**Probleme laboratoare seria 10**

**Soutiile vor fi incarcate colorate cu rosu (inseamna ca deocamdata nu au fost verificate de profesor).**

**Solutiile verificate si care sunt corecte vor fi colorate de catre profesor cu verde.**

**O posibila solutie optima (din punct de vedere al modului de codare - cod compact, etc.) va fi colorata cu albastru.**

**Laborator 2**

Operații pe biți: nu intră la examen (dar este bine să le știți).

**Laborator 3**

Avem nevoie de import math și math.sqrt(x).

*A)* *Algoritmi elementari*

1. Fie x număr natural, citit. Să se verifice dacă x este:

a. pătrat perfect (fără structuri repetitive);

b. număr prim;

c. număr perfect (egal cu suma divizorilor lui, mai puțin numărul însuși; ex.: 6, 28, 496, 8128);

d. palindrom (este egal cu răsturnatul său).

2. Fie n număr natural, citit. Să se afișeze:

a. toate numerele prime mai mici decât n;

def eratostene(nr):

l=list(range(2, nr + 1))

i=0

while i<len(l):

l=list(filter(lambda x: x%l[i] !=0 or x==l[i], l))

i+=1

return l[:-1]

n=int(input())

print(eratostene(n))

b. primele n numere naturale prime;

c. primele n pătrate perfecte;

d. toate pătratele perfecte mai mici decât n.

3. Se citește un șir de valori întregi până la întâlnirea lui 0. Să se afișeze:

a. numărul valorilor de tip “deal” / „vale” (ce au la dreapta și la stânga valori mai mici / mai mari; pentru primul element din șir se va lua în considerare doar valoarea din dreapta, iar pentru ultimul element numai valoarea din stânga);

b. valoarea maximă din șir;

c. media aritmetică a valorilor pozitive.

*B)* *Din nou operații pe biți...*

Rezolvați următoarele cerințe folosind operații pe biți.

1. Dacă un număr natural este de forma să se afișeze puterea *k.*

2. Se citește n, natural. Afișați cea mai mare putere a lui 2 care îl divide pe n.

3. Să se determine în mod eficient numărul de biți nenuli din reprezentarea binară a unui număr natural.

4. Să se genereze toate submulțimile mulțimii , unde numărul natural nenul

se citește de la tastatură *(fără backtracking).*

5. Fie și două numere naturale. Calculați numărul biților din reprezentarea binară internă a numărului a căror valoare trebuie comutată pentru a obține numărul .

*C)* *Liste*

1. Stiind că ordinale = ["primul", "al doilea", "al treilea", "al patrulea", "al cincilea", "al saselea", "al saptelea", "al optulea", "al noulea"], care este outputul urmatoarelor comenzi:

a) print(ordinale[7:5])

[ ]

b) print(ordinale[0:1000])

Intreaga lista (de la indicele i=8 nu mai exista elemente de afisat)

c) print(ordinale[-4:-1])

al saselea al saptelea al optulea

d) print(ordinale[-4:0])

[ ]

e) print(ordinale[-4:])

al saselea al saptelea al optulea al noulea

f) print(ordinale[-1:-4])

[ ]

g) print(ordinale[::])

Intreaga lista

h) print(ordinale[::100])

[‘primul’]

i) print(ordinale[::-100])

[‘al noulea’]

Răspundeți, justificați și apoi rulați pentru a vă verifica răspunsurile.

2. Stiind că l = [1, 2, 3], care este outputul următorului fragment de cod, rulat în ordinea aceasta:

In[1]: l[len(l):] = [4, 5, 6]

print(l)

l= [1,2, [4,5,6]]

In[2]: l[:0] = [-3, -2, -1, 0]

print(l)

[-3,-2,-1,0,1,2,3]

Răspundeți, justificați și apoi rulați pentru a vă verifica răspunsurile.

3. Fie l o listă oarecare. Completați codul l[\_:\_:-1] în locurile marcate cu "\_" cu expresii (nevide) potrivite astfel încât rezultatul rulării acestuia să fie identic cu cel al rulării codului l[::-1].

l[len(l):0:-1]

**Laborator 4**

1. Care este outputul următoarelor comenzi:

a) **print([1, 2, 3] \* 1)**

**[1,2,3]**

b) **print([1, 2, 3] \* 0)**

**[]**

c) **print([1, 2, 3] \* -1)**

**[]**

Răspundeți, justificați și apoi rulați pentru a vă verifica răspunsurile.

