ESERCIZIARIO PYTHON SULLE LISTE MONO E BIDIMENSIONALI

SEMPLICI:

1. Somma degli elementi
   Scrivi un programma che chieda all'utente di inserire 10 numeri interi in una lista. Al termine, stampa:
   - la lista completa;
   - la somma di tutti gli elementi;
   - la media dei valori inseriti.

2. Ricerca del massimo e del minimo
   Crea una lista contenente 8 numeri inseriti dall'utente. Senza utilizzare le funzioni max() e min(), trova e stampa:
   - il valore massimo;
   - il valore minimo;
   - la differenza tra il massimo e il minimo.

3. Numeri pari e dispari
   Chiedi all'utente di inserire 12 numeri interi in una lista. Crea due nuove liste:
   - una contenente solo i numeri pari;
   - una contenente solo i numeri dispari.
   - Infine, stampa le tre liste e il numero di elementi presenti in ciascuna.

4. Eliminazione dei duplicati
   L'utente inserisce 15 numeri interi in una lista (sono ammessi valori ripetuti). Crea una nuova lista contenente gli stessi elementi ma senza duplicati, mantenendo l'ordine della prima comparsa. Al termine stampa:
   - la lista originale;
   - la lista senza duplicati;
   - il numero di elementi eliminati.

MEDIA DIFFICOLTÀ

5. List comprehension:
   Produci utilizzando una list comprehension una lista l di 30 numeri reali casuali da 0 a 1. Produci poi una matrice mat di 5 liste (sempre con le list comprehension) aventi all'interno i numeri n contenuti in l tale che in mat[i] siano contenuti numeri 0.22*i <= n < (1+i)*0.22 (ovviamente i numeri in mat[4] saranno comunque <=1)
   - Stampare l
   - Stampare quanti numeri siano contenuti in ogni riga e gli elementi stessi di ogni riga in ordine crescente
   - stampare poi, la media totale dei numeri, la media di ogni lista e la media delle medie di ogni lista.

6. Somma delle righe e delle colonne
   Scrivi un programma che chieda all'utente di inserire una matrice di 4 righe e 5 colonne di numeri interi.
   Al termine il programma deve:
   - stampare la matrice in modo ordinato;
   - calcolare e stampare la somma di ogni riga;
   - calcolare e stampare la somma di ogni colonna;
   - indicare quale riga ha la somma maggiore.

7. Ricerca in una matrice
   Genera una matrice di 6×6 contenente numeri casuali compresi tra 1 e 100.
   Chiedi all'utente un numero da cercare e:
   - verifica se è presente nella matrice;
   - se presente, stampa tutte le coordinate (riga, colonna) in cui compare;
   - indica quante volte compare;
   - se non è presente, stampa un messaggio appropriato.

8. Matrice simmetrica
   Chiedi all'utente di inserire una matrice quadrata 4×4.
   Il programma deve verificare se la matrice è simmetrica, cioè se l'elemento in posizione (i, j) è uguale all'elemento in posizione (j, i) per tutte le coppie di indici.
   Infine stampa:
   - la matrice;
   - un messaggio che indichi se è simmetrica oppure no.

9. Statistiche della matrice
   Genera una matrice 5×5 di numeri casuali compresi tra 0 e 50.
   Calcola e stampa:
   - il valore massimo e la sua posizione;
   - il valore minimo e la sua posizione;
   - la media di tutti gli elementi;
   - il numero di valori pari e dispari;
   - quanti elementi sono maggiori della media.

10. Rotazione della matrice
    - Chiedi all'utente di inserire una matrice 3×3.
    - Successivamente crea una nuova matrice ottenuta ruotando quella originale di 90° in senso orario, senza modificare la matrice iniziale.
    - Infine stampa entrambe le matrici.

11. Analisi di una tabella di voti
    Una matrice di 8×5 rappresenta i voti di 8 studenti in 5 materie.
    Il programma deve:
    - calcolare la media di ogni studente;
    - calcolare la media di ogni materia;
    - trovare lo studente con la media più alta;
    - trovare la materia con la media più bassa;
    - contare quanti voti insufficienti (minori di 6) sono presenti nell'intera matrice.

