

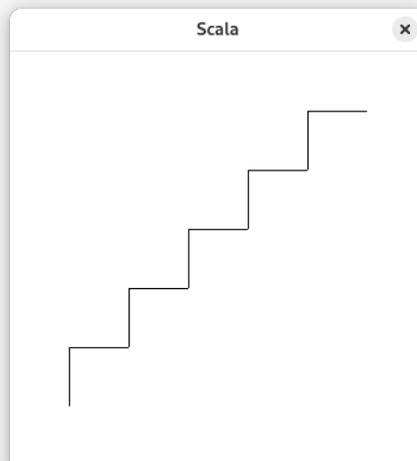
Programmazione e Algoritmi 1 - Prova pratica

A.A. 2024/25 — Compito di esempio

prof. Gianluca Amato

Esercizio 1 (7 punti)

Scrivere un programma che prende in input un numero intero n e, utilizzando la libreria `ezgraphics`, disegna una scala composta da n gradini. Ad esempio, questo è il possibile output del programma per $n = 5$:



Il programma deve rispettare questi requisiti:

- La dimensione (larghezza ed altezza) dei gradini della scala è di 50 pixel.
- La finestra grafica dovrà avere dimensioni variabili a seconda del numero di gradini della scala, in modo da contenerla tutta.
- La scala deve essere centrata nella finestra grafica.

Esercizio 2 (7 punti)

Scrivere una funzione `sottostringa(s1, s2)` che prende come parametri due stringhe e restituisce un booleano. La funzione deve restituire `True` se `s1` è una sottostringa di `s2`, ottenuta cioè scegliendo, in ordine e senza ripetizioni, alcuni caratteri di `s2`. Ad esempio, se la prima stringa è `"ciao"` e la seconda è `"ci23aa4o"`, la funzione restituirà `True`. Se invece la seconda stringa è `"o2xcyia"`, la funzione restituirà `False`.

Esercizio 3 (3 punti)

Scrivere alcuni test con il framework `pytest` per la funzione `sottostringa` dell'esercizio precedente. Si considerino i seguenti casi:

- gli esempi dell'esercizio 5;
- almeno due esempi per cui `s1` è vuota;
- almeno due esempi per cui `s2` è vuota;
- almeno un esempio in cui `s1` ed `s2` sono uguali.

Programmazione e Algoritmi 1 - Prova scritta

A.A. 2024/25 — Compito di esempio

prof. Gianluca Amato

Esercizio 4 (5 punti)

Scrivere una funzione che prende come parametri due stringhe `s1` ed `s2` e restituisce una nuova stringa ottenuta alternando un carattere da `s1` e uno da `s2`. Se una delle due stringhe è più corta dell'altra, la parte in eccesso della stringa più lunga viene aggiunta al risultato. Ad esempio, se `s1 = "ciao"` e `s2 = "12345678"`, la funzione restituirà la stringa `"c1i2a3o45678"`.

Esercizio 5 (5 punti)

Si consideri la seguente funzione in Python che, presa una lista in input, restituisce la stessa lista in ordine inverso.

```
1 def inverti(l):
2     res = []
3     for v in l:
4         res = [v] + res
5     return res
```

Tenendo conto che l'operazione di concatenazione tra liste ha complessità pari a $O(l_1 + l_2)$, dove l_1 ed l_2 sono le lunghezze delle due liste da concatenare, determinare la complessità computazionale della funzione di cui sopra.

Esercizio 6 (7 punti)

Eseguire passo-passo il seguente codice Python, utilizzando l'apposito modulo:

```
1 def dup(l):
2     res = []
3     for v in l:
4         res.append(v)
5         res.append(v)
6     return res
7
8 l1 = [ [0], [1,2] ]
9 l2 = dup(l1)
10 l1[1][1] = "a"
11 print(l2)
```

Descrivere lo stato della memoria (heap e stack) alla fine dell'esecuzione del programma.