

Initiation à la programmation

Durée : 60 min

Barème : / 20 points

Barème : 1 point par question, soit 4 points par exercice

Unités : réponse sans unité = 0,5 point retiré

1 Lecture d'un script

/ 4 pts

- a. $a = 8 - 3 = 5$; $b = 3$.
- b. $b = 5 \times 3 = 15$; $a = 5$; $15 > 12$ est vrai, le lutin dit 15.
- c. $a = 5 - 3 = 2$; $b = 2 \times 3 = 6$; $6 > 12$ est faux, le lutin dit 2.
- d. Si l'on calcule b d'abord, $b = 8 \times 3 = 24$ puis $a = 8 - 24 = -16$: chaque instruction utilise les valeurs courantes des variables.

Méthode — on suit un script ligne par ligne avec un tableau de suivi des variables, en n'oubliant pas qu'une variable modifiée l'est pour toutes les instructions suivantes.

2 Lecture d'un tableau de suivi

/ 4 pts

- a. On ajoute $n = 2$ à s ($s = 2$), puis n devient $2 + 3 = 5$.
- b. Tour 2 : $n = 8$, $s = 7$; tour 3 : $n = 11$, $s = 15$; tour 4 : $n = 14$, $s = 26$.
- c. Le lutin affiche 26 ; $n = 14$.
- d. Tour 5 : $s = 26 + 14 = 40$; tour 6 : $s = 40 + 17 = 57$; tour 7 : $s = 57 + 20 = 77$: 65 n'est jamais atteint, ce n'est pas possible avec ce script.

Le point clé — dans une boucle, l'ordre des deux instructions compte : on ajoute d'abord l'ancienne valeur de n , puis on modifie n ; le tableau de suivi rend cette chronologie visible.

3 Une figure tracée par le lutin

/ 4 pts

- a. Un carré de côté 40 : quatre côtés égaux et quatre angles de 90° .
- b. Trois carrés identiques ayant un sommet commun, tournés de 120° les uns par rapport aux autres.
- c. Longueur : $3 \times 4 \times 40 = 480$ pas ; angle : $3 \times (360 + 120) = 1440^\circ$.
- d. « répéter 5 fois : [répéter 3 fois : [avancer de 30 ; tourner de 120°] ; tourner de 72°] ».

Astuce — pour un polygone régulier à n côtés, l'angle de rotation vaut $360 \div n$; pour répartir k figures autour d'un point, on tourne de $360 \div k$ entre deux figures.

4 Vrai ou faux, avec justification

/ 4 pts

- a. Vrai : $6 \times 60 = 360^\circ$, un tour complet.
- b. Faux : ce n'est pas une équation mais une affectation ; la machine calcule $x + 1$ avec la valeur actuelle puis range le résultat dans x .
- c. Vrai : les blocs contenus dans le « si » sont ignorés, et le script continue après.
- d. Faux : avec deux entiers, le bloc renvoie un entier de 1 à 4.

Erreur fréquente — lire « mettre x à $x + 1$ » comme une égalité mathématique est l'erreur la plus courante ; en programmation, le signe « à » signifie « devient ».

5 Prise d'initiative — estimer π avec un lutin

/ 4 pts

- a. « mettre **c** à **0** ; répéter **2 000** fois : [mettre **x** à aléatoire **-100–100** ; mettre **y** à aléatoire **-100–100** ; si **x × x + y × y** **< 2 500** alors ajouter **1** à **c**] ; dire **c** ».
- b. Carré : **200**² = **40 000** ; disque : $\pi \times 50^2 = 2\,500\pi$; probabilité = $\frac{2\,500\pi}{40\,000} = \frac{\pi}{16} \approx 0,196$.
- c. **2 000** × $\frac{\pi}{16} \approx 393$ points.
- d. La fréquence $\frac{402}{2\,000} = 0,201$ approche $\frac{\pi}{16}$, donc $\pi \approx 16 \times 0,201 = 3,216$; l'estimation s'améliore avec plus d'essais.

Repère — une simulation relie programmation et probabilités : la fréquence observée sur un grand nombre d'essais approche la probabilité, ici un rapport d'aires qui fait apparaître π .