# Liceo Scientifico Statale G.D.Cassini – 3G

## Introduzione a Python

## Potenziamento di Informatica

Alex Carrega – aless and ro. carrega@unige.it $12^{th}$  Aprile 2023

### Importante

• Questo documento è scaricabile tramite il seguente link:

https://alexcarrega.netsons.org/link/cassini-potenziamento-informatica-3g-20230412-soluzione



- Per ogni domanda creare un file X.py dove X è il numero della domanda con un numero di 0 come prefisso per avere esattamente 3 cifre.
- Creare un file **zip** con i vari file **python** del punto precedente e rinominarlo *cognome-nome*. **zip** dove *cognome* e *nome* sono rispettivamente il **nome** e il **cognome** dello studente.
- Fare l'upload del file **zip** tramite il seguente link:

https://alexcarrega.netsons.org/link/cassini-potenziamento-informatica-3g-20230412-consegna



### Note

• La valutazione sarà fatta sulla base dei 93 punti totali secondo la seguente proporzione:

$$v = \frac{p}{93} \times 10$$

devo p e v sono, rispettivamente, i **punti** ottenuti in questa prova e la **valutazione** finale.

### Esempi

- Ipotizziamo che lo studente Paolo Rossi debba rispondere a 25 domande.
- I file **python** saranno: 001.py, 002.py, ..., 025.py.
- Il file zip contenente tali file python sarà chiamato: rossi-paolo.zip.

## 3 Punti 1. 😑 Max tra Due Numeri

Scrivi un programma che chieda due numeri all'utente tramite la funzione input e mostri il più grande tra i due utilizzando la funzione print.

### Importante

• Per quanto Python disponga di una funzione max(), siete invitati ad utilizzare le istruzioni istruzioni if, elif ed pythonelse per la scrittura dell'algoritmo.

```
Solution
   a = input("Inserisci il primo numero: ")
1
2
   b = input("Inserisci il secondo numero: ")
3
4
   if a == b:
5
       print("I numeri sono identici")
6
   elif a > b:
7
       print(f"Il numero più grande tra i due è {a}")
8
   else:
9
       print(f"Il numero più grande tra i due è {b}")
```

# 9 Punti 2. Max tra Tre Numeri

Scrivi un programma che chieda tre numeri a, b, c all'utente e mostri il più grande tra loro.

### Importante

• Per quanto Python disponga di una funzione max(), siete invitati ad utilizzare le istruzioni istruzioni if, elif ed else per la scrittura dell'algoritmo.

```
Solution
    a = input("Inserisci il primo numero: ")
    b = input("Inserisci il secondo numero: ")
 3
    c = input("Inserisci il terzo numero: ")
 4
 5
    if a \ge b and a \ge c:
 6
        print(a)
 7
    elif b \ge a and b \ge c:
 8
        print(b)
 9
    elif c >= a and c >= b:
10
        print(c)
```

## 3 Punti 3. 😑 Il Maggiore tra Tutti!

Scrivi un programma che chieda all'utente una **lista** di numeri e fornisca in output il maggiore tra tutti.

### ${f Importante}$

• Per quanto Python disponga di una funzione max(), siete invitati ad utilizzare le istruzioni istruzioni if, elif ed else per la scrittura dell'algoritmo.

```
Solution

lista_numeri = [42, 9, 23, 11, 17, 56, 3]
numero_maggiore = lista_numeri[0]
for numero in lista_numeri[1:]:
    if numero > numero_maggiore:
        numero_maggiore = numero
print("Il numero maggiore tra tutti è:", numero_maggiore)
```

## 3 Punti 4. Sei una Vocale?

Scrivi un programma che chieda all'utente una **stringa** composta da un solo carattere e dica se si tratta di una vocale oppure no.

```
Solution

1  carattere = input("Inserisci un carattere: ")
2  vocali = "aeiou"
3  if carattere in vocali:
4    print(f"Il carattere {carattere} è una vocale")
5  else:
6    print(f"Il carattere {carattere} non è una vocale")
```

### 3 Punti 5.

### 5. Somma Inarrestabile

Scrivi un semplice programma che, data una lista di numeri, sommi tra loro tutti gli elementi.

