

**str.translate(tabel\_traducere)**

- **tabel\_traducere** – de obicei obținut cu **str.maketrans(x[, y[, z]])** (sau dicționar cu coduri Ascii)

**str.maketrans(x[, y[, z]])**

- **x** - dicționar sau șirul cu caracterele de înlocuit
- **y** – șirul cu caracterele noi (de aceeași lungime cu x) => caracterul **x[i]** se va înlocui cu **y[i]**
- **z** – șir cu caracterele care vor fi șterse

## EXAMPLE

1. Se citește un text conținând separatorii uzuali( ,,:;) Sa se înlocuiască toți separatorii cu spațiu.
2. Se citește un cuvânt format cu litere mici. Să se înlocuiască fiecare vocală din cuvânt cu următoarea literă din alfabet.
3. Aceeași cerință ca la 2, dar în plus să se ștergă semnele: virgula, punct, două puncte.
4. Se citește o propoziție. Să se înlocuiască fiecare cifră < 5 care apare în text cu denumirea ei (1-unu, 2-doi, 3- trei, 4 -patru)

**re.split(tipar\_separator, sir, maxsplit)**

- **tipar\_separator** – expresie regulată care poate fi formată din caractere simple sau caractere speciale prin care putem crea tipare de căutare, precum :
  - \D - se potrivește cu orice caracter care nu este cifră
  - \d - se potrivește cu orice caracter care este cifră
  - [] – delimitează o mulțime de caractere
  - + - expresia se poate repeta, cel puțin o dată

<https://docs.python.org/3/library/re.html>

## EXAMPLE

1. Se dă o propoziție în care cuvintele sunt separate prin unul sau mai multe semne de punctuație uzuale.
  - a) Să se afișeze cuvintele din propoziție ordonate lexicografic.
  - b) Se citesc două cuvinte s și t. Să se afișeze propoziția obținută înlocuind toate aparițiile lui s cu t.
2. (laborator) Jurnalul electronic al Anei conține, în fiecare zi, câte o frază cu informații despre cheltuielile pe care ea le-a efectuat în ziua respectivă. Scrieți un program care să citească o frază de acest tip din jurnalul Anei și apoi să afișeze suma totală cheltuită de ea în ziua respectivă. De exemplu, pentru fraza “Astăzi am cumpărat pâine de 5 RON, pe lapte am dat 10 RON, iar de 15 RON am cumpărat niște cașcaval. De asemenea, mi-am cumpărat și niște papuci cu 50 RON!”, programul trebuie să afișeze suma totală de 80 RON. Fraza se consideră corectă, adică toate numerele care apar în ea sunt numere naturale reprezentând sume cheltuite de Ana în ziua respectivă!

## PROBLEME CU ȘIRURI ȘI LISTE. COMPREHENSIUNE

1. Se citește o propoziție cu cuvintele separate prin spații (unul sau mai multe). Să se creeze o listă cu cuvintele care încep cu o vocală (folosind și comprehensiune)
2. Se consideră o listă de liste (matrice) `ls` (de exemplu `[[4,7, 3], [3,1,20], [5,2,11]]`). Să se creeze o listă cu toate elementele din `ls` ordonate crescător (pentru exemplu `[1, 2, 3, 3, 4, 5, 7, 11, 20]`).
3. Cifrul lui Cezar
  - a) Să se creeze în memorie o listă cu literele mici ale alfabetului folosind comprehensiune și un cuvânt având ca litere toate literele mici ale alfabetului (există `string.ascii_lowercase`).
  - b) Se citește un text ce conține numai litere mici ale alfabetului englez și semnele de punctuație uzuale și un număr natural `k`. Să se afișeze textul cifrat cu cifrul lui Cezar, prin care fiecare **literă** dintr-un text dat este înlocuită cu litera aflată peste `k` poziții la dreapta în alfabet în mod circular (valoarea `k` reprezintă cheia secretă comună pe care trebuie să o cunoască atât expeditorul, cât și destinatarul mesajului criptat).
4. Păsărească - Se citește de la tastatură un text. Se cere să se “traducă” în limba păsărească textul dat astfel: după fiecare vocală se adaugă litera `p` și încă o dată acea vocală (după `a`, `e`, `i`, `o`, `u` se adaugă respectiv `pa`, `pe`, `pi`, `po`, `pu`). Exemplu: “Ana are mere.” devine “Apanapa aparepe meperepe.”.

## OPERATII LISTE

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

5. Se dau două liste `l1` și `l2` de lungime `n`. Să se înlocuiască elementele de pe poziții pare din `l1` cu cele de pe poziția corespunzătoare din `l2` folosind felieri (slice)
6. Se dă o listă de numere naturale. Să se șteargă din listă subsecvența delimitată de primele două zerouri din listă (inclusiv zerourile)
7. Se dă o listă de numere naturale. Să se șteargă din listă toate zerourile
8. Se dă o listă de numere naturale și un număr natural `k`. Să se elimine din listă subsecvența de lungime `k` de sumă minimă (dacă sunt mai multe se va elimina prima = cea mai din stânga) – fără a folosi liste suplimentare
9. 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.
10. 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)

## MATRICE

11. 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).
12. Se citesc `n`, `m` și o matrice cu `n` linii și `m` coloane (numerele sunt date câte unul pe linie). Să se ordoneze crescător elementele de pe prima coloană prin interschimbări de linii.