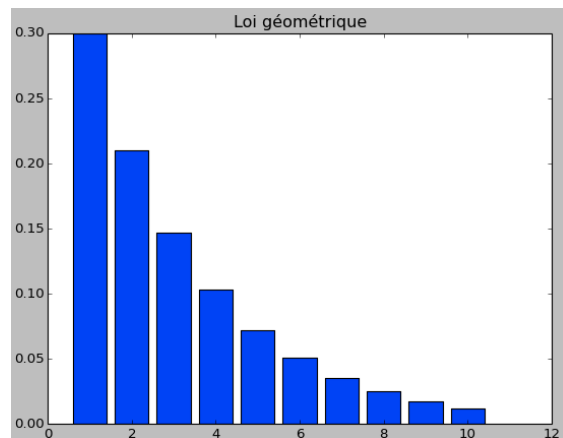


```
# Importation des modules nécessaire
import numpy as np
import matplotlib.pyplot as plt
from random import random
```

```
# Exercice 1
```

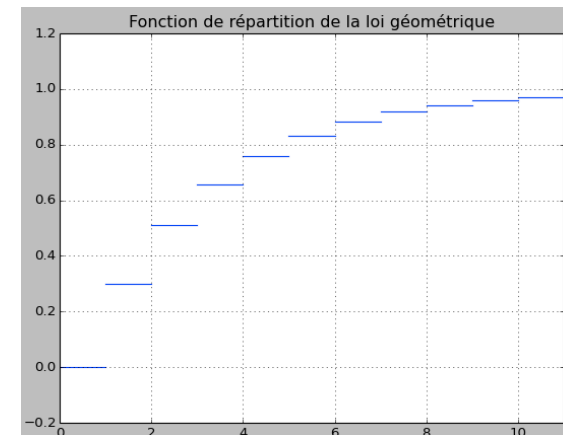
```
# (1.1)
def loiGeometrique(p,k):
    return (1-p)**(k-1) * p
```

```
# (1.2)
K = np.arange(1,11)
yK = loiGeometrique(0.3,K)
xK = [k - 0.4 for k in K]
plt.figure("Exercice 3 : loi géométrique")
plt.title("Loi géométrique")
plt.bar(xK,yK)
plt.show()
```



```
# (1.3)
plt.figure("Exercice 3 : fonction de répartition")
```

```
plt.title("Fonction de répartition de la loi géométrique")
P = 0
plt.plot([0,1],[P,P],'b')
for k in range(1,11):
    P += yK[k-1]
    plt.plot([k,k+1],[P,P],'b')
plt.axis([0,11,-0.2,1.2])
plt.grid()
plt.show()
```



```
# Exercice 2
```

```
# (2.1)
def simulGeometrique(p):
    rang = 0
    tir = 1
    while tir > p:
        rang += 1
        tir = random()
    return rang
```

```
# (2.2)
def simulEspGeometrique(p,n):
    S = 0
    for k in range(n):
        S += simulGeometrique(p)
    return S/n
```

```
# (2.3)
def simulLoiGeometrique(p,k,n):
    compteur = 0
    for i in range(n):
        if simulGeometrique(p) == k:
            compteur += 1
    return compteur/n

# (2.4)
def freqGeometrique(p,N):
    R = []
    for i in range(N):
        n = len(R)
        tir = simulGeometrique(p)
        if tir <= n:
            R[tir-1] += 1
        else:
            R = R + [0]*(tir-n)
            R[tir-1] += 1
    return [k/N for k in R]

# (2.5)
R = freqGeometrique(0.3,1000)
n = len(R)
X = [(1-0.4 + k) for k in range(n)]
plt.figure("Exo 2.4")
plt.title("Fréquence d'apparition des 1er succès")
plt.bar(X,R)
plt.show()
```

Ce qui produit :