DIFFICOLTÀ ELEVATA

12. Classifica e statistiche di una lista
    L'utente inserisce una lista di 30 numeri interi.
    Il programma deve:
    - ordinare la lista in ordine crescente senza usare sort() o sorted();
    - stampare il terzo valore più grande e il terzo valore più piccolo;
    - calcolare la mediana della lista;
    - contare quante volte compare ciascun numero distinto;
    - stampare il numero che compare più frequentemente.

13. Prodotto tra matrici
    Chiedi all'utente di inserire due matrici:
    - una matrice A di dimensione 3×2;
    - una matrice B di dimensione 2×4.

    Il programma deve:
    - verificare che il prodotto tra le due matrici sia possibile;
    - costruire la matrice prodotto A×B senza utilizzare librerie esterne;
    - stampare la matrice risultante;
    - individuare il valore massimo e il valore minimo della matrice prodotto e le loro posizioni.

14. Percorso di somma massima
    Genera una matrice 6×6 di numeri casuali compresi tra 1 e 20.
    Partendo dall'elemento in alto a sinistra, è possibile muoversi solo:
    - verso destra;
    - verso il basso.

    Il programma deve:
    - trovare un percorso che porti all'elemento in basso a destra ottenendo la somma più grande possibile;
    - stampare il percorso seguito;
    - stampare la somma totale degli elementi attraversati;
    - indicare il numero di passi effettuati.

15. Analisi completa di una matrice quadrata
    Chiedi all'utente di inserire una matrice quadrata 5×5.
    Il programma deve:
    - calcolare la somma degli elementi della diagonale principale;
    - calcolare la somma degli elementi della diagonale secondaria;
    - verificare se la matrice è triangolare superiore;
    - verificare se la matrice è triangolare inferiore;
    - verificare se la matrice è simmetrica;
    - determinare quale riga possiede la somma maggiore;
    - determinare quale colonna possiede la somma minore;
    - stampare un riepilogo finale con tutti i risultati ottenuti.

SOLUZIONI

1. Somma degli elementi

```
lista = []

for i in range(10):
    numero = int(input("Inserisci un numero: "))
    lista.append(numero)

somma = 0

for numero in lista:
    somma += numero

media = somma / len(lista)

print("Lista:", lista)
print("Somma:", somma)
print("Media:", media)
```

2. Ricerca del massimo e del minimo

```
lista = []

for i in range(8):
    lista.append(int(input("Inserisci un numero: ")))

massimo = lista[0]
minimo = lista[0]

for numero in lista:
    if numero > massimo:
        massimo = numero

    if numero < minimo:
        minimo = numero

print("Massimo:", massimo)
print("Minimo:", minimo)
print("Differenza:", massimo - minimo)
```

3. Numeri pari e dispari

```
lista = []
pari = []
dispari = []

for i in range(12):
    numero = int(input("Inserisci un numero: "))
    lista.append(numero)
```

```
for numero in lista:

    if numero % 2 == 0:
        pari.append(numero)
    else:
        dispari.append(numero)

print("Lista:", lista)
print("Pari:", pari)
print("Dispari:", dispari)

print("Elementi lista:", len(lista))
print("Elementi pari:", len(pari))
print("Elementi dispari:", len(dispari))
```

4. Eliminazione dei duplicati

```
lista = []

for i in range(15):
    lista.append(int(input("Inserisci un numero: ")))

senzaDuplicati = []

for numero in lista:

    if numero not in senzaDuplicati:
        senzaDuplicati.append(numero)

eliminati = len(lista) - len(senzaDuplicati)

print("Lista originale:", lista)
print("Lista senza duplicati:", senzaDuplicati)
print("Elementi eliminati:", eliminati)
```

5. List comprehension

```
import random

l = [random.random() for i in range(30)]

mat = [[n for n in l if 0.22*i <= n < (i+1)*0.22] for i in range(5)]

print("Lista:")
print(l)
print()

for i in range(5):
```

```
    mat[i].sort()

    print("Riga", i)

    print("Numero elementi:", len(mat[i]))

    print(mat[i])

    print()

mediaTot = sum(l)/len(l)

print("Media totale:", mediaTot)

medie = []

for riga in mat:

    if len(riga) > 0:

        media = sum(riga)/len(riga)

    else:

        media = 0

    medie.append(media)

for i in range(5):

    print("Media lista", i, "=", medie[i])

print("Media delle medie:", sum(medie)/len(medie))
```

