



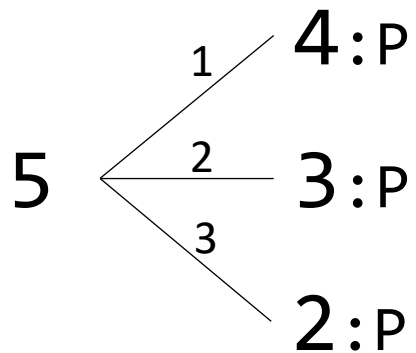
Comment rendre l'ordinateur intelligent?

Un nouvel outil de calcul:
Modulo %

Le jeu: Dans quel cas gagne-t 'on?

Nombre n Allumettes	1	2	3	4	5	6	7	8	9	10	11	12	13
Nombre à prendre	P	1	2	3	P	1	2	3	P	1	2	3	P

Périodique de période 4



Périodicité

Nombre n Allumettes	1	2	3	4	5	6	7	8	9	10	11	12	13
Nombre à prendre	P	1	2	3	P	1	2	3	P	1	2	3	P

Périodique de période 4

Avec 7 et 11, on a un reste de 3

Le reste = (0, 1, 2, 3) < 4

→ Quand le reste de la Division euclidienne de n par 4 vaut 3 -> prendre 2 allumettes

$\Leftrightarrow n\%4 = 3$

$$\begin{array}{r|l} 7 & 4 \\ 3 & 1 \end{array}$$

$$\begin{array}{r|l} 11 & 4 \\ 3 & 2 \end{array}$$

Connaître le reste pour gagner

Nombre n Allumettes	1	2	3	4	5	6	7	8	9	10	11	12	13
Nombre à prendre	P	1	2	3	P	1	2	3	P	1	2	3	P
Reste	0	1	1	1	0	5	5	5	0	9	9	9	0

Le reste = (0, 1, 2, 3) < 4

$n\%4 = 3 \rightarrow$ prendre 2 allumettes

$n\%4 = 2 \rightarrow$ prendre 1 allumettes

$n\%4 = 0 \rightarrow$ prendre 3 allumettes

$n\%4 = 1 \rightarrow$ PERDU

/!\ Avancer dans le tableau de 4 case ne change pas le nombre d'allumettes à prendre

Jeu des Allumettes en python

Joueur



Ordinateur

VS

[1]

```
#-----MODULES-----
import random
import time
```

[2]

```
#-----FONCTION PRINCIPALE-----
def allumettes(n):
    """Cette fonction permet de jouer au jeu des allumettes.
    les joueurs peuvent retirer 1,2 ou 3 allumettes à la fois.
    le perdant est celui qui retire la dernière allumette.
    n est le nombre total d'allumettes au début du jeu.
    t est le nombre de tour qui se déroule durant la partie"""
    t=0
    while n>0:
        t=t+1
        time.sleep(1)
        print("\n- TOUR",t,"-")
        n=joueur(n) #on fait jouer le joueur
        time.sleep(0.7)
        print("Il en reste",n)
        if n<=0:
            time.sleep(1)
            print("Vous avez perdu!")
        else:
            t=t+1
            time.sleep(1)
            print("\n- TOUR",t,"-")
            n=ordi(n) #on fait jouer l'ordinateur
    time.sleep(1); print("\nDurée de la partie:",t,"tours")
    rejouer()
```

[3]

```
#-----JOUEUR-----
def joueur(n):
    """Cette fonction permet au joueur de jouer au jeu"""
    #on annonce à idle combien on prend d'allumettes avec la variable i
    i=-1
    while i<1 or i>3:
        i=int(input("Combien d'allumettes prenez vous?"))
    if i<n:
        n=n-i
    else:
        n=0
    return(n)
```

