Exercice 1

Pour tout entier naturel n supérieur ou égal à 1,

$$p_{n+1} = 0{,}2\, p_n + 0{,}04 \text{ et } p_1 = 0$$

a. Montrer que la suite (u_n) définie pour tout entier naturel n supérieur ou égal à 1 par $u_n = p_n - 0{,}05$ est une suite géométrique dont on donnera le premier terme et la raison r. En déduire l'expression de u_n puis de p_n en fonction de n et r.

b. En déduire la limite de la suite (p_n).

c. On admet dans cette question que la suite (p_n) est croissante. On considère l'algorithme suivant :

Variables	K et J sont des entiers naturels, P est un nombre réel
Initialisation	P prend la valeur 0 J prend la valeur 1
Entrée	Saisir la valeur de K

Traitement	Tant que P < $0{,}05 - 10^{-K}$
	    P prend la valeur $0{,}2 \times P + 0{,}04$
	    J prend la valeur J + 1
	Fin tant que

Sortie	Afficher J

À quoi correspond l'affichage final J ?

Pourquoi est-on sûr que cet algorithme s'arrête ?

CORRECTION

a. $p_{n+1} = 0{,}2\, p_n + 0{,}04$ et $p_1 = 0$

donc $u_{n+1} = p_{n+1} - 0{,}05 = 0{,}2\, p_n + 0{,}04 - 0{,}05$

$u_{n+1} = 0{,}2\, p_n - 0{,}01$ or $u_n = p_n - 0{,}05$ donc $p_n = u_n + 0{,}05$ donc en remplaçant : $u_{n+1} = 0{,}2\,(u_n + 0{,}05) - 0{,}01$

$u_{n+1} = 0{,}2\, u_n$, la suite (u_n) est une suite géométrique de premier terme $u_1 = p_1 - 0{,}05 = -0{,}05$ et de raison $r = 0{,}2$

$u_n = r^{n-1}\, u_1 = -0{,}05 \times 0{,}2^{n-1}$

$p_n = u_n + 0{,}05$ donc $p_n = 0{,}05\,(1 - 0{,}2^{n-1})$.

b. $-1 < 0{,}2 < 1$ donc $\lim\limits_{n \to +\infty} 0{,}2^n = 0$ donc $\lim\limits_{n \to +\infty} p_n = 0{,}05$.

c.

Variables	K et J sont des entiers naturels, P est un nombre réel
Initialisation	P prend la valeur 0 J prend la valeur 1
Entrée	Saisir la valeur de K

Traitement	Tant que P < $0{,}05 - 10^{-K}$
	    P prend la valeur $0{,}2 \times P + 0{,}04$
	    J prend la valeur J + 1
	Fin tant que

Sortie	Afficher J

Pour comprendre l'algorithme faisons le fonctionner pour $k = 3$

P prend la valeur 0

J prend la valeur 1

K = 4 donc $0{,}05 - 10^{-K} = 0{,}049$

Etape 1

P prend la valeur $0{,}2 \times P + 0{,}04$ donc P prend la valeur $0{,}2 \times 0 + 0{,}04$ soit $p_2 = 0{,}04$

J prend la valeur J + 1 donc J = 2

Etape 2

P < $0{,}05 - 10^{-K}$ donc P < 0,049 donc l'algorithme se poursuit

P prend la valeur $0{,}2 \times P + 0{,}04$ donc P prend la valeur $p_3 = 0{,}048$

J prend la valeur J + 1 donc J = 3

Etape 3

P < $0{,}05 - 10^{-K}$ donc P < 0,049 donc l'algorithme se poursuit

P prend la valeur $0{,}2 \times P + 0{,}04$ donc P prend la valeur $p_4 = 0{,}0496$

J prend la valeur J + 1 donc J = 4

Etape 4

P > 0,049 donc P > $0{,}05 - 10^{-K}$ donc l'algorithme s'arrête.

On affiche J = 4

L'affichage final J correspond à la valeur minimale n_0 de l'indice tel que $p_n > 0{,}05 - 10^{-K}$.

La suite (p_n) est croissante et $\lim\limits_{n \to +\infty} p_n = 0{,}05$ donc pour tout $n \in \mathbb{N}^*$, $p_n \le 0{,}05$. D'après la définition de $\lim\limits_{n \to +\infty} p_n$, p_n est aussi proche qu'on le souhaite de 0,05 à partir d'un certain rang, ici on souhaite que $0{,}05 - 10^{-k} \le p \le 0{,}05$ donc cet algorithme s'arrête quelque soit le k choisi.