### Suggerimenti

• Anche se esiste la funzione sum() per risolvere l'esercizio potresti usare il ciclo for.

```
Solution

lista = [3, 6, 100, 23, 42]

risultato = 0

for numero in lista:
    risultato += numero

print(f"Il risultato della somma è {risultato}")
```

### 3 Punti 6. Somma Inarrestabile

Scrivi un programma "moltiplicatore" che, data una lista di numeri, moltiplichi tra loro tutti gli elementi.

```
Solution

lista = [7, 66, 100, 457, 472]

risultato = 1

for numero in lista:
    if numero != 0:
        risultato *= numero

print("Il risultato della moltiplicazione tra tutti gli "
    f"elementi della lista è {risultato}")
```

# 3 Punti 7. Solamente per Soci

Scrivi un programma che a partire da un elemento e una lista di elementi dica in output se l'elemento passato sia presente o meno nella lista.

#### Suggerimenti

• Qualora l'elemento sia presente nella lista, il programma dovrà comunicarci l'indice dell'elemento tramite il metodo index.

```
Solution
    lista = ['Marco', 'Luigi', 'Paolo', 'Giuseppe', 'Maria']
 2
    el = input("Inserisci un nome da cercare: ")
 3
    trovato = False
 4
    for nome in lista:
 5
        if nome == el:
 6
            trovato = True
 7
            break
 8
    if trovato:
 9
        print(f"{el} è presente nella lista all'indice {lista.index(el)}")
10
        print(f"{el} non è presente nella lista.")
11
```

## 3 Punti 8. Generatore di Istogrammi

Scrivi una semplice **funzione** che, data una lista di numeri, fornisca in output un **istogramma** basato su questi numeri, usando asterischi per disegnarlo.

### Esempi

• Data la lista [3, 7, 9, 5], la funzione dovrà produrre questa sequenza:

```
***

*****

*****
```

```
Solution

def istogramma(lista):
    for numero in lista:
        print("*" * numero)
```

# 3 Punti 9. Scriviamo la nostra versione di len()

Scrivi una funzione che restituisca la lunghezza di una stringa o lista passata come parametro.

#### Suggerimenti

• In sostanza, seppur presente, provate a scrivere la nostra versione della funzione len!

```
Solution

def my_len(lst_or_str):
    length = 0
    for unit in lst_or_str:
        length += 1
    return length
```

## 2 Punti 10. 😑 A Ciascuno il Suo

Scrivi una funzione che data in ingresso una lista A contenente n parole, restituisca in output una lista B di interi che rappresentano la lunghezza delle parole contenute in A.

```
Suggerimenti

• Questo esercizio può essere risolto anche usando una list comprehension.

1 quadrati = [n**2 for n in range(10)]

2

3 print(quadrati)

4 [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
Solution
1
   def char_counter(lista_a):
2
       lista_b = []
3
       for parola in lista_a:
4
           lista_b.append(len(parola))
5
       return lista_b
6
7
8
   def char_counter_pro(lista_a):
9
       return [len(parola) for parola in lista_a]
```

## 3 Punti 11. 😑 Il Frequenzimetro

Scrivi una funzione che, data una stringa come parametro, restituisca un **dizionario** rappresentante la "frequenza di comparsa" di ciascun carattere componente la stringa.

```
Esempi
Data una stringa "ababcc", otterremo in risultato {"a": 2, "b": 2, "c": 2}.
```

```
Solution
   def char_freq(str):
1
2
       mappa = \{\}
3
       for carattere in str:
4
           if carattere in mappa:
5
               mappa[carattere] += 1
6
7
                mappa[carattere] = 1
8
       return mappa
```

## 4 Punti 12. 😑 L'Americana

Scrivi una funzione che, dato in ingresso un valore espresso in metri, mandi in print l'equivalente in **miglia terrestri**, **iarde**, **piedi e pollici**. Come risolverai questo esercizio?

Misura	Nome inglese	Equivalente SI				
millesimo di pollice, mil	thou	$0.0254\mathrm{mm}$				
linea	line	$0.635\mathrm{mm}$				
pollice	inch	$25.4\mathrm{mm}$				
mano	hand	$101.6\mathrm{mm}$				
spanna	span	$228.6\mathrm{mm}$				
piede	$\operatorname{foot}$	$304.8\mathrm{mm}$				
gomito (cubito)	$\operatorname{cubit}$	$457.2\mathrm{mm}$				
iarda	yard	3  ft = 914.4  mm				
braccio	$\operatorname{fathom}$	2  yd = 1.8288  m				
barra, pertica	rod, pole, perch	$5.0292\mathrm{m}$				
catena	chain	$20.1168{ m m}$				
furlong	furlong	$201.168{ m m}$				
miglio terrestre	statute mile	1760  yd = 1609.344  m				

```
Solution
1
    def americana(metri):
 2
        conversions = dict()
        conversions["miglia"] = metri / 1609.344
 3
 4
        conversions["piedi"] = metri * 3.280840
 5
        conversions["pollici"] = metri * 39.37008
 6
        conversions["iarde"] = metri * 1.093613
 7
 8
        print(f"{metri} metri corrispondono a:")
 9
        for key, value in conversions.items():
10
            print(f"{key}: {value}")
```