[4]

```
#-----ORDINATEUR-----
def ordi(n):
    """Cette fonction permet à l'ordinateur de jouer au jeu"""
    p=random.randint(1, 3) #on génère une valeur entre 1, 2 ou 3 au hasard
    if p<n:
        time.sleep(2)
        print("L'ordinateur en prends",p)
        n=n-p
        time.sleep(0.7)
        print("Il en reste",n)
        return(n)
    else:
        print("L'ordinateur en prends",n)
        time.sleep(1.5)
        print("L'ordinateur à perdu!")
        return(0)
```

[5]

```
#-----MENU JEU-----
def menu():
    """Cette fonction affiche simplement le menu et les règles du jeu"""
    time.sleep(1); print(">>>Bienvenue dans le jeu des allumettes!")
    time.sleep(0.5); print("\n***** But du jeu: *****")
    time.sleep(0.5); print("\n-Choisissez le nombre total d'allumettes en jeu dans la partie.")
    time.sleep(0.5); print("-Decidez d'un nombre (entre 1 et 3) à retirer.")
    time.sleep(0.5); print("-L'ordinateur joue à son tour, ect.")
    time.sleep(0.5); print("-La partie se termine lorsqu'il n'y a plus d'allumettes.")
    time.sleep(0.5); print("Le perdant est celui qui retire la dernière allumette.")
    time.sleep(1); print("\n***** C'est partie! *****")
```

[6]

```
#-----MENU REJOUER-----
def rejouer():
    """Cette fonction permet de relancer une partie"""
    time.sleep(1); print("\n>>>Relancer une partie?")
    time.sleep(0.5); print(">1: Oui")
    time.sleep(0.5); print(">2: Non")
    r=0
    while r!=1 or r!=2:
        time.sleep(0.5); r=int(input("?"))
        if r==1:
            time.sleep(1); print("***** C'est repartie! *****")
            time.sleep(1.5); z=int(input("Avec combien d'allumettes souhaitez vous jouer?"))
            time.sleep(0.5); print("Initialisation du jeu avec",z,"allumettes en cours...")
            allumettes(z)
        elif r==2:
            time.sleep(0.5); print("Merci d'avoir joué!")
```

[7]

```
#-----EXECUTION-----
menu()
time.sleep(1.5); z=int(input("Avec combien d'allumettes souhaitez vous jouer?"))
time.sleep(0.5); print("Initialisation du jeu avec",z,"allumettes en cours...")
allumettes(z)
```

Le code

[7 Blocs]

Bloc [1]

Le code

7 Blocs

```
#-----MODULES-----  
import random  
import time
```

Bloc [2]

Le code

7 Blocs

```
#-----FONCTION PRINCIPALE-----
def allumettes(n):
    """Cette fonction permet de jouer au jeu des allumettes.
    les joueurs peuvent retirer 1,2 ou 3 allumettes à la fois.
    le perdant est celui qui retire la dernière allumette.
    n est le nombre total d'allumettes au début du jeu.
    t est le nombre de tour qui se déroule durant la partie"""
    t=0
    while n>0:
        t=t+1
        time.sleep(1)
        print("\n- TOUR",t,"-")
        n=joueur(n) #on fait jouer le joueur
        time.sleep(0.7)
        print("Il en reste",n)
        if n<=0:
            time.sleep(1)
            print("Vous avez perdu!")
        else:
            t=t+1
            time.sleep(1)
            print("\n- TOUR",t,"-")
            n=ordi(n) #on fait jouer l'ordinateur
    time.sleep(1); print("\nDurée de la partie:",t,"tours")
    rejouer()
```


Bloc [3]

Le code

7 Blocs

```
#-----JOUEUR-----  
def joueur(n):  
    """Cette fonction permet au joueur de jouer au jeu"""  
    #on annonce à idle combien on prend d'allumettes avec la variable i  
    i=-1  
    while i<1 or i>3:  
        i=int(input("Combien d'allumettes prenez vous?"))  
    if i<n:  
        n=n-i  
    else:  
        n=0  
    return(n)
```

