

## SEMINAR 3

### Colecții de date

1. Se citesc  $n - 1$  numere naturale distincte dintre primele  $n$  numere naturale nenule. Să se afișeze numărul lipsă. De exemplu, dacă se citesc numerele 2 1 5 4, atunci numărul lipsă este 3.

2. Să se afișeze elementele comune tuturor liniilor unui tablou bidimensional. De exemplu, dacă tabloul bidimensional este :

2 1 5 1 3

1 4 2 2

2 1 1 6 8

atunci trebuie afișate numerele 1 2, nu neapărat în această ordine.

3. Să se verifice dacă două șiruri de caractere formate doar din litere mici sunt anagrame sau nu. Două șiruri sunt anagrame dacă sunt formate din aceleași litere, dar așezate în altă ordine (sau, echivalent, unul dintre șiruri se poate obține din celălalt printr-o permutare a caracterelor sale). De exemplu, șirurile emerit și treime sunt anagrame, dar șirurile emerit și treimi nu sunt!

4. Scrieți un program care construiește o matrice pătratică  $M$  de dimensiune  $n$ , având forma următoare:

- pe ultima linie și ultima coloană toate elementele sunt egale cu 1;
- restul elementelor se calculează ca sumă a elementelor vecine de la est și sud.

**Exemplu:** pentru  $n = 4$  matricea  $M$  cerută este:

$$M = \begin{pmatrix} 10 & 10 & 4 & 1 \\ 10 & 6 & 3 & 1 \\ 4 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Seminar 3

⑤ (Seminar 2)

s = input("text: ")

eworte = s.split()

candidates = None

prev = None

suma = 0

for ew in eworte:

try:

aux = float(ew)

if c == None:

c = aux

else:

prev = aux

suma += c \* prev

c = prev = None

except:

pass

print(suma)

② m = int(input("max limit: "))

l = []

for i in range(m):

limit = input("limit: ") "14 10 3 5"

limit = limit.split() ["14", "10", "3", "5"]

limit = [int(y) for y in limit] [14, 10, 3, 5]

~~l = []~~

l.append(limit)

l = [14, 10, 3, 5]

~~l = set(l)~~

← l = set(l[0:])

for x in range(1, m):

L = l.intersection(l[x:])

L = L &amp; set(l[x:])

print(L)

④

```

m = int(input("m = "))
n = 5
for i in range(m):
    H =
    H = [1] * m for x in range(m)
    for i in range(m-2, -1, -1):
        for j in range(m-2, -1, -1):
            H[i][j] = H[i+1][j] + H[i][j+1]
    for lim in range(M):
        print(*lim)

```

①

```

L = [int(x) for x in range(m)] input("Mn: ").split()
for x in
m = len len(L) + 1
for i in range(1, m+1):
    if i not in L:  $\rightarrow O(m)$ 
        print(i)
        break

```

$\rightarrow O(m^2)$

---

$V_2$

$L = \{ \dots \} \rightarrow L$  multiple  
 $i$  not in  $L \Rightarrow O(1)$   $\Rightarrow O(m)$

---

$V_3$

$L = \{ \dots \}$   
 $m = \text{len}(L) + 1$   
 $L.sort()$

```

for x
st = 0
dr = m - 1
while st <= dr:
    mij = (st + dr) // 2
    if L[mij] == i:
        break
    elif L[mij] < i:
        st = mij + 1
    else:
        dr = mij - 1

```

```

else:
    print(i)
    break

```