2. Care este efectul rulării următorului cod?

**for i in range(10):**

**print("i =",i)**

**i = "Ana are mere"**

**print(" i =",i)**

**Se afiseaza:**

**0 1 2 3 4 5 6 7 8 9**

**Ana are mere**

3. Care dintre codurile următoare au ca efect inserarea jumătății numărului după orice număr par dintr-o listă l, fără a folosi liste auxiliare? Justificați, scrieți ce se va afișa pentru următoarele liste l = [1, 14, 18, 23], l = [1, 14, 18, 23, 6, 24], l = [1, 24, 18, 23, 12] și apoi rulați pentru a vă verifica răspunsurile.

**a)**

**for i in range(len(l)):**

**if l[i] % 2 == 0:**

**l.insert(i+1, l[i] // 2)**

**i = i + 1**

**i = i + 1**

**print(l)**

**b)**

**i = 0**

**while i < len(l):**

**if l[i] % 2 == 0:**

**l.insert(i+1, l[i] // 2)**

**i = i + 1**

**print(l)**

**c)**

**i = 0**

**while i < len(l):**

**if l[i] % 2 == 0:**

**l.insert(i+1, l[i] // 2)**

**i = i + 1**

**i = i + 1**

**print(l)**

**d)**

**for el in list(l):**

**if el % 2 == 0:**

**i = l.index(el)**

**l.insert(i+1, l[i] // 2)**

**print(l)**

4. Fie lista **l\_1 = [1, 2, 240, 120, 2, 3, 18, 12, 22, 28, 1004].** Să se șteargă:

a. toate valorile pare;

b. toate valorile prime.

5. Se citește **n**, apoi **n** numere. Să se rearanjeze numerele și să se memoreze într-o listă astfel încât toate valorile nule să fie la finalul acesteia.

6. Ciurul lui Eratostene: memorați toate numerele numerele prime mai mici decât o valoare citită **x**.

7. Fie listele **l\_1, l\_2** și **n, m, k<=n\*m** numere naturale, citite. Se citesc **n, m**  valori întregi ce se memorează în lista **l\_1**, respectiv **l\_2**. Să se afișeze primele **k** perechi **(l1, l2), l1l\_1, l2l\_2**, ce au suma minimă.

8. Fie o listă ce conține liste sau tupluri încuibate (ex.: l = [1, [2,3], [4, 5, (6, (7, 8, 9))], 10]. Să se transforme într-o listă simplă, fără a folosi liste suplimentare:

a) ignorând ordinea elementelor;

b) păstrând ordinea elementelor.

**Laborator 5**

1) Care este outputul următoarelor comenzi:

a) print([n \*\* 2 for n in (0, 1, 2, 3)])

b) print((n \*\* 2 for n in (0, 1, 2, 3) if n > 0))

c) print({n \*\* 2 for n in range(-3, 3)})

d) print({n \*\* 2 : n for n in range(-3, 3)})

Răspundeți, justificați și apoi rulați pentru a vă verifica răspunsul.

2) Care este outputul următoarelor comenzi:

a) print({1, 2, 3} \* 2)

b) print({1, 2, 3} \* 0)

c) print({1, 2, 3} \* -2)

d) print({1, 2, 3} + {4, 5, 6})

e) print({1, 2, 3} + {0, 1, 2})

Răspundeți, justificați și apoi rulați pentru a vă verifica răspunsul.

3) Care este diferența dintre copierea simplă și copierea profundă a mulțimilor? Argumentați.

4) Se citește un șir de n numere întregi. Să se afișeze:

a. câte perechi de numere identice se pot forma din numerele șirului. Ex: pentru n = 11 și șirul 3, 6, 4, 3, 2, 6, 3, 4, 5, 3, 6 se pot forma 4 perechi: (3,3), (3,3), (4,4), (6,6).

b. numerele întregi cuprinse între valoarea maximă și cea minimă care lipsesc din șir.