-
- 1- Demander une valeur `i` d'allumettes à jouer
 - 2- Vérifier si elle est recevable dans le jeu
 - 3- Renvoyer la nouvelle valeur `n` d'allumettes

Bloc [4]

Le code

7 Blocs

```
#-----ORDINATEUR-----
def ordi(n):
    """Cette fonction permet à l'ordinateur de jouer au jeu"""
    p=random.randint(1, 3) #on génère une valeur entre 1, 2 ou 3 au hasard
    if p<n:
        time.sleep(2)
        print("L'ordinateur en prends",p)
        n=n-p
        time.sleep(0.7)
        print("Il en reste",n)
        return(n)
    else:
        print("L'ordinateur en prends",n)
        time.sleep(1.5)
        print("L'ordinteur à perdu!")
        return(0)
```

-
- 1- Générer une valeur aléatoire `p` d'allumettes à prendre
 - 2- Vérifier si elle est recevable dans le jeu
 - 3- Renvoyer la nouvelle valeur `n` d'allumettes

```
#-----MENU JEU-----  
def menu():  
    """Cette fonction affiche simplement le menu et les règles du jeu"""  
    time.sleep(1); print(">>>Bienvenue dans le jeu des allumettes!")  
    time.sleep(0.5); print("\n***** But du jeu: *****")  
    time.sleep(0.5); print("\n-Choisissez le nombre total d'allumettes en jeu dans la partie.")  
    time.sleep(0.5); print("-Decidez d'un nombre (entre 1 et 3) à retirer.")  
    time.sleep(0.5); print("-L'ordinateur joue à son tour, ect.")  
    time.sleep(0.5); print("-La partie se termine lorsqu'il n'y a plus d'allumettes.")  
    time.sleep(0.5); print("Le perdant est celui qui retire la derniere allumette.")  
    time.sleep(1); print("\n***** C'est partie! *****")
```

1- Afficher les règles

```
#-----MENU REJOUER-----
def rejouer():
    """Cette fonction permet de relancer une partie"""
    time.sleep(1); print("\n>>>Relancer une partie?")
    time.sleep(0.5); print(">1: Oui")
    time.sleep(0.5); print(">2: Non")
    r=0
    while r!=1 or r!=2:
        time.sleep(0.5); r=int(input("?"))
        if r==1:
            time.sleep(1); print("***** C'est repartie! *****")
            time.sleep(1.5); z=int(input("Avec combien d'allumettes souhaitez vous jouer?"))
            time.sleep(0.5); print("Initialisation du jeu avec",z,"allumettes en cours...")
            allumettes(z)
        elif r==2:
            time.sleep(0.5); print("Merci d'avoir joué!")
```

-
- 1- Créer une variable `r` correspondant au choix du joueur
 - 2- Vérifier si la réponse est recevable
 - 3- Recommencer ou non une partie

Bloc [7]

Le code

--locs

```
#-----EXECUTION-----  
menu()  
time.sleep(1.5); z=int(input("Avec combien d'allumettes souhaitez vous jouer?"))  
time.sleep(0.5); print("Initialisation du jeu avec",z,"allumettes en cours...")  
allumettes(z)
```

Bloc [4*]:

Le code

7 Blocs

```
#-----ORDINATEUR-----  
def ordi(n):  
    """Cette fonction permet à l'ordinateur de jouer au jeu de manière intelligente"""  
    if n%4==3:  
        p=2  
    elif n%4==2:  
        p=1  
    elif n%4==0:  
        p=3  
    else:  
        p=1  
    print("L'ordinateur en prends",p)  
    n=n-p  
    print("Il en reste",n)  
    if n==0:  
        print("L'ordniateur a perdu!")  
    return(n)
```

**Ordinateur intelligent*

-
- 1- Utiliser modulo % selon la logique du jeu
 - 2- Déterminer le nombre **p** d'allumettes à prendre
 - 3- Renvoyer la nouvelle valeur **n** d'allumettes