## **5 Punti** 13. 😑 Il Signore del Tempo

Scrivi una semplice funzione che converta un dato numero di giorni, ore e minuti, passati dall'utente tramite funzione input, in secondi.

```
Solution
1
   def calcola_secondi():
2
       print("Questa funzione converte un dato numero"
3
             "di Giorni, Ore e Minuti in Secondi.")
4
       da_giorni = int(input("Inserisci il numero di giorni: ")) * 24 * 3600
5
       da_ore = int(input("Inserisci il numero di ore: ")) * 3600
       da_minuti = int(input("Inserisci il numero di minuti: ")) * 60
6
7
       totale = da_giorni + da_ore + da_minuti
8
       print(totale)
```

## 12 Punti 14. 📮 Il Geometra

Scrivi una funzione che, a scelta dell'utente, calcoli l'area di:

- un cerchio
- un quadrato
- un rettangolo
- un triangolo

#### Suggerimenti

• Sentitevi liberi di estendere le potenzialità della funzione quanto meglio credete!

```
Solution
 1
    def geometra():
 2
        print("""
 3
        Benvenuti alla funzione Geometra!
 4
        In fase di selezione, a ciascun possibile calcolo
 5
        corrisponde un valore numerico:
 6
        - Area Quadrato: 1
 7
        - Area Rettangolo: 2
 8
        - Area Triangolo: 3
 9
        - Area Cerchio: 4
10
11
        print('Dunque. Di quale figura geometrica desideri calcolare 1\'area?')
12
13
        scelta = int(input(">>> "))
        if scelta == 1:
14
            print("Hai scelto: Area Quadrato")
15
16
            lato = float(input('Inserisci il valore del lato del quadrato '))
            print(f"L'Area del Quadrato, avente lato {lato} è: {lato * lato}")
17
```

```
18
        elif scelta == 2:
19
            print("Hai scelto: Area Rettangolo")
20
            base = float(input('Inserisci il valore della base '))
21
            altezza = float(input('Inserisci il valore dell'altezza '))
22
            print(f"L'Area del Rettangolo, avente base {base} e "
23
                  f"altezza {altezza} è: {base * altezza}")
24
        elif scelta == 3:
25
            print("Hai scelto: Area Triangolo")
26
            base = float(input('Inserisci il valore della base '))
27
            altezza = float(input('Inserisci il valore dell'altezza '))
28
            print(f"L'Area del Triangolo, avente base {base} e "
29
                  f"altezza {altezza} è: {(base * altezza) / 2}")
30
        elif scelta == 4:
31
            print("Hai scelto: Area Cerchio")
32
            r = float(input('Inserisci il valore del raggio '))
33
            print(f"L'Area del Cerchio, avente raggio {r} è: {(r * r) * 3.14}")
34
        else:
35
            print('Nessun calcolo disponibile per la scelta effettuata!')
```

## 4 Punti 15. 😑 Funzione Genera MAC

Un indirizzo MAC (Media Access Control address) è un indirizzo univoco associato dal produttore, a un chipset per comunicazioni wireless (es. WiFi o Bluetooth), composto da 6 coppie di cifre esadecimali separate da due punti. Scrivi una funzione genera\_mac() che generi degli indirizzi MAC pseudo casuali.

#### Esempi

- 02:FF:A5:F2:55:12
- 00:02:C9:35:32:31
- 66:10:CB:CC:E4:80

#### Suggerimenti

• Utilizzare il modulo random.

```
Solution
1
   import random
2
3
4
   def genera_mac():
5
       char set = "ABCDEF0123456789"
6
       mac_addr = ""
7
       due_punti = 0
8
9
       for _ in range(6):
```

## 2 Punti 16. 😑 Info di Sistema

Scrivi una funzione che fornisca in output il nome del Sistema Operativo utilizzato con eventuali relative informazioni sulla release corrente.

### Suggerimenti

• Per risolvere questo esercizio potreste dover utilizzare una libreria (platform)!

```
Solution

import platform

def sys_info():
    print(f"Il Sistema attualmente in uso è: {platform.system()}")
    print(f"Info Release: {platform.release()}")
```

# 2 Punti 17. 😑 Trova ASCII

Scrivi una funzione che, dato un carattere in ingresso, restituisca in output il **codice ASCII** associato al carattere passato.

