la n. Sa se apeleze functia si sa se afiseze rezultatul.

9) Sa se scrie un list comprehension prin care, pornind de la un sir dar, de exemplu, sir="abcde" sa se obtina intr-o lista toate sirurile formate luand n litere de la inceputul sirului si concatenandu-le la sfarsit, cu n de la 0 pana la lungime(sir)-1. Pentru sirul dat ca exemplu, lista de calculat ar fi: ['abcde', 'bcdea', 'cdeab', 'deabc', 'eabcd']

10) Sa se scrie o functie care returneaza o lista de n liste (unde n e dat ca parametru in functie), cu proprietatea ca prima lista va fi vida, a doua lista va avea un singur element egal cu 1, a treia lista va avea doua elemente egale cu 2, si asa mai departe pana la a n-a lista care va avea n-1 elemente egale cu n-1. De exemplu pentru n=4, lista de liste va fi: [[],[1],[2,2],[3,3,3]]

**2. Sortari liste:**

a) Se considera o lista de numere. Sa se sorteze folosind o functie lambda asa cum s-ar fi sortat daca numerele erau siruri

b) Se considera o lista de numere. Sa se sorteze folosind o functie lambda comparand intai ultima cifra apoi penultima (asa cum am fi sortat daca erai siruri cu literele in ordine inversa.

c) Se considera o lista de numere. Sa se sorteze dupa lungimea numarului.

d) Se considera o lista de numere. Sa se sorteze dupa numarul de cifre distincte.

3. Se da o lista de cuvinte (de exemplu: ["papagal", "pisica","soarece","bolovan","soparla","catel", "pasare"]).

a) (generare matrice, adaugare de elemente) Sa se scrie o functie gen\_matrice() Functia genereaza (indiferent de susbsetul de litere folosite in cuvinte) matricea avand liniile si coloanele corespunzatoare tuturor literelor din alfabet (pentru simplitate folosim alfabetul englezesc - deci fara caractere Unicode speciale precum litere cu diacritice). Prima linie si prima coloana vor contine chiar litere alfabetului in ordine, restul matricii va fi completatv cu 0.

b) (modificare matrice) Sa se scrie o alta functie, completeaza\_matrice(), care primeste ca parametru lista de cuvinte si matricea creata mai devreme. Functia va modifa matricea astfel incat un element de pe pozitia i+1,j+1 reprezinta numarul de dati in care litera corespunzatoare liniei i s-a aflat intr-un cuvant in urma literei corespunzatoare coloanei j. Se va apela functia si se va afisa matricea.

c) (eliminare de linii si coloane) Sa se elimine din matrice liniile si coloanele cu toate elementele 0. Sa se afiseze matricea.

d) (parcurgere matrice) Sa se afiseze perechile (distincte) de litere cu numarul de aparitii mai mare sau egal decat un N (dati voi o valoare, de exemplu 2).

4. Se da o lista de cuvinte in care unele apar de mai multe ori. Se considera aparitie dubla chiar si cand difera tipul literelor (litera mica vs litera mare). De exemplu haha si HahA sunt considerate drept aparitii ale aceluiasi cuvant. Exemplu de lista: ["haha", "poc", "Poc", "POC", "haHA", "hei", "hey", "HahA", "poc", "Hei"] Sa se afiseze pentru fiecare cuvant de cate ori a aparut in lista. Folositi un dictionar pentru a numara usor.

5. Se da o lista de siruri, de exemplu: ["bau-bau", "bobocel", "14 pisici", "1pitic", "pisicel", "botosel", "414", "ham", "-hau", "bob", "bocceluta"].

a) (creare dictionar) Creati un dictionar (enumerandu-i elementele) cu cheile si valorile urmatoare (grupate in paranteze, unde primul element e cheia si al doilea e valoarea): ((1,4),"mic"), ((4,8), "mediu"), ((8,15),"mare")

b) (adaugare in dictionar) Sa se calculeze un dictionar pe baza listei de siruri, unde cheile sunt caractere, iar valoarea corespunzatoare pentru fiecare cheie o reprezinta lista cu sirurile care contin acea litera)

c) (stergere elem din dictionar) Din dictionarul rezultat sa se elimine cheile formate din caractere nealfanumerice (si sa se afiseze perechile de cheie-valoare eliminate).

d) (lungime dictionar) Sa se afiseze cate chei au mai ramas in dictionar

e) (parcurgere chei/valori) Sa se parcurga dictionarul cu cheile caractere, si pentru fiecare caracter sa se afiseze tipurile (mic, mediu, mare) de siruri in care acesta apare. Tipul de sir e dat de al doilea dictionar cu cheile tupluri. Pentru un tuplu t=(n1,n2), un sir se considera ca e de tipul asociat tuplului respeciv daca lungimea sirului este mai mare sau egala cu n1 si mai mica strict decat n2.

**6. List comprehensions** (matrice = liste de liste)

a) Sa se scrie o functie care primeste un parametru n si genereaza o matrice de forma (in exemplu e pentru n=4): 0 1 2 3 1 2 3 4 2 3 4 5 3 4 5 6

Elementul m[i][j] este egal cu i+j

Sa se apeleze functia si sa se afiseze rezultatul.

b)Sa se scrie o functie care primeste ca parametri o matrice m de intregi si un numar n si returneaza matricea rezultata prin copierea doar a liniilor cu suma elementelor mai mare decat n.

Sa se apeleze functia si sa se afiseze rezultatul.

c) Sa se scrie o functie care primeste doua liste de numere, l1 si l2 si returneaza o matrice cu numarul de linii egal cu numarul de elemente din l1 si numarul de coloane egal cu numarul de elemente din l2. Matricea pe fiecare element de pe linia i, coloana j, va avea doar valori de forma:

"<" (daca elementul de pe pozitia i din l1 e mai mic decat elementul de pe pozitia j din l2)

"==" (daca elementul de pe pozitia i din l1 e egal cu elementul de pe pozitia j din l2)

">" (daca elementul de pe pozitia i din l1 e mai mare decat elementul de pe pozitia j din l2)