5) Se citește n, apoi două șiruri având câte n numere întregi (valorile se pot repeta). Să se afișeze:

a) valorile comune celor două șiruri;

b) valorile comune celor două șiruri (cu tot cu repetiții). Ex.: pentru n = 8 și șirurile s1: 1, 2, 2, 3, 4, 4, 4, 5 și 2, 2, 2, 3, 4, 4, 5, 5 se va afișa: 2, 2, 3, 4, 4, 5

c) valorile care apar în primul șir și nu se află în cel de-al doilea.

6) Scrieți un program care să extragă duplicatele dintr-o listă l într-o alta, numită dubluri.

7) Scrieți un program care:

a) realizează un tabel de frecvențe pentru literele mici dintr-un text introdus de la tastatură.

Exemplu:

ana are mere

e : 3

n : 1

a : 3

m : 1

r : 2

b) cerința a) cu tabelul ordonat descrescător după numărul de apariții;

c) cerința b) astfel încât pentru caractere cu frecvențe egale să se aplice ordonarea alfabetică.

Pentru problema 7 putem folosi (reamintim):

· ord(c): codul ASCII al caracterului c;

· chr(x): caracterul cu codul ASCII x.

8) Se citește de la tastatură un număr natural n, apoi, pentru n studenți, numele și grupa.

a) să se afișeze grupele, în ordine crescătoare și pentru fiecare grupă studenții care o compun;

b) să se citească de la tastatură un nume de student și să se afișeze grupa din care face parte.

**Laborator 6**

1. Se citește un text ce conține numai litere mici ale alfabetului englez și spații. Să se afișeze textul cifrat cu:

a. cifrul lui Cezar (deplasare la dreapta cu 3 poziții);

b. deplasarea la dreapta cu n poziții (n citit).

2. Să se elimine toate vocalele dintr-un text.

3. Să se traducă un text în limba păsărească: după fiecare vocală se va insera grupul format din litera ‘p’ și vocala respectivă. Exemplu: pentru textul *Ana are mere* traducerea va fi *Apanapa aparepe meperepe.*

4. Se citește o linie de text. Să se afișeze suma numerelor din text. Se cunoaște că numerele pot fi naturale sau zecimale pozitive (fără semn) și sunt precedate și urmate de spațiu. Exemplu: Ana are 10.5 kg. de mere, 15 kg. de gutui si 5.75 kg. de prune.

5. Se citește un text ce conține litere și semnele de punctuație uzuale.

a) Să se afișeze cuvintele, în ordinea descrescătoare a frecvenței de apariție.

Exemplu: pentru textul *Ana?...Ana are multe, multe mere!... Dar tu,tu cate mere ai?* Se va afișa

Ana : 2

multe : 2

mere : 2

tu : 2

are : 1

Dar : 1

cate : 1

ai : 1

b) Problema a, iar cuvintele cu aceeași frecvență de apariție vor fi afișate în ordine alfabetică.

6. Se citește n ce reprezintă numărul de elevi ai unei școli. Pentru fiecare elev se citesc și se memorează următoarele câmpuri:

· *CNP;*

· *nume și prenume;*

· *medie;*

· *număr de absențe;*

· *număr de frați.*

a) Să se realizeze o corectare a datelor de intrare pentru câmpul ce conține *numele și prenumele*, prin eliminarea spațiilor inutile și scrierea tuturor numelor și prenumelor cu majusculă.

Exemplu: “ dumitrescu - popa liviu - daniel “ se va transforma în “Dumitrescu-Popa Liviu-Daniel”

b) Să se afișeze numărul elevilor ce au unul dintre prenume “Andrei”.

c) Să se adauge câmpul *bursa* și să se completeze conform regulii: elevii care au mai puțin de 10 absențe și au media mai mare decât 9.50 primesc bursa în valoare de 1000 de lei. Pentru ceilalți se va completa cu textul “fara bursa”.

d) Pentru elevul cu CNP-ul 1960707150180 să se modifice bursa la 2000 de lei.

e) Să se șteargă, pentru fiecare elev, câmpul ce reține numărul de frați.

