3. On note x et y les nombres respectifs de boules blanches et noires contenues dans l'urne U_1 (x et y sont donc deux entiers compris au sens large entre 0 et 50).

a) Déterminer la probabilité $f(x,y)$ de tirer une boule blanche (on distinguera l'expression générale, fonction des entiers x et y, des deux cas particuliers où l'une des urnes est vide).

b) Justifier que, pour tous entiers x et y compris au sens large entre 0 et 50, on a l'égalité :

$$f(50-x,50-y)=f(x,y) \ .$$

c) Montrer, pour tout entier y de $[0,50]$, l'inégalité :

$$f(0,y)\leq\frac{1}{2}$$

et en déduire que la répartition des boules recherchée est telle que x appartient à $[1,49]$.

d) Quelles sont les répartitions pour lesquelles la probabilité de tirer une boule blanche est égale à 1/2 ?

4. On note n le nombre de boules contenues dans l'urne U_1 ($n=x+y$).

Montrer que l'on peut se restreindre à des valeurs de n inférieures ou égales à 50.

En fait, on peut même se restreindre à des valeurs de n strictement comprises entre 0 et 50 car $f(0,0)=1/4$ et, d'après le résultat de la question **3. d)**, la probabilité de tirer une boule blanche est égale à 1/2 lorsque n est égal à 50.

5. On suppose désormais que les variables x et n varient de façon continue dans l'intervalle $]0,50[$ et on définit la fonction g de deux variables sur $]0,50[\times]0,50[$ en posant :

$$g(x,n)=\frac{x}{n}+\frac{50-x}{100-n} \ .$$

a) Soit n un réel de $]0,50[$.

Déterminer les variations de la fonction $x\mapsto g(x,n)$ sur l'intervalle $]0,50[$ et en déduire l'entier x de $]0,50[$ pour lequel le réel $g(x,n)$ est maximum sous la condition : $x\leq n$.

b) On pose, pour tout réel n de $]0,50[$:

$$h(n)=g(n,n) \ .$$

Déterminer $h'(n)$; dresser le tableau de variations de la fonction h et en déduire la valeur de l'entier n pour laquelle $h(n)$ atteint son maximum.

6. Conclure relativement à l'objectif de l'exercice annoncé en introduction.

Exercice 2 : probabilités et informatique

On se propose d'étudier des algorithmes de simulation de lois de probabilité, écrits en langage Pascal. La procédure **`randomize`** initialise le générateur de nombres aléatoires. Une fois celle-ci appelée, les appels successifs à la fonction **`random`** simulent la réalisation de variables aléatoires indépendantes de même loi de probabilité uniforme sur $]0,1[$. Des programmes demandés, on n'écrira que la partie « décisive » de l'algorithme, à l'image du premier encadré de la page 3.

Dans tout l'exercice, on désigne par :

- $(\Omega, \mathscr{T}, \mathrm{P})$ un espace probabilisé,
- U une variable aléatoire de Ω dans $\mathbb{R}$ de loi de probabilité uniforme sur $]0,\ 1[$,
- X la variable aléatoire de Ω dans $\mathbb{R}$ dont on souhaite simuler des réalisations,
- p un réel strictement compris entre 0 et 1,
- n un entier naturel non nul.

I. Lois binomiales

1. **a)** Écrire un algorithme qui simule la réalisation d'une variable aléatoire X qui suit la loi de Bernoulli de paramètre p.

b) Justifier que l'algorithme ci-dessous simule la réalisation d'une variable aléatoire X qui suit la loi binomiale de paramètres n et p.

```
x:=0;
for k:=1 to n do
   begin
      u:=random;
      if u < p then x:=x+1;
   end;
write(x);
```

Quel est le nombre d'appels à la fonction **random** ?

