➜ Script pour générer une liste aléatoire de 10 nombres entre -2 et 3 :

```
import numpy.random as rd
L=list(rd.randint(-2,4,size=10))
```

Exercice 1

1. Compléter le script de la fonction nb_neg(L) qui compte le nombre de termes négatifs de la liste L :

```
def nb_neg(L):
    compteur = 0
    for e in ...:
        if e<=0:
            ...
    return compteur
```

2. Proposer une variante pos_neg(L) qui retourne la position des éléments négatifs de la liste L.

Exercice 2 *Listes définies par compréhension*

Définir en une instruction les objets suivant :

- la liste L1 des entiers naturels multiples de trois inférieur à 1000 : $[0,3,6,\ldots,999]$
- la liste L2 décroissante des pairs entre 0 et 1000 : $[1000,998,\ldots,2,0]$
- la liste L3 contenant $\left\{\frac{1}{k^2}; k \in [\![1,12]\!]\right\}$
- la somme S3 $= \sum_{k=1}^{12} \frac{1}{k^2}$
- la fonction f3 : $n \mapsto \sum_{k=1}^{n} \frac{1}{k^2}$
- la somme S4 $= \sum_{k=1}^{499} \frac{\cos(2k+1)}{2k+1}$

  On trouve S4 = -0.2375...
- la fonction pair(L) qui retourne la liste des éléments pairs de L une liste d'entiers
- la fonction nb_neg(L) qui retourne le nombre de termes négatifs de L une liste d'entiers

Exercice 3 *Minimum - Tri par sélection* **Prog**

1. Écrire une fonctions min(L) qui retourne la valeur du minimum de la liste L.
2. Donner une variante pmin(L) qui retourne la position et la valeurs d'un minimum de L.
3. Trier une liste revient à créer une nouvelle liste contenant les éléments ordonnés de la première.

Proposer une fonction tri(L) qui trie une liste L en lui retirant son minimum, successivement.

4. Décrire le déroulement ce tri appliqué à la liste [5,-2,-4,3,-4,-1] en complétant le tableau ci-dessous :

boucle	L	pmin(L)	T
X	[5,-2,-4,3,-4,-1]	X	[]
1	...	...	...
⋮	⋮	⋮	⋮

Exercice 4 *Extrema - Médiane*

1. Donner une fonction extremum(L) qui retourne les positions d'un minimum et d'un maximum de la liste L.
2. Définir une fonction mediane(L) qui calcule la médiane d'une liste d'entiers en appliquant l'algorithme qui suit :

a) Tant que la liste contient au moins 3 éléments :

i. Identifier un maximum et un minimum de la liste,

ii. S'ils ont la même valeur, tous les éléments sont égaux : sortir de la boucle en utilisant l'instruction break

iii. Sinon, supprimer une occurrence de chacune de ces valeurs (en utilisant les méthodes pop ou remove)

b) Lorsque l'on sort de la boucle while, soit la liste contient deux éléments, soit tous les éléments sont égaux (que la liste ne contienne qu'un seul élément ou plusieurs). La médiane est la valeur moyenne du premier élément de la liste liste[0] et du dernier élément liste[-1].

Exercice 5

1. Écrire un procédure Pascal(n) qui affiche les $n+1$ première ligne du triangle de Pascal.

Par exemple, pour n=5 cela donne :

```
n = 0 : [1]
n = 1 : [1, 1]
n = 2 : [1, 2, 1]
n = 3 : [1, 3, 3, 1]
n = 4 : [1, 4, 6, 4, 1]
n = 5 : [1, 5, 10, 10, 5, 1]
```

2. Écrire une fonction binome(k,n) qui retourne $\binom{n}{k}$.

Lycée Saint-Exupery – maffre.saintexupery@gmail.com – MPSI

Exercice 6 Que fait la fonction suivante :

```
def f(L):
    T=[L[0]]
    for e in L:
        if not(e in T):
            T.append(e)
    return T
```

Exercice 7 *Ordre lexicographique* **Classique**

On peut utiliser le principe du dictionnaire pour définir un ordre total sur les listes d'entiers. Considérant deux listes L1 et L2. On dit que L1 ≤ L2 si l'une des assertions suivantes est vérifiée :

- `len(L1)` ≤ `len(L2)` et

pour tout $i \in [\![0, \texttt{len(L1)-1}]\!]$, `L1[i]` = `L2[i]`

- il existe `k` = $\min\{\texttt{i};\texttt{L1[i]} \neq \texttt{L2[i]}\}$ et `L1[k]` < `L2[k]`

Écrire une fonction `lexico(L1,L2)` qui retourne `True` si L1 ≤ L2 et `False` sinon.

`lexico([1,0,1],[1,1,-4])` retourne `True`
`lexico([1,0,0],[0,3])` retourne `False`
`lexico([1,2],[1,2,0])` retourne `True`

Exercice 8 *Tri par comptage* **Prog**

1. On considère une liste d'entiers appartenant à $[\![0, N]\!]$.

On souhaite écrire une fonction `stat(L,N)` qui retourne une liste de $N+1$ éléments contenant à la position k le nombre d'occurrences de k dans L.

`stat([1,0,0,2,1],3)` retourne `[2,2,1,0]`

Que retournerait l'appel `stat([2,1,1,0,1,4],4)` ?

2. Donner `stat(L,N)`.
3. Dénombrer le nombre tests et affections effectués.
4. En déduite une fonction `tri_comptage(L,N)` qui retourne une liste triée associée à L.

Exercice 9 *Tri par comparaisons*

1. On veut définir une fonction `comparaison(L)` qui prend en entrée une liste de nombre, L, et qui renvoie une liste C telle que, pour tout i, `C[i]` compte le nombre d'éléments $j \neq i$ tels que :

- `L[j]<=L[i]` lorsque $j < i$,
- `L[j]<L[i]` lorsque $j > i$,

Que retournerait l'appel `comparaison([2,0,1,4,2,3,1,0` ?

2. Donner la fonction `comparaison(L)`.
3. Dénombrer le nombre tests et affections effectués.
4. On peut vérifier que `liste_comparaison[i]` est l'indice de l'élément `L[i]` lorsque l'on trie L dans l'ordre croissant (en remarquant que les répétitions éventuelle sont prises en compte).

En déduire une fonction `tri_comparaison(L)` qui retourne une liste triée associée à L.