7. Modificați programul de la problema 6) astfel încât informațiile memorate despre elevi să fie grupate astfel:

o Date personale:

§ *CNP;*

§ *nume și prenume;*

§ *număr de frați.*

o Date școlare:

§ *medie;*

§ *număr de absențe;*

§ *bursa*

Observație: puteți folosi dicționare imbricate.

**Laborator 7**

1. Scrieți funcția *interschimbare* care primește ca parametri două numere x, y si le interschimbă.

2. Scrieți o funcție care verifică dacă un număr natural primit ca parametru este palindrom (returnând True / False).

3. Scrieți o funcție care verifică dacă două șiruri de caractere primite ca parametru:

a. rimează (au ultimele două caractere egale);

b. sunt unul anagrama celuilalt.

4. Scrieți o funcție care primește ca parametru un text și returnează lista cuvintelor.

5. Se citește un text ce conține semne de punctuație și litere ale alfabetului englez. Să se afișeze numărul de cuvinte ce rimează cu ultimul cuvânt din text.

*Exemplu*: Alina, masina trebuie sa vina... Da-i lui sa tina pelerina!

Rima pentru *pelerina*: Alina, masina, vina, tina

6. Scrieți o funcție care primește o listă variabilă de argumente (nenumite), valori întregi, ce reprezintă coordonatele unui vector și returnează norma euclidiană a acestuia.

7. Se citesc coordonatele în spațiu pentru *n* puncte din . Să se determine raza sferei centrată în origine pe care se afla cele mai multe puncte și coordonatele acestora.

8. Să se verifice dacă într-un text există cuvinte-anagrame și, în caz afirmativ, să se afișeze angramele, fără repetiție (nu se va face diferența între litere mici/mari), ca în următorul exemplu: pentru șirul *Rac, rac ingrat cu rating, eram intr-un car mare de argint cu arc.* (afirmație fără sens, a se lua în considerare numai cuvintele) se va afișa (nu neaparat în această ordine):

ingrat :

rating

argint

eram :

mare

rac :

car

arc

**Laborator 8**

1. Fie funcțiile următoare. Care este diferența?

| **def functie1(\*l, c = 0):**  **if len(l) == 0:**  **return 0**  **L = []**  **for x in l:**  **if str(x).find(str(c)) != -1:**  **L.append(x)**  **return L** | **def functie2(l, c = 0):**  **if len(l) == 0:**  **return 0**  **L = []**  **for x in l:**  **if str(x).find(str(c)) != -1:**  **L.append(x)**  **return L** |
| --- | --- |
| Ce se afișează în urma apelurilor următoare? Justificați, apoi verificați-vă rulând codurile. | |
| a) A = [103, 345, 1000, 456, 3]  print(functie1(\*A)) | a) A = [103, 345, 1000, 456, 3]  print(functie2(A)) |
| b) A = [103, 345, 1000, 456, 3]  print(functie1(A)) |  |
| c) A = [103, 345, 1000, 456, 3]  print("Lista returnata: ",functie1(A, 30, 530, 45)) | c) A = [103, 345, 1000, 456, 3]  print("Lista returnata: ",functie2(A, 30, 50, 45)) |
| d) A = [103, 345, 1000, 456]  print("Lista returnata: ",functie1(A, 30, 530, 45, c = 3)) | d) - |
| e) A = [103, 345, 1000, 456]  print("Lista returnata: ",functie1(\*A, 30, 530, 45, c = 3)) | e) - |
| f) A = [103, 345, 1000, 456]  print("Lista returnata: ",functie1(\*A, c = 3)) | f) A = [103, 345, 1000, 456, 3]  print("Lista returnata: ",functie2(A, c = 3)) |
| g) A = [103, 345, 1000, 456]  print("Lista returnata: ",functie1(\*A, 3)) | g) A = [103, 345, 1000, 456, 3]  print("Lista returnata: ",functie2(A, 3)) |
| h) print("Lista returnata: ",functie1(103, 345, 1000, 456, c = 3) | h) print("Lista returnata: ",functie2(103, 345, 1000, 456, c = 3) |

2. Sa se citeasca din fisierul *text.txt* un text literar, pe mai multe linii. Sa se afiseze pentru fiecare linie in parte numarul de propozitii. O propoziție se poate încheia cu unul sau mai multe dintre următoarele semne de punctuație: *?!.* În cazul în care pe o linie există mai multe cuvinte de lungime maximă, se va alege unul oarecare. Afișarea se va face în fișierul *mesaje.txt*, ca in exemplu.