#### Suggerimenti

• Anche in questo caso, usare una libreria (**ord**) potrebbe facilitare la risoluzione dell'esercizio!

```
Solution

def trova_ascii():
    carattere = input("Inserisci il carattere che ti interessa convertire: ")
    valore = ord(carattere)
    output = f"Il valore ASCII associato a '{carattere}' è {valore}"
    return output
```

## 9 Punti 18. 😑 Il Numero Perfetto

Un numero perfetto è un **numero naturale** uguale alla somma dei suoi divisori positivi, escluso sé stesso. Scrivi una funzione che verifichi se un numero è perfetto oppure no.

```
Solution
 1
    def perfetto(n):
 2
        somma_divisori = 0
 3
 4
        # Itera sui numeri da 1 a n/2 (escluso n)
 5
        for i in range(1, n//2 + 1):
 6
            # Se i è un divisore di n, si aggiunge alla somma
 7
            if n % i == 0:
 8
                somma_divisori += i
 9
10
        # Se la somma dei divisori è uguale a n, allora n è un numero perfetto
11
        if somma_divisori == n:
12
            return True
13
        else:
14
            return False
15
16
17
    n = int(input("Inserisci un numero intero positivo: "))
18
    if perfetto(n):
19
        print(f"Il numero {n} è un numero perfetto.")
20
21
        print(f"Il numero {n} non è un numero perfetto.")
```

## **2 Punti** 19. 😑 Lista di Colori

Scrivi una funzione che aggiunga ad una lista 10 colori inseriti dall'utente. Il programma deve poi chiedere all'utente di inserire una lettera e mostrare in output solo i colori nella lista che iniziano con quella lettera.

#### Suggerimenti

• Potresti usare la funzione range e il metodo startswith().

```
Solution
   def colori():
1
2
       colors = []
3
       for _ in range(10):
4
           color = input("Inserisci un colore: ")
5
           colors.append(color)
6
7
       # Chiede di inserire una lettera
8
       letter = input("Inserisci una lettera: ")
```

```
preturn [color for color in colors if color.startswith(letter)]
11
12
13 colori()
```

## 3 Punti 20. Print senza andare a capo

Scrivi una funzione che prenda una serie di input dall'utente utilizzando un ciclo while e li stampi con la funzione print senza andare a capo. Il ciclo while si deve interrompere quando l'utente preme INVIO senza scrivere nulla.

```
Solution

def print_senza_andare_a_capo():
    input_utente = input("Inserisci una stringa: ")
    while input_utente != "":
        print(input_utente, end=" ")
        input_utente = input("Inserisci una stringa: ")

print_senza_andare_a_capo()
```

# 3 Punti 21. 😐 La Segreteria

Scrivi una funzione che accetti una lista di **dizionari** rappresentante una scuola. Ogni dizionario rappresenta uno studente e contiene nome, cognome, classe e voti. La funzione deve stampare un elenco di tutti gli studenti e calcolare la media dei voti di ciascuno.

```
Solution
 1
    def segreteria(studenti):
 2
        for studente in studenti:
 3
            media_voti = sum(studente["voti"]) / len(studente["voti"])
 4
            studente["media"] = media_voti
 5
        return studenti
 6
 7
 8
    studenti = [
        {"nome": "Mario", "cognome": "Rossi", "classe": "3A", "voti": [6, 7, 8]},
 9
        {"nome": "Luigi", "cognome": "Bianchi", "classe": "3A", "voti": [7, 8, 9]},
10
11
        {"nome": "Anna", "cognome": "Verdi", "classe": "3B", "voti": [8, 9, 10]},
12
    ]
13
14
   media_studenti = segreteria(studenti)
15
    print(f"{media_studenti}/n")
```

