PENSÉE INFORMATIQUE · REPÈRES ET MÉTHODES

Instructions, entrées et sorties

ÉTAPE 1 · Exécuter dans le bon ordre

Prénom : ..... Date : .....

1 Suivre une séquence

Programme A : choisir un nombre ; ajouter 4 ; multiplier par 3 ; retirer 2.

Exécute-le pour 0 ; 2 ; 5 ; 1,5. Note chaque étape.

Présente les résultats dans un tableau entrée / étapes / sortie.

2 L’ordre compte

Programme B : choisir un nombre ; multiplier par 3 ; ajouter 4 ; retirer 2.

Teste les mêmes entrées et compare à A.

Explique pourquoi permuter deux instructions peut changer le résultat.

3 Retrouver l’entrée

Programme : multiplier par 2 puis ajouter 5.

Retrouve les entrées qui donnent 11 ; 25 ; 46.

Vérifie chacune en exécutant le programme.

Instructions, entrées et sorties

ÉTAPE 2 · Écrire et corriger

Prénom : ..... Date : .....

4 Produire des instructions

Écris un programme qui transforme n en 4 × n + 3, sans utiliser la lettre dans tes consignes.

Teste les entrées 1 ; 3 ; 10.

Écris un autre programme qui donne les mêmes sorties.

5 Repérer une erreur

But : calculer le double du nombre augmenté de 7. On veut 2 × (n + 7).

Un élève écrit : « doubler, puis ajouter 7 ». Teste avec n = 3.

Corrige les instructions et précise pourquoi la formulation doit être exacte.

6 Une trace d’exécution

Entrée 6. Étapes annoncées : ajouter 2 → 8 ; tripler → 24 ; diviser par 4 → 8.

Repère la première erreur et corrige la sortie.

Construis une autre trace avec une erreur à faire chercher à un camarade.

Instructions, entrées et sorties

ÉTAPE 3 · Concevoir un programme utile

Prénom : ..... Date : .....

7 Prix d’une commande

Chaque cahier coûte 2,50 € ; les frais fixes sont 4 €.

Écris les instructions pour calculer le total à partir du nombre de cahiers.

Teste 1 ; 4 ; 10 cahiers. Identifie l’entrée et la sortie.

8 Convertir une durée

Écris des instructions pour transformer un nombre entier de minutes en heures et minutes.

Exécute-les pour 95 ; 148 ; 245 min.

Explique le rôle du quotient entier et du reste dans la division par 60.

9 Vérifier un algorithme

Choisis l’un des programmes précédents et teste une entrée nulle.

Propose une entrée interdite ou sans sens dans le contexte.

Ajoute une consigne pour éviter cette entrée.