Situation 1

Un kangourou se trouve en un point d'une route déserte d'Australie occidentale. A chaque seconde il saute d'un mètre vers l'avant ou d'un mètre vers l'arrière avec la même probabilité.

1. Recopier et compléter la fonction « millesauts » proposée ci-contre qui simule le déplacement du kangourou après avoir effectué mille sauts.

2. Exécuter la fonction « millesauts » plusieurs fois. Quelle est la plus grande valeur renvoyée (en valeur absolue) ?

```
from random import random
def millesauts():
  position=0
  for i in ... :
    if ... :
      position=position+1
    else:
      position=position-1
  return position
```

3. En utilisant la fonction précédente, écrire le script d'une fonction « centmetres(n) » qui simule « n » expériences de mille sauts et qui renvoie le nombre de fois où le kangourou s'est éloigné de plus de 100 mètres de sa position initiale au bout des mille sauts lors des « n » expériences. Exécuter la fonction « centmetres(n) » pour n=10000. Commenter.

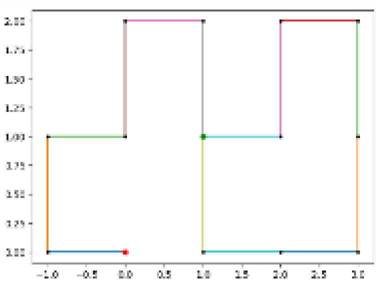
Situation 2

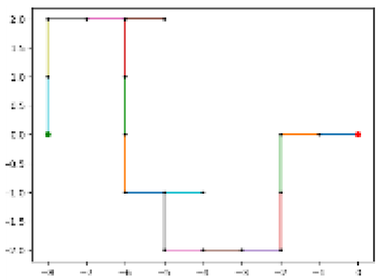
Le plan est muni d'un repère orthonormé $\left(O,\vec{i},\vec{j}\right)$.

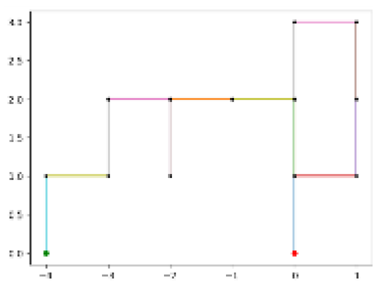
On place un robot à l'origine du repère. A chaque seconde, le robot se déplace de façon aléatoire en effectuant une translation de vecteur $\vec{i}$, $-\vec{i}$, $\vec{j}$ ou $-\vec{j}$, chaque déplacement ayant la même probabilité.

On souhaite écrire une fonction « marcealeatoire(n) » qui simule le déplacement du robot au cours du temps et propose un graphique. Pour cela, compléter le script ci-contre dans lequel on a stocké les tirages aléatoires d'un entier compris entre 1 et 4 dans une liste, les abscisses dans une autre et les ordonnées dans une troisième.

```
from random import randint
import matplotlib.pyplot as plt

def marchealeatoire(n):
  abs=[0]
  ord=[0]
  r=[]
  for i in range(n):
    r.append(randint(1,4))
    if r[i]==1:
      abs.append(...)
      ord.append(...)
    if r[i]==2:
      abs.append(...)
      ord.append(...)
    if r[i]==3:
      abs.append(...)
      ord.append(...)
    if r[i]==4:
      abs.append(...)
      ord.append(...)

  plt.plot(abs[0],ord[0],"r.",ms=10)
  plt.plot(abs[-1],ord[-1],"g.",ms=10)

   for k in range(n):
    plt.plot([abs[k],abs[k+1]],[ord[k],ord[k+1]])
    plt.plot(abs[k],ord[k],"k.",ms=5)

  plt.show()
```



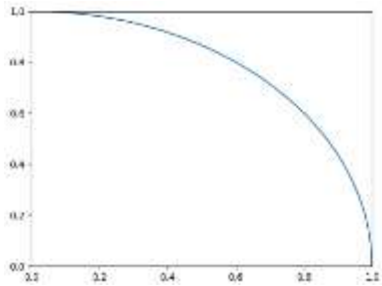




Situation 3

Dans un repère on considère la courbe représentative du quart de cercle de centre O et rayon 1.

```python
import math
import random
import matplotlib.pyplot as plt

plt.axis(xmin=0,xmax=1,ymin=0,ymax=1)
abs=[x/300 for x in range(301)]
ord=[math.sqrt(1-x*x) for x in abs]
plt.plot(abs,ord)
plt.show()
```

