LISTE

TABLOURI BIDIMENSIONALE

- listă de liste

ex: M = {[1,2,3] , [4,5,6] , [7,8,9]}
M = {[1,2] , [4,8,9] , [9]}
M = {}
nrlim = int. (input ("Nr linii: "))
for i in range (nrlim)
    linie = [int (x) for x in input ("Linie"). split()]
    M.append (linie)

ex: m = 3 (nr linii)
m = 4 (nr coloane)
M = [[0] *m] *m   nu e corect
M = [[0,0,0,0],
       [0,0,0,0]
       [0,0,0,0]]
M. [0][2] = 7
var. corectă:
M = [[0 for i in range (m)] for i in range(m)]

- tuplu = listă imutabilei
ex: T = (1,2,3,4)
T [2] = 100   => eroare - nu se pot modifica valorile

- crearea unui tuplu:
    T = ()
    T = (0,)
    T = (03L, "Pop. Ion.", 95a)
    T = tuple (x+i for x in range (5))
    T = (1,1,3,4,5)

METODE PT TUPLURI

1. count (val)

2. index (val)

PARCURGEREA UNEI LISTE

- $L = \{x+1 \text{ for } x \text{ in range } (10)\} \Rightarrow L = \{1,2,3, \ldots 10\}$

```
for x in L:
    if x%2 == 0:
        x = 2*x
print(L) => L = {1,2, ... 10}
```
$x = $ "read-only" func. ca o copie

-
```
for k in range (len (L)):
    if L[k] %2 == 0:
        L[k] = 2* L[k]
```
$k \neq$ "read only"

-
```
for k, v in enumerate (L):
    print (k, v)
```

$k = 0/1/2/3 \ldots /9$ index
$v = 1/2/3/ \ldots /10$ valoare

ÎMPACHETAREA / DESPACHETAREA TUPLURILOR / LISTELOR

- Împachetare: $t = 1,2,3 \Rightarrow t = (1,2,3)$

- despachetare: $x,y,z = t \Rightarrow \begin{cases} x = 1 \\ y = 2 \\ z = 3 \end{cases}$

- $t = (131, \text{"pop"}, \text{"len"}, 9.50)$

$gr, *nume, med = t$

$\hookrightarrow 2\text{"pop"}, \text{"len"} \}$

- $M = [\{1,2,3\}, \{4,5,6\}, \{7, 8, 9\}]$

print (*M, sep = "\n")

af: {1,2,3}
     {4,5,6}
     {7,8,9}

sau

```
for linie in M:
    print (* linie )
```

af: 1 2 3
     4 5 6
     7 8 9

| TABELE DE DISPERSIE (HASH-TABLE) |

- Funcție de dispersie / hash function

$$s = "test"$$

$$s.\_\_hash\_\_() = 123456$$

$$hash(s) = 123456$$

$$C1: object_1 == object_2 \Rightarrow hash(object_1) == hash(object_2)$$

combinat

$$object_1 != object_2 \overset{?}{\Rightarrow} hash(object_1) == hash(object_2) \Rightarrow$$

$$\Rightarrow COLIZIUNE \ (e \ inutil)$$

$$C2: hash(object) = constant \ pt \ rulare$$

- $b = 8$ buckets

$$h(x) = x \% b = x \% 8$$ (im)

valori: $12, 27, 15, 20$

hashuri: $4, 3, 7, 4$ coliziune

```
t
0
1
2
3  → 27
4  → 12 → 20
5
6
7  → 15
```
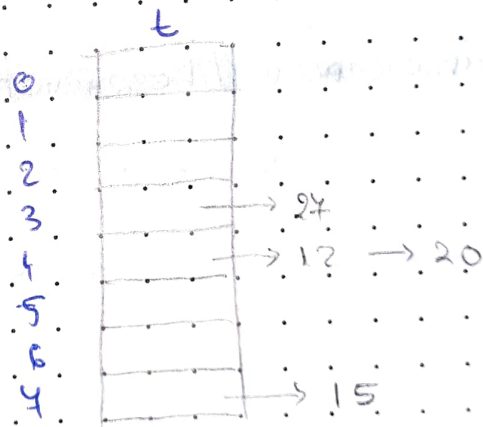
| MULTIMI / CLASA SET |

- multimi = colecție mutabilă de valori imutabile fără duplicate

1) $s = set()$

2) $s = \{1,2,1,2,2,2,3\} \Rightarrow s = \{1,2,3\}$

3) s

- Operatori: 1) in, not in

2) $<, >, <=, =<, ==, !=$

$\{3,1,2\} == \{1,2,3\}$

$\{1,3\} < \{3,2,1\}$ TRUE

$\{2,1,3\} > \{1,2\}$

3). (Funcții de mulțimi)

| → reuniune

& → intersecție

- → diferență

∧ → dif. simetrică!

$A \triangle B = (A \setminus B) \cup (B \setminus A)$

① len (mulțime)

② set (secvență)   ex: set ("teste") = { "t", "e", "s", ... , ... }

③ min (mult.) / max (mult.)

④ sum (mulțime)

① add (element)

② update (secvență)

③ remove (val.) → eroare KeyError (try ... except)

④ discard (val.)

⑤ clear ()

⑥ union (secvență) ; intersection (secvență)

difference (secv.) ; symmetric_difference (secv.)

ex: A = set ({1,...,1,2,,3,2,5})   =>   R = A & L   => eroare

L = [1,3,5,3,5,5]

R = A & set (L) abia după =>   R = A. intersection (L)