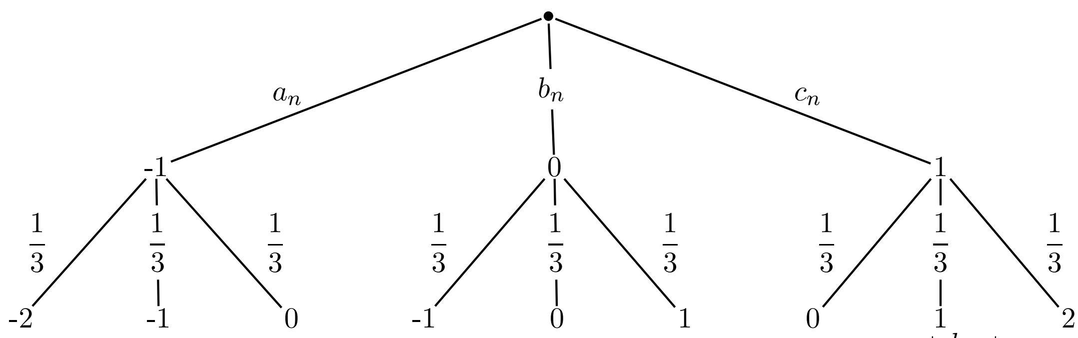
Partie A : modélisation et simulation

1. Dans l'algorithme, on part de $x = 0$ et la seule instruction qui change x est une incrémentation : on ne peut donc pas terminer avec $x = -1$.
   Le couple $\boxed{(10\,;\,0)}$ a pu être obtenu. Par exemple si les choix au hasard ont donné dix fois de suite 0, on incrémente 10 fois de suite x, sans bouger y qui reste à 0 ; puis, comme x vaut 10, on sort du tant que.
   L'algorithme ne change y à chaque itération que d'une unité (en plus ou en moins) au maximum, et s'arrête dès que $y \notin [-1; 1]$. Ainsi il est impossible d'afficher une valeur de y qui ne soit pas dans $[-2; 2]$: Tom se serait arrêté avant. On ne peut donc pas terminer avec $y = 4$.
   Le couple $\boxed{(10\,;\,2)}$ a pu être obtenu, par exemple si les choix au hasard ont donné huit fois 0, puis deux fois 1.
2. On pourrait être tenté d'écrire que Tom a réussi la traversée lorsque $x = 10$, mais l'exemple donné dans la question 1 nous montre que pas forcément. Il est donc plus judicieux de tester la valeur de y pour savoir si Tom a réussi ou pas. On peut donc écrire la condition suivante à la place de l'affichage initial :

```
Si y = 2 ou y = −2, alors
        Afficher « Tom est tombé »
Sinon
        Afficher « Tom a réussi la traversée »
Fin si
```

Partie B

1. Après 0 déplacements, donc tout au début de la traversée, on sait que Tom part du centre du pont, donc à $y = 0$. Ainsi $a_0 = 0, b_0 = 1, c_0 = 0.\square$
2. Comme l'indique l'énoncé, c'est une bonne idée de procéder par un arbre pondéré :



   Ainsi l'événement B_{n+1} se situe sur les branches 3, 5 et 7, d'où $b_{n+1} = \dfrac{a_n + b_n + c_n}{3}.\square$
   L'événement A_{n+1} se situe sur les branches 2 et 4, d'où $a_{n+1} = \dfrac{a_n + b_n}{3}.\square$
3. D'après les formules précédentes, $p(A_1) = a_1 = \dfrac{a_0 + b_0}{3} = \dfrac{0+1}{3} = \boxed{\dfrac{1}{3}}$.
   $p(B_1) = b_1 = \dfrac{a_0 + b_0 + c_0}{3} = \dfrac{0+1+0}{3} = \boxed{\dfrac{1}{3}}$.
   En lisant sur l'arbre de la question précédente, on voit également que l'événement C_{n+1} se situe sur les branches 6 et 8, d'où $c_{n+1} = \dfrac{b_n + c_n}{3}$. Ainsi $p(C_1) = c_1 = \dfrac{b_0 + c_0}{3} = \dfrac{1+0}{3} = \boxed{\dfrac{1}{3}}$.

4. Notons P_2 l'événement ≪ Tom se trouve sur le pont au bout de deux déplacements ≫. Tom se trouve sur le pont au bout de deux déplacements si son ordonnée est encore de -1, 0 ou 1 au bout de deux déplacements, donc $P_2 = A_2 \bigsqcup B_2 \bigsqcup C_2$.

   Donc $\mathrm{P}(P_2) = a_2 + b_2 + c_2 = \dfrac{a_1 + b_1}{3} + \dfrac{a_1 + b_1 + c_1}{3} + \dfrac{b_1 + c_1}{3} = \dfrac{\frac{7}{3}}{3} = \boxed{\dfrac{7}{9}}$

5. Notons T l'événement ≪ Tom traverse le pont ≫. Tom traverse le pont si son ordonnée est encore de -1, 0 ou 1 au bout de dix déplacements, donc $T = A_{10} \bigsqcup B_{10} \bigsqcup C_{10}$.

   Donc $\mathrm{P}(T) = a_{10} + b_{10} + c_{10} \approx 0,040272 + 0,056953 + 0,040272 \approx \boxed{0,137}$ à $0,001$ près.

   Effectivement les valeurs approchées du tableau sont à 10^{-6} près donc la marge d'erreur est de 3×10^{-6} qui est largement inférieure à 10^{-3} donc on peut utiliser ces valeurs approchées pour le calcul à 10^{-3} près (par contre on n'aurait pas pu les utiliser pour obtenir un résultat à 10^{-6} près !)

   Formellement, voici la meilleure approximation qu'on puisse obtenir à partir du tableau :

$$\begin{array}{rcl}
0,040271 \leq & a_{10} & \leq 0,040273 \\
0,056952 \leq & b_{10} & \leq 0,056954 \\
0,040271 \leq & c_{10} & \leq 0,040273 \\
\hline
0,137497 \leq & a_{10} + b_{10} + c_{10} & \leq 0,1375
\end{array}$$