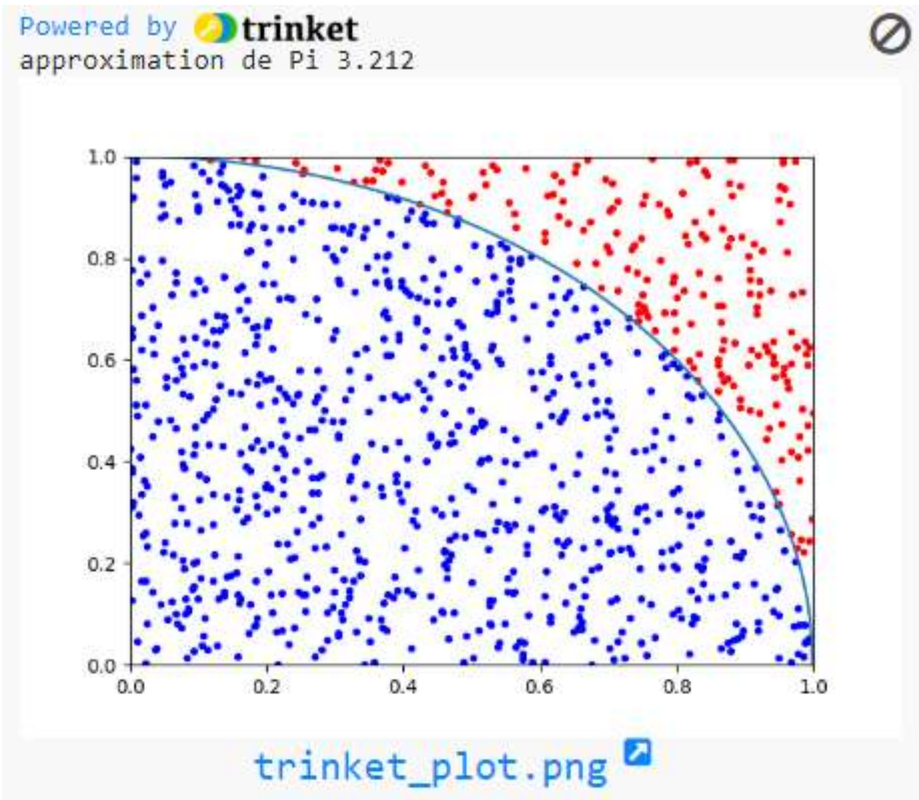


Quelle est la probabilité qu'un point M pris au hasard parmi les points d'abscisses et d'ordonnées comprises entre 0 et 1 se trouve dans le quart de disque ?

En reprenant le script proposé ci-dessus et en exploitant l'algorithme écrit en langage naturel proposé ci-contre, programmer la fonction « montecarlo(n) » prenant pour argument le nombre de points à générer aléatoirement et renvoyant une approximation du nombre Pi.

Saisir N
$p \leftarrow 0$
Pour i allant de 1 à N
$x \leftarrow$ un nombre aléatoire entre 0 et 1
$y \leftarrow$ un nombre aléatoire entre 0 et 1
Si $x^2 + y^2 < 1$ alors $p \leftarrow p+1$
Afficher $\frac{p}{N} \times 4$

Compléter la fonction « montecarlo(n) » afin qu'elle affiche en bleu les points générés aléatoirement contenus dans le quart de disque et en rouge générés aléatoirement situés à l'extérieur du disque. On prendra pour exemple la capture d'écran proposée ci-dessous :



Situation 4

A la cour de Florence, au XVII siècle, de nombreux jeux de société étaient pratiqués. Parmi ceux-ci, l'un faisait intervenir la somme numéros sortis lors du lancer de trois dés. Le Duc de Toscane, qui avait sans doute observé un grand nombre de parties de ce jeu, avait constaté que la somme « 10 » était obtenue légèrement plus souvent que la somme « 9 ».

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | dé1 | dé2 | dé3 | somme | | somme | appartions | fréquences |
| 2 | 4 | 5 | 6 | 15 | | "9" | 48 | 0,096 |
| 3 | 2 | 6 | 2 | 10 | | "10" | 79 | 0,158 |
| 4 | 5 | 5 | 3 | 13 | | | | |
| 5 | 6 | 4 | 5 | 15 | | | | |
| 6 | 4 | 2 | 2 | 8 | | | | |
| 7 | 5 | 5 | 5 | 15 | | | | |
| 8 | 5 | 6 | 2 | 13 | | | | |
| 9 | 2 | 4 | 5 | 11 | | | | |

On souhaite réaliser une feuille de calcul comme ci-dessus pour effectuer la simulation de 500 lancers de trois dés. La touche F9 permet d'obtenir d'autres simulations : les résultats confirment-ils l'observation faite par le duc de Toscane ? Vérifier ce résultat en calculant les probabilités respectives d'obtenir une somme égale à 9 et d'obtenir une somme égale à 10. S'aider d'un arbre.

Situation 5

Une dame nourrit son chat Gédéon avec des aliments en boîte.

Chaque jour elle choisit une boîte au hasard parmi les trois variétés : volaille, bœuf ou lapin.

Cette dame a remarqué que :

- Si on lui sert de la volaille, Gédéon finit toujours sa gamelle.
- Si on lui sert du bœuf, Gédéon finit sa gamelle une fois sur deux.
- Si on lui sert du lapin, Gédéon finit sa gamelle une fois sur trois seulement.

Créer une feuille de calcul permettant de simuler 500 repas servis à Gédéon. La touche F9 permet d'obtenir d'autres simulations.

Vers quelle valeur semble se stabiliser la fréquence de l'événement « Gédéon finit sa gamelle ». Retrouve-t-on le résultat théorique ?

| | | |
|---|---|---|
| 1 | volaille | finit |
| 2 | boeuf | finit |
| 3 | lapin | finit |
| 2 | boeuf | ne finit pas |
| 3 | lapin | ne finit pas |
| 2 | boeuf | finit |
| 3 | lapin | finit |
| 3 | lapin | finit |
| 2 | boeuf | finit |
| 1 | volaille | finit |
| 3 | lapin | ne finit pas |
| 2 | boeuf | finit |
| 3 | lapin | finit |
| 2 | boeuf | finit |
| 2 | boeuf | ne finit pas |
| 2 | boeuf | ne finit pas |
| 1 | volaille | finit |
| 3 | lapin | ne finit pas |
| 2 | boeuf | ne finit pas |
| 1 | volaille | finit |
| 2 | boeuf | ne finit pas |
| 2 | boeuf | finit |
| 3 | lapin | ne finit pas |
| 1 | volaille | finit |
| 1 | volaille | finit |