

Problema 1. Scrivere una funzione ricorsiva che, data una stringa *s*, restituisce *vero* se la stringa contiene la stessa quantità di cifre numeriche e caratteri alfabetici, *falso* altrimenti. Nota: ricordare i metodi *isdigit()* e *isalpha()* del tipo di dato *str* in Python.

Problema 2. Data la seguente definizione della funzione `foo()`, quale valore viene assegnato alla variabile *x* nel comando `x = foo(10,10)` ? Motivare la risposta.

```
def foo(x, y+10) :  
    # @param x: int  
    # @param y: int  
    # @return int  
    y = y - 10  
    return x+y
```

Problema 3. Date le rappresentazioni in base B dei seguenti tre numeri:

x = 1101 (B = 4)

y = 11001 (B = 3)

z = 1001011 (B = 2)

determinare la relazione d'ordine tra i tre (ovvero, quale dei tre è, rispettivamente, il più piccolo, il più grande e quello intermedio).

Problema 4. Date le seguenti definizioni di funzioni e variabili, determinare: (i) gli spazi dei nomi (*namespace*) locali delle singole funzioni e quello globale; (ii) il risultato visualizzato sullo schermo se le funzioni vengono attivate a partire dalla funzione `start()`. Per entrambi i punti (i) e (ii) considerare i seguenti due casi: (a) il comando (in grigio nel testo) `"global a"` è effettivamente presente nella funzione `Q()`; (b) il comando `"global a"` non è presente nella funzione `Q()`.

```
a = 'A'  
  
def Q():  
    global a  
    print a  
    a = 'b'  
  
def start():  
    a = 'a'  
    R(a)  
    print a  
  
def R(x) :  
    a = 'B'  
    Q()  
    x = a  
    print x
```

Problema 5. Si definisca una classe *PoligonoRegolare*. Ogni oggetto della classe ha come attributi il numero di lati e la lunghezza di un lato, e come operazioni due metodi che restituiscono, rispettivamente, la lunghezza del lato e il perimetro del poligono. Si definisca inoltre una classe *Quadrato*, derivata dalla classe *PoligonoRegolare*, che ha un metodo aggiuntivo che restituisce l'area del quadrato. Dare la definizione completa delle due classi in Python, che include la definizione dei

rispettivi costruttori e dei metodi indicati. Fornire, inoltre, alcuni frammenti di codice Python che esemplificano l'uso delle classi (creazione di oggetti delle due classi, uso dei metodi).

Problema 6. Scrivere una funzione (più eventuali funzioni ausiliarie) che, data un'immagine *A*, verifica se *A* contiene almeno una riga i cui pixel hanno tutti lo stesso colore.

Nota: la valutazione della soluzione proposta include anche l'adeguatezza della sua strutturazione.

Problema 7. Si definisca una classe che ha come attributo una tupla di *N* interi aventi valore iniziale 0, con *N* definito al momento della creazione di un oggetto della classe. Inoltre, la classe dispone di un metodo per visualizzare tutta la tupla, e di un metodo (avente parametri *k* e *i*) che modifica l'attributo tupla rimpiazzandolo con una nuova tupla dove il valore *k* è stato sommato all'elemento in posizione *i* della vecchia tupla. Definire poi una classe derivata dalla prima, dove il valore iniziale da assegnare a tutti gli elementi della tupla (uguale per tutti) può essere specificato al momento della creazione di un oggetto della classe, tramite un opportuno parametro del costruttore. Fornire, inoltre, alcuni frammenti di codice Python che esemplificano l'uso delle classi (creazione di oggetti delle due classi, uso dei metodi).

Problema 8. Si definisca come *luminosità* di un colore la media aritmetica del valore delle tre componenti di una sua rappresentazione RGB. Si definisca come *luminosità media* di un insieme di pixel la media aritmetica della luminosità dei colori dei pixel dell'insieme. Scrivere una funzione (più eventuali funzioni ausiliarie) che, data una immagine *A*, verifica se tutte le righe di *A* hanno la stessa luminosità media.

Problema 9. Date le seguenti definizioni di funzioni e variabili, determinare: (i) gli spazi dei nomi (*namespace*) locali delle singole funzioni e quello globale; (ii) il risultato visualizzato sullo schermo se le funzioni vengono attivate a partire dalla funzione `start()`. Per entrambi i punti (i) e (ii) considerare i seguenti due casi: (a) il comando (in grigio nel testo) `"global a"` è effettivamente presente nella funzione `K()`; (b) il comando `"global a"` non è presente nella funzione `K()`.

```
a = 0.3

def K():
    global a
    a = 4

def F(x):
    K()
    x = 8
    print a+x

def start():
    a = 0.5
    F(a)
    print a
```

Problema 10. Date le seguenti rappresentazioni di numeri reali in virgola mobile (coppia **mantissa** ed **esponente**), con base *B*=2, *k*=3 cifre per la mantissa, e *h*=2 cifre per l'esponente, determinare il

valore numerico corrispondente (espresso nella usuale notazione decimale). Nelle rappresentazioni mantissa ed esponente sono rappresentati in modulo e segno; lo spazio eventualmente occupato dai due segni non è incluso in k e h ; la virgola è posizionata a sinistra della cifra di peso maggiore della mantissa (rappresentazione normalizzata):

a) (110 -10) **b)** (-001 -10) **c)** (010 10)

(suggerimento: per semplificare i calcoli, si può esprimere il risultato in forma di numero frazionario)

Problema 11. Si considerino i seguenti comandi Python. Quale è il valore della variabile x al termine della loro esecuzione? Motivare la risposta.

```
dict = {[0,0]:1, [0,1]:2, [0,3]:3}
x = dict[[0,1]]
```

Problema 12. Scrivere una funzione **ricorsiva** che, data una stringa s , verifica se la stringa è formata da due metà uguali tra loro (ovvero verifica se $s == s1+s2$, con $s1==s2$). Se la lunghezza della stringa è dispari, il carattere centrale va escluso dal confronto.

Nota: nel definire la funzione, l'operatore Python di uguaglianza $==$ può essere applicato solo a stringhe di lunghezza ≤ 1 .