

I seguenti esercizi possono essere risolti con l'ausilio del computer ma molti di questi anche con solo un attento ragionamento. L'importante è che tutti i risultati siano riportati sul foglio, qualunque codice rimasto sul computer non potrà essere valutato.

Sarà possibile ottenere 12 punti per sezione (max 36 punti). Qualunque voto uguale o superiore a 32 equivale ad una lode. Un esercizio può fornire anche solo una parte dei suoi punti, a seconda della qualità dello svolgimento.

A - Python core (12 punti)

A1) Scrivi una list comprehension che data una lista composta da stringhe restituisca una lista con le stringhe **con due caratteri in meno in coda**: (2 punti)

(es. ['ACCG', "GAAC", 'GGGGG', 'CNAAA']
-> ['AC', "GA", 'GGG', 'CNA']

A2) Genera e scrivi tutti gli "etero-dimeri" possibili composti dai quattro nucleotidi. L'output deve essere una serie di stringhe - "AG", "GC" etc. (3 righe di codice: 3 punti, altrimenti a scendere) [hint: usa itertools e il metodo <string>.join()]

A3) alpha="ARNDCSEQHILKMFSTWYV". Che differenza c'è tra itertools.product(alpha, repeat=2) e itertools.combinations(alpha, 2)? (2 punti)

A4) Trova, spiega l'errore e descrivi come lo risolveresti: (1 punto)

```
info = ["meth1 50", "meth2 90", "meth2 85"]
mydict = {}
for row in info:
    spl = row.split()
    method=spl[0]
    score=spl[1]
```

```
mydict[method].append(score)
```

A5) Hai una funzione definita così:

```
def myfunc(a, b):
    <do things with a and b>
```

questa funzione è inserita all'interno di un codice più lungo in cui viene chiamata varie volte.
e.g.

```
<...>
myfunc(varA, varB)
<...>
```

Modifica la definizione della funzione in modo che possa accettare anche altri argomenti senza dover modificare le chiamate alle funzioni già inserite nel codice. (2 punti)

A6) Supponiamo di aprire un file in formato FASTA ciclando sul file object con un *for*:

```
f = open(<filepath>)
for line in f:
    <code>
```

come faccio a selezionare tutte le righe contenenti le sequenze (e non gli ID)? (2 punti)

B - NumPy (12 punti)

B1) Hai un vettore numerico con molti "NaN". Vuoi calcolarne la media **ma senza modificarlo**. Chiaramente non voglio che l'output sia "NaN". Che faccio? (1 punti)

B2) Dato l'array:

```
arr = np.array([12, 10, 0, 20, 'patientA'])
```

scrivi un comando che seleziona, con una maschera booleana, gli elementi con valore intero < 11 (2 punti)

B3) Cosa succede se sommo due **array numpy** fatti così: $[[4,4,4], [3,2,1]] + [3,2,1]$? Come si chiama questa "proprietà"? Come dovrebbe essere il secondo addendo per avere in output $[[4,4,4],[4,4,4]]$? (2 punti)

B4) NotAnotherZebra - Risolvi il seguente problema:

Tre bambini: Bryan, Sean, Tony

Tre supereroi: Batman, Spiderman, Superman

Tre età: 6, 8 e 10 anni

1. A Bryan piace Spiderman
2. A Tony non piace superman
3. Al più giovane piace Spiderman
4. Il bambino a cui piace Superman ha 8 anni

Riporta le tre triplette di risposta (Età di ogni bambino e supereroe preferito)

Anche se non sei sicuro del risultato scrivi il procedimento (7 punti)

C - Pandas (10 punti)

Supponi di avere un Pandas DataFrame in una variabile *df*:

country	beer_servings	spirit_servings	wine_servings	continent
Afghanistan	0	0	0	AS
Albania	89	132	54	EU
Algeria	25	NaN	14	AF
Andorra	245	138	312	EU
Angola	217	57	45	AF
...

Scrivi un comando che:

C1) crei un'ulteriore colonna ('funWithFlags') che contenga in ogni riga : il quadrato del valore di *wine_servings* se < 100, altrimenti il logaritmo naturale. (2 punti)

C2) Rimuova le righe contenenti NaNs (assumendo che i NaNs siano in una forma riconoscibile dai metodi di pandas, come ad esempio np.nan) (2 punti)

C3) Se volessi modificare il DataFrame **creandone una copia** da assegnare ad una variabile, qual è l'opzione da utilizzare e dove va inserita? (2 punti)

C4) Restituisca le medie del consumo di birra per continente (pandas way)? (2 punti)

C5) Restituisca il consumo di vino dei paesi Europei (niente medie, voglio i dati paese per paese) (2 punti)

D - Plotting (2 punti)

D1) Dato il vettore di numeri reali *data*, se scrivo

```
fig, ax = plt.figure(figsize = (20,20))
```

```
ax.plot(data)
```

ottengo un errore: spiega cos'è e come risolverlo (2 punti)