## **2 Punti** 22. Gestione Login

Scrivi un programma che crei un file CSV per memorizzare in un dizionario i dati degli utenti registrati su un sito web. I dati richiesti per ogni utente sono: username, password, email e data di registrazione. Il programma deve permettere di salvare le informazioni nel file, leggere i dati e stamparli a schermo.

```
Solution
 1
    import csv
 2
 3
 4
    def crea_file_csv(dizionario, nome_file):
 5
        with open(nome_file, 'w', newline='') as file_csv:
 6
            writer = csv.writer(file_csv)
 7
            writer.writerow(['username', 'password', 'email',
 8
                              'data registrazione'])
 9
            for utente in dizionario.values():
10
                writer.writerow([utente['username'], utente['password'],
11
                                  utente['email'], utente['data_registrazione']])
12
13
14
    def leggi_file_csv(nome_file):
15
        with open(nome_file, 'r') as file_csv:
16
            reader = csv.reader(file_csv)
17
            for row in reader:
18
                print(row)
19
20
21
    # Dizionario degli utenti
22
    utenti = {
23
        1: {'username': 'piero', 'password': 'p4ssw0rd',
24
             'email': 'mario@gmail.com', 'data_registrazione': '2023-01-01'},
25
        2: {'username': 'lisa', 'password': 's3cr3t',
26
            'email': 'luigi@yahoo.com', 'data_registrazione': '2023-01-02'},
27
        3: {'username': 'rita', 'password': 'p3ach',
28
             'email': 'princess@castle.com', 'data_registrazione': '2023-01-03'}
29
    }
30
31
    # Crea il file CSV
32
    crea_file_csv(utenti, 'utenti.csv')
33
34
    # Legge il file CSV e stampa i dati a schermo
35
    leggi_file_csv('utenti.csv')
```

# **2 Punti** 23. 😑 Testi di canzoni

Scrivi una funzione che permetta di inserire una canzone e salvarla in un **file di testo**. Il programma deve chiedere all'utente di inserire il titolo e il testo della canzone, e poi salvare quest'ultimo in un file intitolato titolo-canzone.txt.

### Suggerimenti

• Dovrai utilizzare l'istruzione with.

```
Solution
 1
    def salva_testo_canzone(titolo, testo):
 2
        nome_file = f"{titolo}.txt"
 3
        with open(nome_file, 'w') as file_testo:
 4
            file_testo.write(testo)
 5
        print(f"Testo della canzone '{titolo}' salvato in '{nome_file}'.")
 6
 7
 8
    titolo_canzone = input("Inserisci il titolo della canzone: ")
 9
    testo_canzone = input("Inserisci il testo della canzone: ")
10
11
    salva_testo_canzone(titolo_canzone, testo_canzone)
```

## 2 Punti 24. 😑 Il Sistema Solare

Scrivi una funzione che crei una **tupla** contenente i nomi dei pianeti del sistema solare, la loro tipologia (gassoso o roccioso) e il numero di satelliti naturali conosciuti. Il programma deve quindi stampare a schermo il contenuto della tupla e il numero totale di satelliti.

```
Solution
 1
    pianeti = (
 2
        ("Mercurio", "roccioso", 0),
        ("Venere", "roccioso", 0),
 3
        ("Terra", "roccioso", 1),
 4
        ("Marte", "roccioso", 2),
 5
        ("Giove", "gassoso", 95),
 6
 7
        ("Saturno", "gassoso", 83),
        ("Urano", "gassoso", 27),
 8
 9
        ("Nettuno", "gassoso", 14)
10
    )
11
12
13
    def info_pianeti():
14
        # Stampa a schermo il contenuto della tupla
15
        print("I pianeti del sistema solare sono:")
16
        for pianeta in pianeti:
17
            print(f"{pianeta[0]}: {pianeta[1]}, {pianeta[2]} satelliti")
18
19
        # Calcola il numero totale di satelliti naturali
20
        num_satelliti = sum([pianeta[2] for pianeta in pianeti])
21
22
        # Stampa a schermo il numero totale di satelliti naturali
23
        print("Il numero totale di satelliti naturali conosciuti"
24
              f"nel sistema solare è di {num_satelliti}")
25
```

26 27

info\_pianeti()

### 3 Punti 25. 😑 Sport di squadra e individuali

Scrivi una funzione che prenda come argomento un set di sport preferiti dall'utente e stampi un messaggio di testo che indica se si tratta di uno sport di squadra o individuale.

### ${f Suggerimenti}$

• Per valutare la stringa inserita potrebbe essere utile utilizzare il metodo lower.

```
Solution
1
   def sport(sport_preferito):
       2
3
4
5
                         "nuoto", "ciclismo", "judo"}
6
7
       if sport_preferito.lower() in sport_squadra:
8
          print("Il tuo sport preferito è uno sport di squadra.")
9
       elif sport_preferito.lower() in sport_individuale:
10
          print("Il tuo sport preferito è uno sport individuale.")
11
12
          print("Mi dispiace, non conosco questo sport.")
13
14
15
   sport("tennis")
```

Esercizio	1	2	3	4	5	6	7	8	9	10	11	12	13
Punti	3	9	3	3	3	3	3	3	3	2	3	4	5
Esercizio	14	15	16	17	18	19	20	21	22	23	24	25	Totale
Punti	12	4	2	2	9	2	3	3	2	2	2	3	93