2. Soient p_1 et p_2 des réels strictement compris entre 0 et 1. On pose : $q_1 = 1 - p_1$ et $q_2 = 1 - p_2$.
On effectue n épreuves de Bernoulli indépendantes, de paramètres p_1. Y désignant le nombre de succès lors de cette première série d'épreuves, on effectue Y épreuves de Bernoulli indépendantes, de paramètres p_2. On s'intéresse alors à la variable aléatoire X donnant le nombre de succès dans cette seconde série d'épreuves.

a) Terminer le programme de simulation de X ci-dessous.

```
y:=0;
for k:=1 to n do
   begin
      u:=random;
      if u < p1 then y:=y+1;
   end;
x:=0;
for k:=1 to *****
      *************
            :
      *************
write(x);
```

b) **i)** Préciser la loi de Y et l'ensemble des valeurs prises par X.

ii) Soit i un entier compris au sens large entre 1 et n.
Quelle est la loi de X conditionnée par l'événement $(Y = i)$?

iii) Vérifier que, pour tous entiers i et j tels que : $0 \leq j \leq i \leq n$, on a l'égalité :

$$\binom{n}{i} \times \binom{i}{j} = \binom{n}{j} \times \binom{n-j}{i-j},$$

puis, pour tout entier j compris au sens large entre 0 et n, établir, en utilisant la formule des probabilités totales :

$$\mathrm{P}(X=j)=\binom{n}{j}(p_1p_2)^j\sum_{i=j}^{n}\binom{n-j}{i-j}(p_1q_2)^{i-j}q_1^{\,n-i}.$$

iv) Après avoir simplifié le résultat précédent, reconnaître la loi de X.

c) Déterminer le nombre moyen d'appels à la fonction **`random`** du programme de simulation de X.

II. Méthode d'inversion

Cette méthode repose sur la détermination d'une fonction Q telle que $Q(U)$ suive la même loi que X.

(On rappelle que U est une variable aléatoire de loi de probabilité uniforme sur $]0,\ 1[$.)

1. Loi exponentielle

Dans cette question, X suit la loi exponentielle de paramètre a (a réel strictement positif).

a) Démontrer que les variables aléatoires X et $-\dfrac{1}{a}\ln(1-U)$ ont la même fonction de répartition.

b) Justifier alors l'algorithme de simulation de X suivant :

```
u:=random;
x:=-ln(u)/a;
write(x);
```

2. Loi géométrique

Dans cette question, X suit la loi géométrique de paramètre p.

a) En considérant que X est le temps d'attente du premier succès dans un processus de Bernoulli, écrire en langage Pascal un programme de simulation de X qui utilise une boucle **`repeat`**.

Quel est le nombre moyen d'appels à la fonction **`random`** ?

b) La méthode précédente étant « coûteuse » pour les petites valeurs de p, on se propose d'écrire un nouvel algorithme qui n'appelle qu'une seule fois la fonction **`random`**.

i) On suppose que $Y:\Omega\to\mathbb{R}$ est une variable aléatoire qui suit la loi exponentielle de paramètre a.

- Prouver que, pour tout entier naturel k non nul : $\mathrm{P}(k-1\le Y<k)=(\mathrm{e}^{-a})^{k-1}(1-\mathrm{e}^{-a})$.
- En déduire que l'on peut choisir a tel que X et la variable aléatoire $\mathrm{Ent}(Y)+1$ suivent la même loi (on rappelle que Ent désigne la fonction partie entière).

ii) Écrire l'algorithme correspondant. On rappelle qu'en langage Pascal, si une variable **`x`** contient un réel positif ou nul, **`trunc(x)`** retourne la partie entière de ce réel.

3. Cas d'une loi discrète finie

On suppose ici que X prend un nombre fini n de valeurs $x_1,\dots,x_n$, avec $x_1<x_2<\dots<x_n$.

On pose : $F_0=0$ et, pour tout entier k compris au sens large entre 1 et n :

$$p_k=\mathrm{P}(X=x_k)\,,\quad F_k=\mathrm{P}(X\le x_k)\,.$$

Soit Q la fonction « en escalier » de $]0,\ 1[$ dans $\mathbb{R}$ définie par :

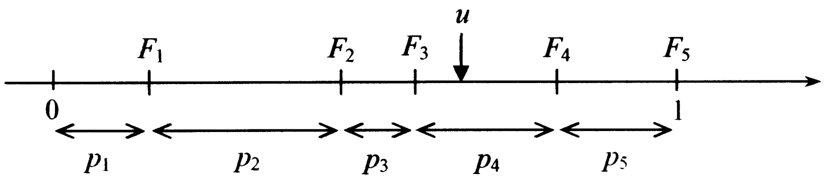
pour tout entier k compris au sens large entre 1 et n, $Q(u)=x_k$ si $F_{k-1}<u\le F_k$.

