

Esercizio 1: Scrivere un file binario in Fortran (o C) e leggere il suo contenuto con numpy.

Nel file binario scrivere ad esempio

```
do i = 1, 10
  data(i) = ((i - 1) * 5) + (i + 1)
  write(6,*) stringa,i,data(i)
  write(1) stringa,i,data(i)
enddo
```

dove “stringa”, è una stringa, “i” è un intero e “data” è reale

N.B. In Fortran fare attenzione al record del tipo:
nbyte record, nbyte stringa, nbyte intero, nbyte real, nbyte record

Esercizio 2: leggere il file fort.11 (timestep, angolo) con numpy e costruire il grafico della distribuzione degli angoli tra 0 – 360 gradi a step di 5 gradi. Infine, stampare a video il grafico della distribuzione.

hint: usare la funzione np.histogram

Esercizio 3: Dato un gioco nel quale si vince se la somma di 4 dadi è minore di 10, determinare (forza bruta) se conviene giocarci qualora si vinca 10 volte il valore della posta.

hint: usare np.random.randint per creare un array bidimensionale di interi random con shape (4, n)

Esercizio 4: Calcolare l'integrale Monte-Carlo unidimensionale di una funzione data (ad es $f(x)=1+2x$)

$$\int_a^b f(x) \approx \frac{b-a}{n} \sum_{i=1}^n f(x_i)$$

dove i punti x_i sono numeri random uniformemente distribuiti nell'intervallo a, b . Elaborare un programma python che implementi tale integrazione con un loop esplicito e tramite vettorizzazione via numpy. Infine, testare il cpu time delle due porzioni del programma tramite la funzione `time()` / `%timeit`

hint: `np.random.uniform()`

Esercizio 5: leggere le concentrazioni di CO_2 sul monte Cimone. Fare un plot delle medie annuali (ultima colonna) contro l'anno. Infine, fare una interpolazione lineare di questi dati e stamparla a video con matplotlib

hint: `np.linalg.lstsq()`