*Exemplu de fișier de intrare (text.txt)*

*Acesta este examenul numarul 3 din aceasta sesiune. Avem subiecte usoare, cred ca luam note intre 8 si 10!... Ana a terminat deja?*

*Maine plec la munte. Iau o pauza de 2 zile.*

*Diana trece pe la mine deseara. Ii voi povesti cum a fost la examen. Stie ca am invatat mult. Foarte mult!*

*Exemplu de fișier de iesire (mesaje.txt)*

*Linia numarul 1 din fisier contine 3 propozitii, iar cel mai lung cuvant de pe aceasta linie este “examenul”.*

*Linia numarul 2 din fisier contine 2 propozitii, iar cel mai lung cuvant de pe aceasta linie este “Maine”.*

*Linia numarul 3 din fisier contine 4 propozitii, iar cel mai lung cuvant de pe aceasta linie este “invatat”.*

3. Din fisierul *date.txt* se citeste de pe prima linie un numar natural *n*. Se citesc apoi de pe următorele *n* linii, datele a *n* candidate la examenul de admitere la facultate, astfel:

*CNP, nume prenume, nota admitere*

Datele sunt separate prin virgule și unul sau mai multe spații

Sa se afiseze in fișierul *rezolvari.out* (ca în exemplu).

a) câți candidați sunt fete/băieți;

b) câți candidați sunt născuți înainte de anul 2001 (absolvenți ai promoțiilor anterioare);

c) candidații care au luat nota de trecere (peste 5.00) la examenul de admitere.

*Exemplu de fișier de intrare (date.txt)*

4

5020721100789 , Pop-Demetrescu Nicolae, 9.76

6011101780500, Marinescu Elena , 10.00

1870911800500, Toma Dan-Viorel, 4.65

5021023170485 , Popescu Marian, 3.40

*Exemplu de fișier de ieșire (rezolvari.out)*

1 fete, 3 baieti

1 candidati din promotiile anterioare

2 candidati au luat nota de trecere

5. Din fișierul *numere.txt* se citesc numere naturale, separate prin unul sau mai multe spații, pe unul sau mai multe linii. Să se afișeze în fișierul *sume.txt* suma cifrelor pentru fiecare număr din fișierul *numere.txt*. Se vor evita structurile repetitive (recomandate: map, filter).

**Laborator 9**

1. Să se afișeze ce zi a săptămânii este data curentă.

*Exemplu:* dacă rulăm codul în data de 09.12.2021 se va afișa *joi*.

2. Se citește de pe prima linie a fisierului *date.txt* numarul natural *n* apoi, pentru *n* persoane, pe cate o linie, numele și data nașterii, in format *zz.ll.aaaa*. Să se afișeze în fișierul *luni.txt* care sunt studenții născuți în fiecare lună a anului.

*Exemplu:*

**INPUT**

*date.txt*

4

Pop Ioan-Daniel, 13.02.2002

Toma Laura-Elena, 15.03.2000

Vergu Matei, 25.11.2003

Jercan Dan, 18.03.2003

**OUTPUT**

Febuarie: Pop Ioan-Daniel

Martie: Toma Laura-Elena, Jercan Dan

Noiembrie: Vergu Matei

3. Fara a folosi structuri repetitive, afisați o singură dată, în ordine lexicografică, vocalele folosite intr-un text citit din fisierul *text.txt*.

**INPUT**

*text.txt*

Ana are mere.

**OUTPUT**

A, a, e

4. Creati modulul *numere*. Fără a folosi structure repetitive, scrieți o funcție *suma\_cif\_k* ce primește un număr variabil de parametri poziționali (valori naturale) și returnează lista numerelor ce au suma cifrelor egală cu *k*, primit ca parametru. Adăugați funcția în modulul *numere*. Apelați funcția în alt fișier sursă (importând modului *numere*).

