



## Fondamenti di Disegno e Programmazione

Codice esame : 054391, Modulo di Fondamenti di Informatica

**Appello del 16 Gennaio 2020**

Nome:

---

Cognome:

---

Codice Persona:

Firma:

---

- Il punteggio massimo è di 32 punti, un punteggio di 18 punti è dunque da ritenersi pienamente sufficiente.
- Scrivere solo sui fogli distribuiti, utilizzando il retro delle pagine in caso di necessità e cancellando le parti di brutta con un tratto di penna. Non separare questi fogli.
- È consentito utilizzare libri, eserciziari, appunti o calcolatrici. Tali strumenti, con gli oggetti personali, devono essere riposti sul banco.
- Non è consentito uscire dall'aula durante la prova o lasciare l'aula conservando il tema della prova in corso.
- Il possesso di telefoni cellulari, calcolatori o altri strumenti con capacità di comunicazione con l'esterno (e.g. tablet) – anche se spenti – durante la prova è equiparato a un tentativo di copia.
- Tempo a disposizione per l'esame: 2 ore.

### Punteggi

Esercizio	1	2	3.1	3.2	3.3	3.4	3.5	Totale
Punteggio								
Punteggio Massimo	2	3	5	5	7	8	2	32

## 1 Aritmetica binaria - punti 2

Si codifichi il numero  $-15$  in complemento a 2 su 5 bit e lo si sommi al numero  $(00010)_{\text{comp-a-2}}$ .

*Soluzione: si può ottenere  $-15$  codificato in complemento a 2 calcolando dapprima la codifica di 15 in binario naturale, ovvero  $(01111)_2$ , calcolando il complemento bit-a-bit, ovvero 10000 e sommando 1, ottenendo  $(10001)_{\text{comp-a-2}}$ . Sommando  $(00010)_{\text{comp-a-2}}$  e  $(10001)_{\text{comp-a-2}}$  si ottiene  $(10011)_{\text{comp-a-2}}$ , ovvero  $(-13)_{10}$*

## 2 Analisi di codice - punti 3

Cosa stampa il seguente programma Python quando viene eseguito, sostituendo a W,X,Y,Z le ultime quattro cifre del proprio codice persona nella riga marcata da # <-- QUI?

```
l=[W,X,Y,Z] # <-- QUI
```

```
def somma(a,b):  
    return a+b
```

```
def incremento(a):  
    a=a+1  
    return
```

```
print(somma(l[0]+[2]))  
print(incremento(l[1]))
```

*Soluzione: il programma stampa la somma del primo e del terzo numero presenti nella lista l, come risultato del calcolo della funzione **somma**, dopodichè stampa **None**, in quanto la funzione **incremento** non ritorna alcun valore.*

### 3 Sintesi di codice

Si consideri un archivio degli alberi presenti all'interno dell'area metropolitana milanese. Ogni albero è rappresentato con una tupla di tre elementi: una stringa contenente il nome, un intero contenente l'altezza in centimetri, un intero contenente l'età.

#### 3.1 Rappresentazione dei dati – Punti 5

Quali sono i tipi di dato o i contenitori Python adatti a rappresentare i seguenti concetti:

- Un catalogo contenente i nomi di tutte le specie arboree
- Una classificazione delle specie arboree in “ad alto fusto” e “cespugliose”
- La serie temporale costituita dal numero di alberi piantati, per ogni mese, negli ultimi 10 anni.

Soluzione:

- Un insieme è una buona scelta: il catalogo delle specie è naturalmente privo di duplicati
- Un dizionario avente come chiave il nome della specie arborea e come valore le stringhe “ad alto fusto” e “cespugliose”
- Una lista, o una tupla di numeri, di  $10 \times 12 = 120$  elementi

### 3.2 Implementazione – 1 – Punti 5

Si implementi la funzione `decano` che riceve un archivio degli alberi come parametri e restituisce l'albero più vecchio.

### 3.3 Implementazione – 2 – Punti 7

Si implementi la funzione `trova_gemelli` che riceve un archivio di alberi e restituisce l'*insieme* di alberi che hanno un almeno gemello, ovvero l'insieme di alberi di cui ne compare almeno uno identico in nome, altezza ed età nell'archivio.

### 3.4 Implementazione – 3 – Punti 8

Si implementi la funzione **riordina**. La funzione **riordina** riceve come parametro un percorso di un file testuale, contenente un archivio di alberi. Il file contiene i dati relativi ad un albero (nome, altezza, età) per rigo, separati da virgola. La funzione carica da file gli alberi in un archivio, lo ordina per nome dell' albero e riscrive l' archivio su file, utilizzando lo stesso formato testuale. Allo scopo di implementare la funzione **riordina** è consentito utilizzare qualunque funzione built-in di Python.

Esempio di contenuto di file:

```
faggio,250,15  
noce,400,35  
pioppo,600,10
```

### 3.5 Implementazione – 3 – Punti 2

Si implementi una funzione *ricorsiva*, chiamata **bilanciata**, che riceve come parametro una lista di interi lunga  $l$  e restituisce **True** se l'intero in posizione  $i$ -esima è sempre minore di quello in posizione  $(l - i)$ -esima, per ogni  $0 \leq i \leq \lfloor \frac{l}{2} \rfloor$ .