Q est appelée la fonction quantile de X.

a) Vérifier que, pour tout entier k compris au sens large entre 1 et n, on a : $\mathrm{P}\big(Q(U)=x_k\big)=p_k$.

b) D'après la question précédente, $Q(U)$ et X ont la même loi. Programmer une simulation de X revient donc à trouver, pour une réalisation u de U, le dernier indice k tel que : $u>F_{k-1}$, et décider que X prend la valeur x_k.

i) Dans la situation représentée par le schéma ci-dessous, quelle est la valeur prise par X ?



ii) En considérant que les variables **F[1]**,..., **F[n]** d'un tableau **F** contiennent les valeurs $F_1,\dots,F_n$, celles **T[1]**,..., **T[n]** d'un tableau **T** les valeurs $x_1,\dots,x_n$, compléter le programme de simulation de X ci-dessous, et démontrer que le nombre moyen m de passages dans la boucle **while** vérifie : $m=\sum_{k=1}^{n}(k-1)p_k$.

```
k:=1;
u:=random;
while *** do ***;
x:= ***;
write(x);
```

4. Loi binomiale

Dans cette question, on suppose que X suit la loi binomiale $\mathscr{B}(2n,\ 1/2)$.

a) Un premier programme

On adapte le programme précédent (question **II 3. b) ii)**) de simulation de X au cas de la loi binomiale $\mathscr{B}(2n,\ 1/2)$.

i) • Démontrer que, pour tous entiers naturels r et m tels que : $1\le r\le m$, on a :

$$\binom{m}{r}=\frac{m}{r}\binom{m-1}{r-1}.$$

• En déduire l'égalité : $\displaystyle\sum_{k=1}^{2n}k\binom{2n}{k}=n\,2^{2n}$.

ii) Exprimer le nombre moyen m_n de passages dans la boucle **while** en fonction de n.

b) Amélioration du programme

On pose dans cette question :

$$a_0=\binom{2n}{n},\ a_1=\binom{2n}{n-1},\ a_2=\binom{2n}{n+1},\ a_3=\binom{2n}{n-2},\ a_4=\binom{2n}{n+2},\dots,\ a_{2n-1}=\binom{2n}{0},\ a_{2n}=\binom{2n}{2n}.$$

i) • Démontrer que, pour tout entier k compris au sens large entre 0 et $n-1$: $\binom{2n}{k}\le\binom{2n}{k+1}$.

• En déduire les inégalités : $a_0\le a_1\le\ldots\le a_{2n}$.

ii) Pour améliorer le programme précédent, on range dans l'ordre décroissant les probabilités $p_0, p_1, \ldots, p_{2n}$ que X prennent les valeurs $0, 1, \ldots, 2n$ en posant :

$$p'_0 = p_n = \frac{1}{2^{2n}} a_0 ,\ p'_1 = p_{n-1} = \frac{1}{2^{2n}} a_1 ,\ p'_2 = p_{n+1} = \frac{1}{2^{2n}} a_2 , \ldots,\ p'_{2n} = p_{2n} = \frac{1}{2^{2n}} a_{2n} ,$$

les variables successives du tableau `F` contenant les nombres $\sum_{i=0}^{k} p'_i$ (l'entier k variant de 0 à $2n$), et les variables successives du tableau `T` contenant les valeurs n, $n-1$, $n+1$, $n-2$, $n+2, \ldots$, 0, $2n$.

- Préciser alors le nombre moyen m'_n de passages dans la boucle **`while`** en fonction de n ; pour cela, on admettra l'égalité : $\sum_{k=0}^{2n} k\, a_k = \left(2n + \frac{1}{2}\right) \times \binom{2n}{n} - 2^{2n-1}$.
- Enfin, proposer un équivalent simple de m'_n quand n tend vers $+\infty$ en utilisant la formule de Stirling :

$$n! \underset{+\infty}{\sim} \sqrt{2n\pi} \left(\frac{n}{\mathrm{e}}\right)^n ,$$

et conclure quant à l'intérêt du réagencement des probabilités $p_0, p_1, \ldots, p_{2n}$.

* * *