Elevamento a potenza senza l'operatore ** e la funzione pow

```
b = int(input('Inserisci la base: '))
e = int(input("Inserisci l'esponente: "))
risultato = 1
for i in range(1, e+1):
    risultato *= b
print(b, 'elevato a', e, 'è uguale a', risultato)
```

Comprensione

Si consideri il seguente brano di codice:

```
lista = [3, 6, 8, 16, 8, 16]

ok = True
i = 0
while i<len(lista) // 2 and ok:
    if lista[i] > lista[len(lista)-1-i]:
        ok = False
    i += 1
if ok:
    print('OK')
else:
    print('NO')
```

Si descriva sinteticamente la funzione svolta dal brano. Si discuta inoltre l'effetto che si otterrebbe utilizzando nell'if la condizione alternativa "lista[i] > lista[-1-i]".

Soluzione:
Il brano verifica se gli elementi nella prima metà della lista sono minori dei corrispondenti elementi speculari nella seconda metà della lista.
Il brano stamperà pertanto 'OK' perché lista[0] < lista[5], lista[1] < lista[4], e così via.
La condizione alternativa produce lo stesso risultato, in quanto l'indice -1-i è sempre equivalente all'indice len(lista)-1-i.

Comprensione

Si consideri il seguente brano di codice:

```
v = [1, 2, 3, 4, 6, 2, 1]
d = 7

s = 0
while s < d and v[s] == v[d-1]:
    s += 1
    d -= 1
if s < d-1:
    print('NO');
else:
    print('SI');
```

Si descriva sinteticamente la funzione svolta dal brano.

Soluzione:
Il brano di codice verifica se la sotto-lista di lunghezza d, ottenuta come prefisso di v, è *palindroma*, cioè il primo elemento è uguale all'ultimo (v[0]==v[d-1]), il secondo al penultimo (v[1]==v[d-2]) e così via.
Il brano stamperà pertanto 'NO' perché la condizione non è rispettata: il valore 3 è diverso dal valore 6.

Gestione liste

```
lunghezza = int(input('Inserisci la lunghezza della lista: '))
lista = []
for i in range(lunghezza):
    prompt = "Inserisci l'elemento di indice " + str(i) + ': '
    lista.append(int(input(prompt)))

print('--Conteggio zeri e pari--')
zeri = 0
pari = 0
for n in lista:
    if n == 0:
        zeri += 1
    if n % 2 == 0:
        pari += 1
print('La lista contiene',zeri,'zeri e',pari,'elementi pari')

print('--Verifica tutti uguali ad un dato valore--')
valore = int(input('Inserisci il valore: '))
tutti_uguali = True
i = 0
while i < lunghezza and tutti_uguali:
    if lista[i] != valore:
        tutti_uguali = False
    i += 1
if tutti_uguali:
    print('Tutti gli elementi sono uguali a',valore)
else:
    print('Non tutti gli elementi sono uguali a',valore)

print('--Verifica lista ordinata--')
i = 0
ordinata = True
while i < lunghezza - 1 and ordinata:
    if lista[i] > lista[i+1]:
        ordinata = False
    i += 1
if ordinata:
    print('La lista è ordinata')
else:
    print('La lista non è ordinata')

print('--Calcolo lista somma--')
seconda_lista = []
for i in range(lunghezza):
    prompt = "Inserisci l'elemento di indice " + str(i) + ' della seconda lista:
'
    seconda_lista.append(int(input(prompt)))
lista_somma = []
for i in range(lunghezza):
    lista_somma.append(lista[i] + seconda_lista[i])
print('La lista somma è',lista_somma)
```

Comprensione

Si consideri il seguente brano di codice:

```
lista =[3,2,4,7,2,3,5]

for i in range(len(lista)):
    if i % 2 == 0 and lista[i] % 2 == 0:
        print(lista[i])
    if i % 2 != 0 and lista[i] % 2 != 0:
        print(lista[i])
```

Si descriva sinteticamente la funzione svolta dal brano.

Soluzione:
Il brano stampa gli elementi della lista che rispettano una delle seguenti condizioni:

- essere di valore pari e in posizione pari, oppure
- essere di valore dispari e in posizione dispari.

