Laborator 3

Tratarea exceptiilor

https://docs.python.org/3/tutorial/errors.html

 # Structura generala

```
try:
    # Some Code
except:
    # Executed if error in the try block
else:
    # execute if no exception
finally:
    # Some code .....(always executed)
```

 # Tratarea mai multor exceptii

```
try:
  # do something
  pass
except ValueError:
  # handle ValueError exception
  pass
except (TypeError, ZeroDivisionError):
  # handle multiple exceptions
  # TypeError and ZeroDivisionError
  pass
except:
  # handle all other exceptions
  pass
```

 # Aruncarea manuala a exceptiilor

```
try:
    a = int(input("Enter a positive integer: "))
    if a <= 0:
        raise ValueError("That is not a positive number!")
except ValueError as ve:
    print(ve)
```

Exercitii – laborator 3

1) List comprehensions:

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

b) Pentru un numar natural n citit de la tastatura, sa se genereze lista de forma 1,-2,3,-4,... pana la n (cu semnul corespunzator).

c) Se da o lista de numere. Sa se obtina lista cu elementele impare din lista data.

d) Se da o lista de numere. Sa se obtina lista cu elementele aflate pe pozitii impare in lista data.

e) Sa se obtina, 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.

f) Sa da o lista. Sa se obtina 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)]

g) 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.

h) Sa se obtina lista cu toate permutarile circulare ale unui sir dat. De exemplu, pentru sir="abcde" vom obtine lista ['abcde', 'bcdea', 'cdeab', 'deabc', 'eabcd'].

i) 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, etc. (asa cum am fi sortat daca erai siruri cu literele in ordine inversa.

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

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

e) sortare numere intai dupa lungime si apoi dupa valoare.

OPERAȚII CU LISTE

Problemele din seminar:

https://docs.python.org/3/tutorial/datastructures.html

1. Se dau două liste l1 si l2 de lungime n. Să se înlocuiască elementele de pe poziții pare din l1 cu cele de pe poziția corespunzătoare din l2 folosind feliere (slice)
2. Se dă o listă de numere naturale. Să se șteargă din listă subsecvența delimitată de primele două zerouri din listă (inclusiv zerourile)
3. Se dă o listă de numere naturale. Să se șteargă din listă toate zerourile
4. Se dă o listă de numere naturale și un număr natural k. Să se elimine din listă subsecveța de lungime k de sumă minimă (dacă sunt mai multe se va elimine prima = cea mai din stânga) – fără a folosi liste suplimentare
5. Se dă un vector de numere naturale ordonat crescător (toate elementele sale se vor da pe o linie separate prin spațiu). Să se elimine duplicatele din vector.
6. Se dă o listă de numere reale (toate elementele sale se vor da pe o linie separate prin spațiu). Să se insereze câte un 0 după fiecare element negativ (fără a folosi liste suplimentare)

SORTĂRI

1. Se citește o propoziție cu cuvinte separate prin spațiu. Să se formeze o nouă propoziție cu cuvintele din prima propoziție care au lungime cel puțin 2 ordonate descrescător după lungime.
2. Se citește un vector de numere naturale (cu elementele date pe o linie, separate prin spațiu). Să de ordoneze elementele din vector crescător după suma cifrelor, iar în caz de egalitate, descrescător după valorile lor
   v = [11, 45, 20, 810, 179, 81, 1000] => v = [1000, 20, 11, 810, 81, 45, 179]
3. Se citesc un număr natural n și următoarele informații despre n elevi: nume (fără spații), prenume (fără spații), grupa, o lista de note (numere naturale). Informațiile despre fiecare student se dau pe linii separate:
   3
   Marineanu Maria 22 10 9 5
   Mihaliu Dan 22 4 5 10 10
   Podaru Ilie 21 10 10 8 8
   a) Citiți datele despre studenți și memorați-sub forma:
   [['Marineanu', 'Maria', 22, [10, 9, 5]], ['Mihaliu', 'Dan', 22, [4, 5, 10, 10]], ['Podaru', 'Ilie', 21, [10, 10, 8, 8]]]
   b) Adăugați la fiecare student situația sa școlară: promovat (True) sau nepromovat (False). Pentru a fi considerat promovat, un student trebuie să aibă toate notele mai mari sau egale cu 5.
   c) Afișați studenții ordonați pe grupe crescător, iar în cadrul fiecărei grupe ordonați alfabetic.
   d) Afișați studenții ordonați pe grupe crescător, iar în cadrul fiecărei grupe se vor afișa întâi studenții promovați ordonați descrescător după medie (și în caz de egalitate după

nume), apoi cei nepromovați ordonați crescător după numărul de note mai mici decât 5 (de restanțe).

e) Determinați studenții cu media maximă folosind funcția max care are și ea parametrul key

4. Să se sorteze o listă de numere naturale astfel încât numerele pare sortate descrescător să fie poziționate după cele impare sortate crescător.

MATRICE, VECTORI

1. Se citesc m, n și o matrice cu m linii și n coloane, elementele unei linii fiind date pe o linie (elementele unei linii date pe o linie separate cu spațiu). Să se construiască în memorie și să se afișeze matricea transpusă (folosind și comprehensiune).
2. Se citesc m, n și o matrice cu m linii si n coloane (numerele sunt date câte unul pe linie). Să se ordoneze crescător elementele de pe prima coloana prin interschimbări de linii și să se afișeze matricea obținută (fiecare element se va afișa pe 5 caractere).
3. Se da un număr natural n>2. Să se afișeze primele n linii din triunghiul lui Pascal (daca c este numărul maxim de cifre ale unui număr din triunghi, toate numerele se vor afișa pe c+1 caractere)
4. **Ciurul lui Eratostene.** Se dă un număr natural n. Să se creeze o listă cu numerele prime mai mici sau egale cu n.
5. Se dau două mulțimi cu elementele ordonate crescător (câte una pe linie). Să se determine eficient reuniunea și intersecția celor două mulțimi (fără a folosi set).