Exercice 2

On étudie le mouvement aléatoire d'une puce. Cette puce se déplace sur trois cases notées A, B, C.

A l'instant 0, la puce est en A. Pour tout entier naturel n :

- si à l'instant n la puce est en A, alors à l'instant $(n + 1)$, elle est :
  - soit en B avec une probabilité égale à $\frac{1}{3}$,
  - soit en C avec une probabilité égale à $\frac{2}{3}$,
- si à l'instant n la puce est en B, alors à l'instant $(n + 1)$, elle est :
  - soit en C, soit en A de façon équiprobable,
- si à l'instant n la puce est en C, alors elle *y* reste.

On note A (respectivement B, C) l'événement : « à l'instant n la puce est en A » (respectivement en B, en C).

On note a (respectivement b, c) la probabilité de l'événement A (respectivement B, C). On a donc: $a_0 = 1$, $b_0 = c_0 = 0$.

On code la case A par 0, la case B par 1 et la case C par 2.

On nomme pos la variable qui contient la position de la puce à un instant donné, c'est-à-dire la valeur 0, 1 ou 2, tirage une variable qui contiendra un tirage aléatoire d'un nombre entre 0 et 1.

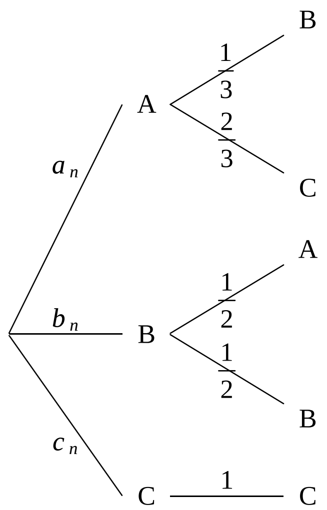
1. Compléter l'algorithme ci-dessous pour simuler le passage de l'instant donné à l'instant suivant (Nbaléatoire désigne un nombre aléatoire entre 0 et 1).

```
Affecter 2 à a.
Si pos = 0 Alors Affecter Nbaléatoire à tirage
|           Si tirage < 1/3 Alors Affecter … à a
|           |            Sinon …
|           FinSi
FinSi
Si pos = 1 Alors Affecter Nbaléatoire à tirage
|           Si tirage < 0,5 Alors Affecter … à a
|           |            Sinon…
|           FinSi
FinSi
Affecter a à pos
```

2. Compléter l'algorithme pour simuler une marche jusqu'à l'instant n, la valeur de n étant saisie par l'utilisateur ($n \in \mathbb{N}^*$), et afficher la case d'arrivée.

CORRECTION

En faisant un arbre de choix :



1. Il faut donc simuler un tirage permettant d'avoir une probabilité égale à $\frac{1}{3}$, pour cela on utilise que le tirage au sort d'un nombre réel dans l'intervalle [0 ; 1] a une probabilité $\frac{1}{3}$ d'être dans l'intervalle $\left[0;\frac{1}{3}\right]$.

et une probabilité $\frac{1}{2}$ d'être dans l'intervalle $\left[0;\frac{1}{2}\right]$.

Variables	pos est un entier naturel *a* est un entier naturel Nbaléatoire est un réel
Initialisation	Lire pos
Traitement	Nbaléatoire PREND_LA_VALEUR random() SI (pos=0) ALORS DEBUT_SI SI (Nbaléatoire<1/3) ALORS DEBUT_SI pos PREND_LA_VALEUR 1 FIN_SI SINON DEBUT_SINON pos PREND_LA_VALEUR 2 FIN_SINON FIN_SI SINON DEBUT_SINON SI (pos=1) ALORS DEBUT_SI SI (Nbaléatoire<0.5) ALORS DEBUT_SI pos PREND_LA_VALEUR 0 FIN_SI SINON DEBUT_SINON pos PREND_LA_VALEUR 2 FIN_SINON FIN_SI SINON DEBUT_SINON pos PREND_LA_VALEUR 2 FIN_SINON FIN_SINON
Sortie	AFFICHER pos

Cet algorithme permet donc en partant d'une position donnée (lire pos) d'aller soit en A, soit en B soit en C.

2.

