

## INTERROGATION DE PYTHON N°4

### EXERCICE

**10 points**

On considère une particule sautant sur les sommets d'un triangle  $ABC$  selon le protocole suivant :

- à l'instant  $n = 0$ , la particule est sur le sommet  $A$ .
- si à l'instant  $n$  la particule est sur un sommet, alors elle reste sur place ou saute sur l'un des deux autres sommets du triangle du façon équiprobable.

On définit  $X_n$  la variable aléatoire définie par

$$X_n = \begin{cases} 1 & \text{si la particule est au sommet } A \text{ à l'instant } n \\ 2 & \text{si la particule est au sommet } B \text{ à l'instant } n . \\ 3 & \text{si la particule est au sommet } C \text{ à l'instant } n \end{cases}$$

1. Si la particule est au sommet  $B$  à l'instant  $n$  quelle est la probabilité qu'elle aille en  $A$ , en  $C$  ou qu'elle reste en  $B$  ?

.....

2. On rappelle que la commande `rd.randint(a, b)` renvoie un nombre aléatoire entre  $a$  et  $b-1$  suivant la loi uniforme  $\mathcal{U}[[a; b-1]]$ .

Compléter la fonction Python prenant en argument un entier  $n$  et renvoyant en sortie une liste de nombres entier dans l'intervalle  $[[1;3]]$  représentant les  $n$  sommets parcourus par la particule au cours des ses  $n$  premiers sauts.

Par exemple une sortie possible de la commande `deplacements(10)` est la liste `[1, 3, 3, 3, 2, 1, 2, 3, 3, 1, 3]`.

```

1 import ..... as .....
2
3 def deplacements(n):
4     L = [ ..... ]
5     for _ in range(n):
6         L.append( ..... )
7     return L
```

3. Comment obtenir  $X_n$  à l'aide de la fonction `deplacements(n)` ? On rappelle que si  $L$  désigne une liste alors `L[-1]` donne le dernier élément de la liste.

.....

4. (a) Compléter la fonction suivante permettant de donner une estimation de l'espérance de la variable  $X_{50}$  sur un nombre `nbsim` de simulation de la variable :

```

1 import ..... as .....
2
3 def estim_EspX(nbsim):
4     L = [ ..... for _ in range( ..... )]
5     return np ..... ( L)
```

- (b) En entrant `estim_EspX(1000)` on trouve 1,699. Comment interpréter ce résultat ?

.....

.....

5. Compléter la fonction suivante qui, étant donné un nombre de simulations `nbsim`, renvoie la liste `frequencies` donnant les fréquences du nombre de fois où la particule est retournée au sommet *A*, *B* ou *C* au cours de ses 100 premiers déplacements :

```

1 def frequencies(nbsim):
2     L=[deplacements( ..... )[-1] for _ in range( ..... )]
3     frequencies=[0.,0.,0.]
4     for k in range(nbsim):
5         j=L[ ..... ]
6         frequencies[j-1]+= .....
7     return frequencies

```

6. On rappelle que dans la bibliothèque `matplotlib.pyplot` la commande `plt.bar(X,L)` renvoie le diagramme en bâtons des valeurs de la liste `L` en fonction des valeurs de la liste `X`.

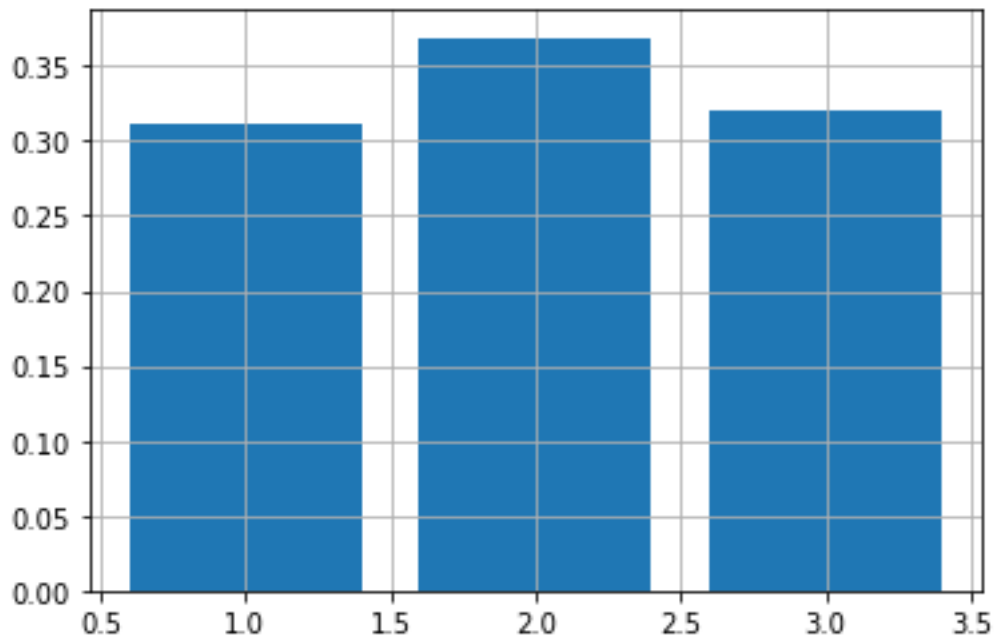
Compléter les commandes suivantes permettant d'afficher le graphique ci-après :

```

1 import ..... as .....
2
3 nbsim = 1000
4 plt.hist([ ..... , ..... , ..... ], ..... )
5 plt.grid()
6 plt.show()

```

Lorsque l'on exécute ces commandes on obtient le graphique suivant :



Interpréter ce résultat en utilisant un langage probabiliste dans le contexte de l'exercice.

.....

.....

.....

.....