

TD17 : corrigé

ECE3 Lycée Carnot

27 mai 2011

Exercice 1

```
PROGRAM geometrique ;
USES wincrt ;
VAR t : ARRAY[1..99] OF real ; n, i, z : longint ; p : real ;
BEGIN
Randomize ;
WriteLn('Choisissez le nombre de simulations') ;
ReadLn(n) ;
WriteLn('Choisissez le paramètre de la loi') ;
ReadLn(p) ;
FOR i := 2 TO 99 DO t[i] := 0 ;
FOR i := 1 TO n DO
BEGIN
z := 0 ;
REPEAT z := z+1 UNTIL random < p ;
t[z] := t[z]+1 ;
END ;
FOR i := 1 TO 20 DO WriteLn('On a obtenu ',t[i],' fois la valeur ',i) ;
END.
```

Un exemple de résultats obtenus avec $p = \frac{1}{6}$ et $n = 10\,000$:

k	1	2	3	4	5	6	7	8	9
$ X = k $	1 692	1 391	1 099	980	814	656	542	478	412

k	10	11	12	13	14	15	16	17	18	19	20
$ X = k $	319	271	226	171	168	138	120	78	78	61	49

Exercice 2

```
PROGRAM loipascal ;
USES wincrt ;
VAR t : ARRAY[2..99] OF real ; n, i, z : longint ;
BEGIN
Randomize ;
WriteLn('Choisissez le nombre de simulations') ;
ReadLn(n) ;
FOR i := 2 TO 99 DO t[i] := 0 ;
FOR i := 1 TO n DO
BEGIN
z := 0 ;
```

```

REPEAT z := z+1 UNTIL random < 1/3;
REPEAT z := z+1 UNTIL random < 1/3;
t[z] := t[z]+1;
END;
FOR i := 2 TO 20 DO WriteLn('On a obtenu ',t[i],' fois la valeur ',i);
END.

```

Voilà ce qu'on obtient par exemple pour 10 000 simulations, à comparer avec les résultats de l'exercice précédent :

k	2	3	4	5	6	7	8	9
$ X = k $	1 112	1 517	1 450	1 291	1 154	868	689	519

k	10	11	12	13	14	15	16	17	18	19	20
$ X = k $	369	272	222	163	109	84	57	38	27	25	7

Exercice 3

```

PROGRAM mengoli;
USES wincrt;
VAR n,i,k : longint;
BEGIN
Randomize;
WriteLn('Choisissez le nombre de simulations');
ReadLn(n);
FOR i := 1 TO n DO
BEGIN
k := 0;
REPEAT k := k+1 UNTIL random < 1/(k+1);
WriteLn('On a tiré une boule blanche au tirage numéro ',k);
END;
END.

```

Exercice 4

```

PROGRAM ivrogne;
USES wincrt;
VAR t : ARRAY[-99..99] OF longint; n,k,i,j,z : longint;
BEGIN
Randomize;
WriteLn('Choisissez le nombre de pas effectués par l'ivrogne à chaque simulation');
ReadLn(k);
WriteLn('Choisissez le nombre de simulations');
ReadLn(n);
FOR i := -99 TO 99 DO t[i] := 0;
FOR i := 1 TO n DO
BEGIN
z := 0;
FOR j := 1 TO k DO
IF random < 1/2 THEN z := z+1 ELSE z := z-1;
t[z] := t[z]+1;
END;
END;

```

```

FOR i := -k TO k DO WriteLn('On a obtenu ',t[i],' fois la valeur ',i);
END.

```

Voici maintenant un programme qui effectue n simulations du temps d'attente de la j -ème fois où on repasse par la case départ, n et j étant demandés à l'utilisateur :

```

PROGRAM ivrognebis ;
USES wincrt ;
VAR n,i,j,z,a,b : longint ;
BEGIN
Randomize ;
WriteLn('Choisissez le nombre de simulations');
ReadLn(n);
WriteLn('Choisissez le nombre de fois que notre ivrogne doit repasser par son point de départ');
ReadLn(j);
FOR i := 1 TO n DO
BEGIN
z := 0 ; a := 0 ; b := 0 ;
REPEAT
a := a+1 ;
IF random<1/2 THEN z := z+1 ELSE z := z-1 ;
IF z=0 THEN b := b+1 ;
UNTIL b=j ;
WriteLn('Notre ivrogne a effectué ',a,' pas avant de revenir ',j,' fois au départ');
END ;
END.

```

On constate, assez logiquement, que le temps moyen de la ballade augmente quand on augmente j , mais qu'il semble que l'ivrogne finisse par revenir autant de fois qu'on le souhaite à son point de départ (attention tout de même à ne pas prendre un j trop grand si on ne veut pas faire planter notre cher Pascal).