6. Somma delle righe e delle colonne

```
mat = []

for i in range(4):
    riga = []

    for j in range(5):
        numero = int(input(f"Elemento [{i}][{j}]: "))
        riga.append(numero)

    mat.append(riga)

print("Matrice:")
```

```
for riga in mat:
    print(riga)

sommeRighe = []

for riga in mat:
    sommeRighe.append(sum(riga))

print()

for i in range(4):
    print("Somma riga", i, "=", sommeRighe[i])

print()

for j in range(5):

    somma = 0

    for i in range(4):
        somma += mat[i][j]

    print("Somma colonna", j, "=", somma)

indice = 0

for i in range(1,4):

    if sommeRighe[i] > sommeRighe[indice]:
        indice = i

print()
print("Riga con somma maggiore: “, indice)
```

7. Ricerca in una matrice

```
import random

mat = []

for i in range(6):

    riga = []

    for j in range(6):
        riga.append(random.randint(1,100))
```

```
    mat.append(riga)

print("Matrice:")

for riga in mat:
    print(riga)

numero = int(input("Numero da cercare: "))

trovato = False
contatore = 0

for i in range(6):

    for j in range(6):

        if mat[i][j] == numero:

            trovato = True
            contatore += 1

            print("Trovato in posizione:", i, j)

if trovato:
    print("Compare", contatore, "volte")
else:
    print("Numero non presente")
```

8. Matrice simmetrica

```
mat = []

for i in range(4):

    riga = []

    for j in range(4):
        riga.append(int(input(f"[{i}][{j}]: ")))

    mat.append(riga)

simmetrica = True

for i in range(4):

    for j in range(4):

        if mat[i][j] != mat[j][i]:
```

```
            simmetrica = False

    print()

    for riga in mat:
        print(riga)

    if simmetrica:
        print("La matrice è simmetrica")
    else:
        print("La matrice NON è simmetrica")
```

9. Statistiche della matrice

```
    import random

    mat = []

    for i in range(5):

        riga = []

        for j in range(5):
            riga.append(random.randint(0,50))

        mat.append(riga)

    for riga in mat:
        print(riga)

    massimo = mat[0][0]
    minimo = mat[0][0]

    posMax = (0,0)
    posMin = (0,0)

    somma = 0

    pari = 0
    dispari = 0

    for i in range(5):

        for j in range(5):

            valore = mat[i][j]

            somma += valore
```

```
            if valore > massimo:
                massimo = valore
                posMax = (i,j)

            if valore < minimo:
                minimo = valore
                posMin = (i,j)

            if valore % 2 == 0:
                pari += 1
            else:
                dispari += 1

    media = somma/25

    cont = 0

    for riga in mat:

        for valore in riga:

            if valore > media:
                cont += 1

    print()

    print("Massimo:", massimo, posMax)
    print("Minimo:", minimo, posMin)
    print("Media:", media)
    print("Pari:", pari)
    print("Dispari:", dispari)
    print("Maggiori della media:", cont)
```

10. Rotazione della matrice

```
    mat = []

    for i in range(3):

        riga = []

        for j in range(3):
            riga.append(int(input(f"[{i}][{j}]: ")))

        mat.append(riga)

    rot = [[0,0,0],[0,0,0],[0,0,0]]
```

```
for i in range(3):

    for j in range(3):

        rot[j][2-i] = mat[i][j]

print()

print("Matrice originale")

for riga in mat:
    print(riga)

print()

print("Matrice ruotata")

for riga in rot:
    print(riga)
```

11. Analisi di una tabella di voti

```
import random

voti = []

for i in range(8):
    riga = []
    for j in range(5):
        riga.append(random.randint(2,10))
    voti.append(riga)

print("Tabella voti")

for riga in voti:
    print(riga)

medieStudenti = []

for riga in voti:
    medieStudenti.append(sum(riga)/5)

medieMaterie = []

for j in range(5):
    somma = 0
    for i in range(8):
```

```
            somma += voti[i][j]
        medieMaterie.append(somma/8)

    studente = 0

    for i in range(1,8):
        if medieStudenti[i] > medieStudenti[studente]:
            studente = i

    materia = 0

    for j in range(1,5):
        if medieMaterie[j] < medieMaterie[materia]:
            materia = j

    insufficienti = 0

    for riga in voti:
        for voto in riga:
            if voto < 6:
                insufficienti += 1

    print("Media studenti:", medieStudenti)
    print("Media materie:", medieMaterie)
    print("Studente migliore:", studente)
    print("Materia peggiore:", materia)
    print("Voti insufficienti:", insufficienti)
```

