

Programmazione

Si scriva una funzione "alternati" che riceve in input una lista l e restituisce True se la lista contiene valori alternati pari e dispari, cioè non ci sono due elementi consecutivi entrambi pari o entrambi dispari, e False altrimenti. Si invochi inoltre la funzione sulle liste [2, 5, 8, 3, 2], [1, 2, 3, 4, 0] e [1, 1, 2, 3, 4, 5] e se ne discutano i risultati.

Soluzione:

```
def alternati(l):
    precedente_pari = l[0] % 2 == 0
    ok = True
    i = 1
    while i < len(l) and ok:
        if l[i] % 2 == 0 and precedente_pari or l[i] % 2 == 1 \
            and not precedente_pari:
            ok = False
        precedente_pari = l[i] % 2 == 0
        i += 1
    return ok

print(alternati([2, 5, 8, 3, 2]))
print(alternati([1, 2, 3, 4, 0]))
print(alternati([1, 1, 2, 3, 4, 5]))
```

L'output sarà True, False e False perché solo la prima lista contiene valori alternati.

Programmazione

Si scriva una funzione "controlla_lista" che riceve in input una lista di interi e

- se non tutti gli elementi sono uguali, stampa il massimo, il minimo e gli indici di tali elementi;
- se tutti gli elementi sono uguali, ne stampa il valore.

La funzione non deve utilizzare le funzioni "max" e "min" di Python e inoltre può utilizzare un solo ciclo for.

Soluzione:

```
def controlla_lista(lista):
    massimo = lista[0]
    indiceMassimo = 0
    minimo = lista[0]
    indiceMinimo = 0
    for i in range(1, len(lista)):
        if lista[i] > massimo:
            massimo = lista[i]
            indiceMassimo = i
        elif lista[i] < minimo:
            minimo = lista[i]
            indiceMinimo = i
    if massimo == minimo:
        print('Tutti gli elementi sono uguali a', lista[0])
    else:
        print('Il massimo è', massimo, 'e si trova in posizione', indiceMassimo)
        print('Il minimo è', minimo, 'e si trova in posizione', indiceMinimo)
```

Programmazione

Si scriva una funzione "genera_lista" che riceve in input due liste di interi V1 e V2 di uguale lunghezza e restituisce una lista di interi V3 di lunghezza pari a quella di V1 e V2, il cui generico elemento i è così ottenuto:

- Se i è pari, V3[i] è la somma degli elementi di V1 con indice maggiore di i.
- Se i è dispari, V3[i] è il prodotto degli elementi di V2 con indice maggiore di i.
- L'ultimo elemento di V3 è posto uguale a 0.

Ad esempio, se V1 = [2, 5, 11, 3, 7] e V2 = [1, 2, 4, 3, 2], la lista restituita è V3 = [26, 24, 10, 2, 0].

Soluzione:

```
def genera_lista(V1 , V2):
    lunghezza = len(V1)
    V3 = []
    for i in range(lunghezza-1):
        if i%2 == 0:
            somma = 0
            for j in range(i+1, lunghezza):
                somma += V1[j]
            V3.append(somma)
        else:
            prodotto = 1
            for j in range(i+1, lunghezza):
                prodotto *= V2[j]
            V3.append(prodotto)
    V3.append(0)
    return V3
```

Comprensione

Si consideri la seguente funzione:

```
def verifica(matrice):
    for i in range(len(matrice)):
        for j in range(len(matrice[0])):
            if i == j and matrice[i][j] != 1 or i != j and matrice[i][j] != 0:
                return False
    return True
```

Si descriva sinteticamente il comportamento della funzione.

Soluzione:

La funzione verifica se la matrice data è la matrice identità.

Comprensione

Si considerino le seguenti funzioni:

```
def verifica_1(matrice_quadrata):  
    n = len(matrice_quadrata)  
    for i in range(n):  
        for j in range(n):  
            if matrice_quadrata[i][j] != matrice_quadrata[j][i]:  
                return False  
    return True  
  
def verifica_2(matrice_quadrata):  
    n = len(matrice_quadrata)  
    for i in range(1,n):  
        for j in range(0,i):  
            if matrice_quadrata[i][j] != matrice_quadrata[j][i]:  
                return False  
    return True
```

Si descriva sinteticamente il comportamento delle funzioni.

Soluzione:

Entrambe le funzioni verificano se la matrice data è simmetrica, ma la seconda risparmia confronti inutili escludendo i casi in cui gli indici i e j corrispondono al triangolo superiore o alla diagonale principale.

Programmazione

Si scriva una funzione "prodotto_scalare_righe" che riceve in input una matrice M di interi e due interi r1 ed r2 e restituisce il prodotto scalare delle righe r1 ed r2 di A.

Soluzione:

```
def prodotto_scalare_righe(M, r1, r2):  
    ps = 0  
    for j in range(len(M[0])):  
        ps += M[r1][j] * M[r2][j]  
    return ps
```

Programmazione

Si scriva una funzione "colonna_inversa" che riceve in input una matrice di interi e restituisce True se e solo se la prima colonna è uguale all'ultima letta in ordine inverso.

Soluzione:

```
def colonna_inversa(matrice):  
    r = len(matrice)  
    c = len(matrice[0])  
    for i in range(r):  
        if matrice[i][0] != matrice[-1-i][c-1]:  
            return False  
    return True
```

Programmazione

Si scriva una funzione "triang_sup" che riceve in input una matrice quadrata e restituisce True se e solo se la matrice è triangolare superiore.

Soluzione:

```
def triang_sup(mat):  
    for i in range(1, len(mat)):  
        for j in range(0, i):  
            if mat[i][j] != 0:  
                return False  
    return True
```

Programmazione

Si scriva una funzione "triang_inf" che riceve in input una matrice quadrata e restituisce True se e solo se la matrice è triangolare inferiore.

Soluzione:

```
def triang_inf(mat):
    for i in range(0, len(mat)):
        for j in range(i+1, len(mat)):
            if mat[i][j] != 0:
                return False
    return True
```

Programmazione

Si scriva una funzione "costruisci_lista" che riceve in input una lista di interi V1 e restituisce una lista di interi V2 di pari lunghezza, il cui generico elemento i è così ottenuto:

- Se la media degli elementi di V1 con indice maggiore o uguale a i è maggiore o uguale a V1[i], allora V2[i] è uguale a tale media.
- Altrimenti, V2[i] è uguale alla differenza tra la somma degli elementi alla sinistra di V1[i] e la somma degli elementi alla destra di V1[i] (ovviamente se non c'è nessun elemento alla destra o alla sinistra tale somma vale zero).

Soluzione:

```
def costruisci_lista(V1):
    V2 = []
    for i in range(len(V1)):
        somma = 0
        for j in range(i, len(V1)):
            somma += V1[j]
        media = somma // (len(V1) - i)
        if media >= V1[i]:
            V2.append(media)
        else:
            sommal = 0
            for k in range(i):
                sommal += V1[k]
            differenza = sommal - (somma - V1[i])
            V2.append(differenza)
    return V2
```