

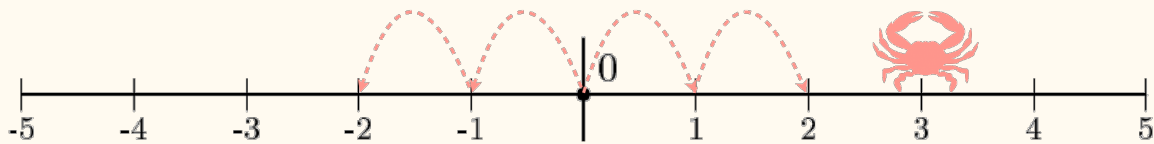


# La marche du crabe

## Un exemple de marche aléatoire

Nom : .....  
 Prénom : .....  
 Classe : .....  
 Date : .....

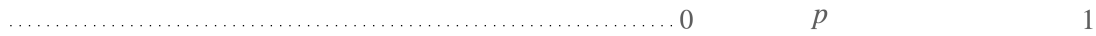
On considère un crabe qui se déplace horizontalement sur le segment  $[-x, x]$ . A l'instant initial, le crabe est à l'origine. A chaque instant, il se déplace de 1 unité vers la droite avec probabilité  $p$  et de 1 unité vers la gauche avec probabilité  $1 - p$ . A  $x$  unités vers la droite et à  $x$  unités vers la gauche se trouvent les extrémités : lorsque le crabe atteint l'une de ces deux extrémités, il s'arrête.



### Nombre de déplacements et extrémité atteinte en fonction de $p$

1 **Question préliminaire** : La fonction `random()` du module `random` renvoie un nombre réel au hasard entre 0 et 1. En vous aidant du schéma ci-dessous, donner la probabilité

- que le nombre renvoyé soit dans la zone rouge :



- que le nombre renvoyé soit dans la zone verte :

2 Compléter la fonction `nbDepla(p, x)` suivante pour qu'elle simule la marche du crabe et renvoie

- \* le nombre de déplacements effectués avant d'atteindre une extrémité,
- \*  $x$  ou  $-x$  selon si l'extrémité atteinte lors de l'expérience est celle de droite ou celle de gauche.

```
def nbDepla(p,x) :
    position = ...
    nbD = ...
    while position < ... and position > ... :
        a = rd.random()
        if a < p :
            position = ...
        else :
            position = ...
        nbD = ...
    return ... , ...
```

### Questions

On prend  $p = 1/4$  et  $x = 100$ . Tester plusieurs fois la fonction `nbDepla` pour ces valeurs. Que remarquez-vous?

.....  
 On prend  $p = 1/2$  et  $x = 100$ . Tester plusieurs fois la fonction `nbDepla` pour ces valeurs. Que remarquez-vous?

.....



# La marche du crabe

## Un exemple de marche aléatoire

Nom : .....  
Prénom : .....  
Classe : .....  
Date : .....

### Nombre de déplacements moyen en fonction de $x$

- 3 Modifier la fonction nbDepla pour qu'elle ne renvoie **que** le nombre de déplacements effectués par le crabe.
- 4 Écrire une fonction nbDeplaMoyen( $p, x, n$ ) qui renvoie le nombre de déplacements **moyen** nécessaires pour atteindre une extrémité si l'on répète  $n$  fois l'expérience. On appellera la fonction nbDepla pour obtenir le nombre de déplacements nécessaires pour atteindre une extrémité lors d'**une** marche.

### Questions

Effectuer les tests suivants :

```
nbDeplaMoyen(1/4, 100, 1000)
```

```
nbDeplaMoyen(1/2, 100, 1000)
```

### Une courbe bien connue

- 5 Dans cette question, on prend  $p = 1/2$ . On donne la fonction courbe( $n, xmax$ ) suivante.

```
import matplotlib.pyplot as plt
```

```
def courbe(n, xmax) :
```

```
    Abscisses = [ k for k in range(1, int(xmax)+1) ]
```

```
    Ordonnees = [ nbDeplaMoyen(1/2, x, n) for x in Abscisses ]
```

```
    plt.plot(Abscisses, Ordonnees, 'b-o')
```

```
    plt.title("Temps de sortie moyen en fonction de la distance à 0")
```

```
    plt.xlabel("Distance de la sortie à l'origine")
```

```
    plt.ylabel("Temps de sortie moyen (pour "+str(n)+" simulations)")
```

```
    plt.show()
```

Taper dans le Shell :

```
courbe(20, 20), puis courbe(50, 20), puis courbe(100, 20), puis courbe(1000, 30)
```

### Questions

Que représentent les arguments  $n$  et  $xmax$  de la fonction courbe ?

.....  
.....  
.....  
.....

Quelle courbe reconnaît-on ?

.....  
.....