12. Classifica e statistiche di una lista

```
    lista = []

    for i in range(30):
        lista.append(int(input("Numero: ")))

    for i in range(len(lista)-1):
        for j in range(len(lista)-1-i):
            if lista[j] > lista[j+1]:
                lista[j], lista[j+1] = lista[j+1], lista[j]

    print("Lista ordinata:", lista)

    print("Terzo più piccolo:", lista[2])
    print("Terzo più grande:", lista[-3])

    mediana = (lista[14] + lista[15]) / 2
```

```
print("Mediana:", mediana)

frequenze = {}

for numero in lista:
    if numero in frequenze:
        frequenze[numero] += 1
    else:
        frequenze[numero] = 1

print(frequenze)

massima = 0
numeroMax = None

for numero in frequenze:
    if frequenze[numero] > massima:
        massima = frequenze[numero]
        numeroMax = numero

print("Numero più frequente:", numeroMax)
```

13. Prodotto fra matrici

```
A = []
B = []

print("Matrice A")

for i in range(3):
    riga = []
    for j in range(2):
        riga.append(int(input()))
    A.append(riga)

print("Matrice B")

for i in range(2):
    riga = []
    for j in range(4):
        riga.append(int(input()))
    B.append(riga)

C = []

for i in range(3):
```

```
        riga = []

        for j in range(4):

            somma = 0

            for k in range(2):
                somma += A[i][k] * B[k][j]

            riga.append(somma)

        C.append(riga)

    print("Prodotto:")

    for riga in C:
        print(riga)

    massimo = C[0][0]
    minimo = C[0][0]

    posMax = (0,0)
    posMin = (0,0)

    for i in range(3):
        for j in range(4):

            if C[i][j] > massimo:
                massimo = C[i][j]
                posMax = (i,j)

            if C[i][j] < minimo:
                minimo = C[i][j]
                posMin = (i,j)

    print("Massimo:", massimo, posMax)
    print("Minimo:", minimo, posMin)
```

14. Percorso di una somma massima

```
    import random

    mat = []

    for i in range(6):
        riga = []
        for j in range(6):
```

```
        riga.append(random.randint(1,20))
    mat.append(riga)

for riga in mat:
    print(riga)

dp = [[0]*6 for _ in range(6)]

dp[0][0] = mat[0][0]

for i in range(6):
    for j in range(6):

        if i == 0 and j == 0:
            continue

        sopra = -1
        sinistra = -1

        if i > 0:
            sopra = dp[i-1][j]

        if j > 0:
            sinistra = dp[i][j-1]

        dp[i][j] = max(sopra, sinistra) + mat[i][j]

print("Somma massima:", dp[5][5])
print("Numero passi:", 10)
```

15. Analisi completa di una matrice quadrata

```
mat = []

for i in range(5):

    riga = []

    for j in range(5):
        riga.append(int(input()))

    mat.append(riga)

diagP = 0
diagS = 0

for i in range(5):
```

```
    diagP += mat[i][i]
    diagS += mat[i][4-i]

sup = True
inf = True
sim = True

for i in range(5):

    for j in range(5):

        if i > j and mat[i][j] != 0:
            sup = False

        if i < j and mat[i][j] != 0:
            inf = False

        if mat[i][j] != mat[j][i]:
            sim = False

sommeRighe = [sum(riga) for riga in mat]

rigaMax = sommeRighe.index(max(sommeRighe))

sommeColonne = []

for j in range(5):

    somma = 0

    for i in range(5):
        somma += mat[i][j]

    sommeColonne.append(somma)

colMin = sommeColonne.index(min(sommeColonne))

print("Somma diagonale principale:", diagP)
print("Somma diagonale secondaria:", diagS)
print("Triangolare superiore:", sup)
print("Triangolare inferiore:", inf)
print("Simmetrica:", sim)
print("Riga con somma maggiore:", rigaMax)
print("Colonna con somma minore:", colMin)
```