Variables	pos est un entier naturel *a* est un entier naturel Nbaléatoire est un réel *n* EST_DU_TYPE NOMBRE *k* EST_DU_TYPE NOMBRE
Initialisation	pos PREND_LA_VALEUR 0 LIRE *n* *k* PREND_LA_VALEUR 0
Traitement	TANT_QUE (*k*<=*n*) FAIRE DEBUT_TANT_QUE Nbaléatoire PREND_LA_VALEUR random() SI (pos=0) ALORS DEBUT_SI SI (Nbaléatoire<1/3) ALORS DEBUT_SI pos PREND_LA_VALEUR 1 FIN_SI SINON DEBUT_SINON pos PREND_LA_VALEUR 2 FIN_SINON FIN_SI SINON DEBUT_SINON SI (pos=1) ALORS DEBUT_SI SI (Nbaléatoire<1/2) ALORS DEBUT_SI pos PREND_LA_VALEUR 0 FIN_SI SINON DEBUT_SINON pos PREND_LA_VALEUR 2 FIN_SINON FIN_SI SINON DEBUT_SINON pos PREND_LA_VALEUR 2 FIN_SINON FIN_SINON *k* PREND_LA_VALEUR *k*+1 FIN_TANT_QUE
Sortie	AFFICHER pos

Pour afficher la case, il aurait suffit de rajouter des lignes :
si pos = 0 afficher A
si pos = 1 afficher B
sinon afficher C

Exercice 4

1. Soit U une variable aléatoire suivant la loi uniforme sur [0 ; 1 [. λ est un réel strictement positif.

On considère la variable aléatoire $T = -\frac{1}{\lambda}\ln(1 - U)$.

a. Justifier que T est bien définie et prend ses valeurs dans [0 ; + ∞ [.

b. Montrer que, pour $t \geq 0$, $P(T \leq t) = P(U \leq 1 - e^{-\lambda t})$; en déduire l'expression de $P(T \geq t)$ en fonction de t.

c. On désigne par f la fonction de densité de la variable aléatoire T.

Que vaut, pour $t \geq 0$, $F(t) = \int_0^t f(x)\,dx$?

En déduire l'expression de $f(t)$ pour $t \geq 0$ et conclure.

2. Que simule sur un tableur l'instruction : = – 0,125*1n (1 – ALEA()) ?

Pourquoi peut-on aussi utiliser, plus simplement : = – 0,125 1n (ALEA()) ?

CORRECTION

1. *a*. $0 \leq U < 1$ donc $0 \geq -U > -1$ donc $1 \geq 1 - U > 0$

$1 - U > 0$ donc T est bien définie

$1 - U \leq 1$ donc $\ln(1 - U) \leq \ln 1$ soit $\ln(1 - U) \leq 0$ or $-\frac{1}{\lambda} < 0$ donc $T \geq 0$

T prend ses valeurs dans [0 ; + ∞ [.

1. *b*. $T \leq t \Leftrightarrow -\frac{1}{\lambda}\ln(1 - U) \leq t \Leftrightarrow \ln(1 - U) \geq -\lambda t \Leftrightarrow 1 - U \geq e^{-\lambda t} \Leftrightarrow 1 - e^{-\lambda t} \geq U$ donc $P(T \leq t) = P(U \leq 1 - e^{-\lambda t})$

U une variable aléatoire suivant la loi uniforme sur [0 ; 1 [donc $P(U \leq 1 - e^{-\lambda t}) = 1 - e^{-\lambda t}$ donc $P(T \leq t) = 1 - e^{-\lambda t}$.

T suit une loi exponentielle de paramètre λ.

c. f est la fonction de densité de la variable aléatoire T donc pour $t \geq 0$, $F(t) = \int_0^t f(x)\,dx = P(T \leq t) = 1 - e^{-\lambda t}$.

F est la primitive de f nulle en 0 donc $F'(t) = f(t) = -(-\lambda)\,e^{-\lambda t} = \lambda\,e^{-\lambda t}$.

T suit une loi exponentielle de paramètre λ.

2. U = ALEA() suit une loi uniforme sur [0 ; 1 [donc T = – 0,125*1n (1 – ALEA()) suit une loi exponentielle de paramètre λ.

$\lambda = \frac{1}{0{,}125} = 8$. T simule donc le tirage au hasard d'un nombre dans [0 ; + ∞ [.

$T_2 = -0{,}125 \ln U$ n'est pas définie pour U = 0

T_2 simule le tirage au hasard d'un nombre dans] 0 ; + ∞ [selon une loi exponentielle de paramètre 8.

$P(T_1 = 0) = P(U = 0) = 0$ on peut donc considérer que T_1 et T_2 simulent le tirage au hasard d'un nombre dans] 0 ; + ∞ [selon une loi exponentielle de paramètre 8.