

# Guide d'évaluation pour la partie pratique de l'épreuve de la spécialité Numérique et sciences informatiques

Quatre compétences du programme sont évaluées directement par cette épreuve :

- **Programmation** : concevoir des solutions algorithmiques et traduire un algorithme dans un langage de programmation.
- **Autonomie** : faire preuve d'autonomie, d'initiative et de créativité.
- **Compréhension** : comprendre et réutiliser des codes sources existants, développer des processus de mise au point et de validation de programmes.
- **Oral** : présenter un problème ou sa solution, développer une argumentation dans le cadre d'un débat.

Chaque compétence compte pour la note finale selon un pourcentage donné dans le tableau ci-dessous :

Compétence	Pondération
<b>P</b>	35 %
<b>A</b>	20 %
<b>C</b>	25 %
<b>O</b>	20 %

Chaque compétence est évaluée de manière globale sur l'ensemble du sujet et les évaluateurs attribuent un niveau de maîtrise de la compétence parmi Insuffisant, Fragile, Suffisant et Maîtrisé selon la grille d'évaluation suivante.

Dans le cas d'une absence ou d'un refus de participation à l'épreuve, l'évaluateur doit cocher la case « Non traité » pour chaque compétence sur Santorin.

<b>Compétences</b>	<b>Insuffisant</b>	<b>Fragile</b>	<b>Suffisant</b>	<b>Maîtrisé</b>
<b>PROGRAMMATION</b>	Le code produit ne fonctionne pas ou est incohérent avec la consigne.	Le code fonctionne partiellement ou sur des cas simples uniquement. La logique est laborieuse, la syntaxe approximative.	Le code est fonctionnel et répond au cahier des charges. Respect des structures de base et des types.	Code fonctionnel, efficace, lisible (nommage pertinent) et robuste (gestion des cas limites).
<b>AUTONOMIE</b>	Élève passif qui attend la solution lors des appels. Ne teste pas son code ou ignore les erreurs affichées.	Sollicite de l'aide dès la première difficulté sans chercher. Tente des modifications au hasard.	Fait preuve de recherche avant d'appeler. Utilise les tests fournis pour valider ses étapes. Sait où chercher dans la documentation/le sujet.	Proactif. Anticipe les problèmes. Propose des pistes de résolution pertinentes avant même l'intervention de l'examineur.
<b>COMPRÉHENSION</b>	Incapable d'expliquer le rôle du code fourni. Ne voit pas le lien entre les problèmes observés et le code. N'arrive pas à utiliser le code fourni. Ne parvient pas à proposer de test adéquat pour valider le fonctionnement attendu.	Compréhension superficielle. Identifie les problèmes avec une aide forte, mais peine à proposer une correction logique. Peine à utiliser le code fourni. A besoin d'aide pour définir des tests pertinents pour valider le fonctionnement attendu.	Explique correctement le fonctionnement global. Identifie la source des problèmes et propose une correction valide. Est capable de réaliser des tests pertinents.	Analyse fine du code existant. Capable de justifier la pertinence d'une structure de données ou d'un algorithme. Tests exhaustifs.
<b>ORAL</b>	Communication quasi inexistante ou monosyllabique. Vocabulaire technique absent. Ne sait pas expliquer sa production.	Discours confus ou hésitant. Vocabulaire imprécis. Lit son écran sans recul.	S'exprime clairement lors des appels. Vocabulaire technique correct. Répond aux questions sur ses choix.	Discours fluide et structuré. Argumentation convaincante. Capable de critique constructive sur son propre travail ou le sujet.

