ALGORITHMIQUE · PARCOURS · MISE EN PRATIQUE

Vérifier et mettre en pratique

Mobilisez conditions, boucles et fonctions pour vérifier un résultat, justifier l'arrêt et construire trois mini-projets expliqués en Python et MATLAB.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Construire des cas de test et calculer leurs résultats attendus.
- Justifier l'arrêt et préciser l'opération comptée.
- Assembler conditions, boucles et fonctions dans trois projets vérifiables.

Prérequis. Avoir parcouru les fonctions ; la récursivité est utile pour certains prolongements.

Passer d’un exemple compris à un programme vérifiable

La mise en pratique assemble les notions du parcours. Un contrat décrit les entrées autorisées et le résultat recherché. Un cas de test associe une entrée à un résultat attendu. Une justification de terminaison explique pourquoi le déroulement s'arrête. Le chapitre consacré au coût précise ensuite quelle opération est comptée.

Ces questions restent distinctes. Un programme peut terminer mais calculer un mauvais résultat ; quelques tests réussis ne démontrent pas son comportement pour toutes les données possibles. Les exercices de vérification demandent donc de calculer des attentes indépendantes du code et de distinguer une erreur de programmation d'une entrée hors contrat.

Les chapitres de cette partie

- 28. Vérifier un résultat et traiter les cas limites

- 29. Expliquer pourquoi un algorithme termine
- 30. Construire de petits programmes complets

Construire trois projets avec des règles explicites

Le dernier chapitre propose un calcul de commande, une analyse de mesures et un traitement de commandes textuelles. Les prix sont exprimés en centimes ; une suite de mesures peut être vide ; une commande d'arrêt ne doit pas être traitée comme un ajout. Chaque décision découle d'une règle annoncée, et non d'une supposition sur une application réelle.

Avant d'ouvrir la correction, écrivez le résultat de deux entrées ordinaires et d'un cas limite. Tracez ensuite la méthode, exécutez le langage choisi et expliquez chaque écart. L'objectif n'est pas de reproduire trois fichiers à l'identique, mais de pouvoir modifier une règle en précisant les conséquences sur le contrat et les vérifications.

À retenir - fiche mémo

Tester

Associez des entrées choisies à des résultats calculés indépendamment.

Expliquer

Séparez le résultat correct, l’arrêt et le travail compté.

Assembler

Réutilisez les notions connues dans des projets aux règles explicites.

Pseudo-code

```
// example.pseudo
// review_checklist
// State the contract.
// Predict an ordinary case and a boundary case.
// Run, compare and explain.
```

Point de vigilance. Une interface plus complète ne peut pas être déduite des seuls scripts de démonstration.

Consulter la cheatsheet : Vérifications et projets · **Télécharger la fiche du thème en PDF**

Version web : https://www.clearcode.fr/courses/programming/algorithmic/practice