**Laborator 10**

1. In timpul saptamanii Scoala Altfel, intrarea in Muzeul National al Aviatiei Romane este permisa numai pentru grupuri organizate, cu programare. Tarifele sunt urmatoarele:

*10 lei: elevi si studenti*

*5 lei: pensionari*

*20 lei: adulti*

*copii (care nu sunt elevi): gratuit*

Daca numarul de persoane care platesc bilet este mai mare de 20, atunci se aplica o reducere de 10%. Sa se citeasca din fisierul de intrare date.txt cate persoane sunt intr-un grup, din fiecare categorie in parte, cu formatul dat in exemplu si sa se afiseze suma preturilor biletelor, calculata pentru intregul grup.

Conventii pentru citire datelor: E (elev), S (student), P (pensionar), A (adult), C (copil).

*Exemplu*

**INPUT**

date.txt

30 E, 3 A, 2 C, 2 S

**OUTPUT**

342 lei

2. Sa se citeasca de la tastatură o dată calendaristica în formatul *zz.ll.aaaa*, ce reprezintă data nasterii pentru o persoana. Sa se afiseze ziua din saptamana cand va fi urmatoarea aniversare.

*Exemplu*

Pentru 17.12.1980 se va afișa ‘Vineri’ (următoarea aniversare va fi vineri, 17.12.2021)

Pentru 12.03.1980 se va afișa ‘Sambata’ (următoarea aniversare va fi sâmbătă, 12.03.2022)

3. Într-o școală există obiceiul ca în fiecare sâmbătă sa se organizeze o petrecere pentru cei care care își aniversează ziua in săptămâna respectivă. Se citește din fisierul *date.txt* numarul natural *n* apoi, pentru *n* persoane, pe cate o linie, numele și data nașterii, in format *zz.ll.aaaa*. Să se afișeze un mesaj de forma:

*Dragii nostri ..(persoana sarbatorita 1)..., ... (persoana sarbatorita 2).., ..etc... / Draga ..(o singura persoana sarbatorita)...*

*Va / Te invitam la petrecerea organizata pentru voi / tine sambata, ...(data)...* .

*Cu drag,*

*Școala X*

*Exemplu:*

**INPUT**

*date.txt*

6

Tomescu Elena, 15.12.2000

Toma Diana, 16.12.1999

Popescu Dan, 07.12.2000

Georgescu Mara, 11.12.1998

Florea Andrei, 12.03.2000

**OUTPUT**

(pentru cod executat în perioada 13-17 dec. 2021)

*Dragii nostri Tomescu Elena, Toma Diana,*

*Va invitam la petrecerea organizata pentru voi sambata, 18.12.2021*.

*Cu drag,*

*Școala X*

*Dragii nostri Popescu Dan, Georgescu Mara,*

*Va invitam la petrecerea organizata pentru voi sambata, 10.12.2022*.

*Cu drag,*

*Școala X*

*Draga Florea Andrei,*

*Te invitam la petrecerea organizata pentru tine sambata, 12.03.2022*.

*Cu drag,*

*Școala X*

**Laborator 11**

1. Din fisierul *date.txt* se citeste de pe prima linie *n* număr natural, apoi, de pe următoarele *n* linii se citesc pentru *n* persoane numele si data nasterii. Pe baza acestor date, folosind metoda *substitute,* creati un mesaj personalizat de forma:

*Draga …*,

*Mai sunt .... zile pana la ziua ta, cand vei implini … ani. Am facut rezervare inca de pe acum in Bucegi, la cabana “Muntele Rosu”, in perioada …. - ……*

Perioada va incepe cu 3 zile inainte de ziua de nastere si se va incheia la alte 3 zile dupa aceasta.

2. Rezolvati urmatoarele cerințe folosind ori de câte ori este posibil expresii regulate.

Se citește un text. Să se afișeze:

a. cel mai scurt / lung cuvant;

b. cuvintele care conțin numai vocale;

c. propozițiile cu cele mai multe cuvinte;

d. primul cuvânt din fiecare propozitie.

3. Citiți de la tastatură și memorați CNP-ul, numele și salariul pentru n angajați. Să se afișeze angajații ordonați descrescător după salariu. Cei cu același salariu vor fi ordonați alfabetic.