

## Exercices d'algorithmique en première

### Probabilités

#### **Anniversaires** (Boucle Tant que)

```
anniversaires():=  
local n, p, q;  
saisir("Probabilité à ne pas dépasser",q);  
n := 1;  
p := 1;  
tantque p>=q faire  
  p := p*(365-n)/365;  
  n := n+1;  
ftantque  
n := n-1;  
afficher(n);  
}  
::
```

#### **Le lièvre et la tortue S1** (Instructions conditionnelles - Boucle Pour et Tant que)

```
lievre_tortue_v1():=  
local D,T,L,xT,xL,N,k;  
saisir(N);  
T:=0;  
L:=0;  
pour k de 1 jusque N faire  
  xT:=0;  
  xL:=0;  
  tantque xT<2 et xL<2 faire  
    D:=floor(3*alea(0,1))+1;  
    si D ==1 alors  
      xL:=2;  
      L:=L+1;  
    sinon  
      xT:=xT+1;  
      si xT=2 alors  
        T:=T+1;  
      fsi  
    fsi  
  ftantque  
fpour;  
afficher("Tortue : "+evalf(T/N));  
afficher("Lièvre : "+evalf(L/N));  
}  
::
```

## Le lièvre et la tortue S2 (Instructions conditionnelles - Boucle Pour et Tant que)

```
lievre_tortue_v2():=  
local D,T,L,xT,xL,N,k;  
saisir(N);  
T:=0;  
L:=0;  
pour k de 1 jusque N faire  
  xT:=0;  
  xL:=0;  
  tantque xT<6 et xL<6 faire  
    D:=floor(6*alea(0,1))+1;  
    si D ==6 alors  
      xL:=6;  
      L:=L+1;  
    sinon  
      xT:=xT+1;  
      si xT=6 alors  
        T:=T+1;  
      fsi  
    fsi  
  ftantque  
fpour;  
afficher("Tortue : "+evalf(T/N));  
afficher("Lièvre : "+evalf(L/N));  
}  
::
```

## Marche aléatoire (Instructions conditionnelles - Boucles pour imbriquées)

```
marche_aleatoire():=  
local L,xP,N,k,G;  
saisir(N);  
G:=0;  
pour k de 1 jusque N faire  
  xP:=0;  
  pour L de 1 jusque 4 faire  
    si alea(0,1)<0.5 alors  
      xP:=xP+1  
    sinon  
      xP:=xP-1  
    fsi  
  fpour  
  si xP==0 alors  
    G:=G+1  
  fsi  
fpour  
afficher("Fréquence de A : "+evalf(G/N));  
}  
::
```

## Le 6 la première fois (Instructions conditionnelles - Boucle Pour et Tant que)

```
six_prem_fois():=  
local G,D,L,N,k;  
saisir(N);  
G:=0;  
pour k de 1 jusque N faire  
  D:=0;  
  L:=0;  
  tantque D!=6 et L!=4 faire  
    D:=floor(6*alea(0,1))+1;  
    si D==6 alors  
      G:=G+1  
    sinon  
      L:=L+1  
    fsi  
  ftantque  
fpour  
afficher("Fréquence d'obtention du 6 : "+evalf(G/N));  
}  
::
```

## Égalité au jeu (Instructions conditionnelles - Boucles pour imbriquées)

```
egalite_au_jeu():=  
local k,N,G1,G2,E,L;  
saisir(N);  
E:=0;  
pour k de 1 jusque N faire  
  G1:=0;  
  G2:=0;  
  pour L de 1 jusque 100 faire  
    si alea(0,1)<0.5 alors  
      G1:=G1+1  
    sinon  
      G1:=G1-1  
    fsi  
    si alea(0,1)<0.5 alors  
      G2:=G2+1  
    sinon  
      G2:=G2-1  
    fsi  
  fpour  
  si G1==G2 alors  
    E:=E+1  
  fsi  
fpour  
afficher("Fréquence d'égalité des gains : "+evalf(E/N));  
}  
::
```



## Casino (Instructions conditionnelles - Boucles pour imbriquées)

```

casino():={
local k,N,O,J,G,L;
saisir(N);
G:=0;
pour k de 1 jusque N faire
  G:=G-1;
  O:=0;
  J:=0;
  pour L de 1 jusque 5 faire
    si alea(0,1)<0.5 alors
      O:=O+1
    fsi
  fpour
  pour L de 1 jusque 5 faire
    si alea(0,1)<0.5 alors
      J:=J+1
    fsi
  fpour
  si O==J alors
    G:=G+4
  fsi
fpour
afficher("Gain moyen estimé du joueur : "+evalf(G/N,2));
}
;;

```



## Diviseurs (Sous-programme - Instructions conditionnelles - Boucle Pour)

<pre> diviseurs(N):={ local k,nbrediv; nbrediv:=0; pour k de 2 jusque floor(sqrt(N)) faire   si (N-(floor(N/k))*k)==0 alors     nbrediv:=nbrediv+2   fsi fpour si (floor(sqrt(N))^2==N alors   nbrediv:=nbrediv+1 sinon   nbrediv:=nbrediv+2 fsi retourne(nbrediv); } ;; </pre>	<pre> jeu():={ local k,R,N,G,D; saisir(R); G:=0; pour k de 1 jusque R faire   N:=floor(100*alea(0,1))+1;   D:=diviseurs(N);   si D&lt;3 alors     G:=G-6   sinon     si D&lt;=6 alors       G:=G+1     sinon       G:=G+5   fsi fsi fpour afficher("Le gain moyen estimé est de : "+evalf(G/R,2)); } ;; </pre>
---	--