Il brano stamperà pertanto 4, 7, 2 e 3 in quanto: lista[0]==3 (valore dispari 3 in posizione pari 0) non rispetta né la prima né la seconda condizione; lista[1]==2 (valore pari 2 in posizione dispari 1) non rispetta né la prima né la seconda condizione; lista[2]==4 (valore pari 4 in posizione pari 2) rispetta la prima condizione; lista[3]==7 (valore dispari 7 in posizione dispari 3) rispetta la seconda condizione, e così via.

Comprensione

Si consideri il seguente brano di codice:

```
x = [3,2,5,4]
y = [8,20,6,9]
d = 4
n = [False] * d

for i in range(d):
    if y[i] % x[-1-i] == 0:
        n[i] = True
print(n)
```

Si descriva sinteticamente la funzione svolta dal brano.

Soluzione:
Il brano di codice verifica che ogni elemento della lista y sia divisibile per l'elemento della lista x che si trova nella posizione simmetrica. L'output sarà [True,True,True,True], in quanto 8%4==0, 20%5==0, ecc.

Comprensione

Si consideri il seguente brano di codice:

```
def funzione_esercizio(x, l):
    i = 0
    while i < len(l):
        if l[i] % x != 0:
            l[i] = 0
        else:
            l[i] /= x
        i += 1

lista = [9,6,2,7,15,10]
valore = 3
funzione_ esercizio(valore,lista)
for n in lista:
    if n != 0:
        print(n)
```

Si descriva sinteticamente il comportamento della funzione "funzione_esercizio(x, l)" e si indichino gli output prodotti dal brano.

Soluzione:
La funzione verifica quali elementi della lista l sono divisibili per x. In particolare, se l'i-esimo elemento della lista è divisibile per x, allora esso viene posto pari al quoziente della divisione; altrimenti, viene posto pari a 0. Il brano stampa gli elementi diversi da zero della lista ottenuta – l'output sarà quindi 3 (9/3), 2 (6/3) e 5 (15/3).

Comprensione

Si consideri il seguente brano di codice:

```
def f(a,b):
    s = 0
    for i in range(1,a+1):
        p = 1
        for j in range(1,b+1):
            p *= i
        s += p
    return s

print(f(2,4))
print(f(4,3))
```

Si descriva sinteticamente il comportamento della funzione "f(a,b)" e si indichino gli output prodotti dal brano.

Soluzione:
La funzione calcola $\sum_{i=1}^{a} i^b$. Gli output prodotti dal brano sono quindi i valori 17 e 100.

Comprensione

Si consideri il seguente brano di codice:

```
def prodotto(p,i):
    risultato = 1
    for j in range(len(p)):
        if not(p[j]):
            risultato *= i[j]
    return risultato

a = [True, False, False, True, True, False, True]
b = [-5, 128, 56, 15, -6, 12, -3]

print(prodotto(a,b))
```

Si descriva sinteticamente il comportamento della funzione "prodotto(p,i)" e si indichino gli output prodotti dal brano.

Soluzione:
La funzione considera gli indici j tali che p[j] è False e calcola il prodotto degli elementi aventi gli stessi indici nella lista i. In altre parole, la lista p indica alla funzione quali elementi *non* considerare nel calcolo del prodotto. L'output prodotto è quindi il valore 86016, cioè, il prodotto tra 128, 56 e 12.

Programmazione

Si scriva una funzione "rollup" che riceve in input una lista l di lunghezza n (con n pari) e restituisce una lista di lunghezza n/2 il cui i-esimo elemento è pari a l[2*i]+l[2*i+1]. Si invochi inoltre la funzione sulla lista [1, 3, 6, 1, 6, 3, 8, 9] e se ne discutano i risultati.

Soluzione:

```
def rollup(l):
    risultato = []
    for i in range(0,len(l)//2):
        risultato.append(l[2*i]+l[2*i+1])
    return risultato

print(rollup([1,3,6,1,6,3,8,9]))
```

L'output sarà [4,7,9,17] perché 1+3=4, 6+1=7, 6+3=9 e 8+9=17.