

Notes générales (à lire attentivement avant de répondre aux questions)

1. N'indiquez votre nom, prénom et école **que sur la première page**. Sur toutes les autres pages, vous ne pouvez écrire que dans les **cadres prévus** pour votre réponse. Si, suite à une rature, vous êtes amené à écrire hors d'un cadre, répondez obligatoirement sur la même feuille, sans quoi votre réponse ne pourra être corrigée.
2. Vous ne pouvez avoir que de quoi écrire avec vous; les calculatrices, GSM... sont **interdits**.
3. Vos réponses doivent être écrites **au stylo ou au bic** bleu ou noir. Pas de réponses laissées au crayon. Si vous désirez des feuilles de brouillon, demandez-en auprès d'un surveillant.
4. Vous **devez** répondre aux questions ouvertes en **pseudo-code** ou dans l'un des **langages de programmation autorisés** (Java, C, C++, Pascal, Python et PHP). Les erreurs de syntaxe ne sont pas prises en compte pour l'évaluation. Vous ne pouvez utiliser aucune fonction prédéfinie à l'exception de celles présentées à la page suivante.
5. Vous **ne pouvez** à aucun moment **communiquer** avec qui que ce soit, excepté avec les surveillants ou les organisateurs. Toute question portant sur la compréhension de la question ou liée à des problèmes techniques ne peut être posée qu'aux organisateurs. Toute question logistique peut être posée aux surveillants.
6. Les participants **ne peuvent en aucun cas quitter leur place** pendant l'épreuve, par exemple pour aller aux toilettes ou pour fumer une cigarette. Selon le lieu où vous présentez l'épreuve, il peut vous être interdit de manger ou boire durant cette dernière.
7. Vous avez **exactement une heure et quarante minutes** pour répondre à ce questionnaire. Un **aide-mémoire** sur le pseudo-code et les conventions prises se trouve à la page suivante.

Bonne chance !

Aide-mémoire du pseudo-code

Dans ce questionnaire, il y a des questions pour lesquelles vous devez compléter une partie de code. On vous demande de les compléter avec soit des *expressions*, soit des *instructions*. Les expressions ont une valeur, les instructions effectuent une action.

Les expressions possèdent une valeur et sont utilisées dans les calculs, pour modifier les valeurs des variable et dans les conditions. Certaines expressions ont comme valeur un nombre (entier ou réel) et d'autres un booléen (**true** ou **false**). Les instructions représentent des actions à exécuter. Un algorithme est une suite d'instructions.

Les expressions les plus basiques sont les entiers (comme 3, -12...) et les variables (comme x , sum ...). On peut également construire des expressions avec des opérateurs comme le montre le tableau suivant.

| opération | notation | exemples | |
|--|------------|--|--|
| égalité | = | 3 = 2 vaut false | 1 = 1 vaut true |
| différence | ≠ | 1 ≠ 4 vaut true | 3 ≠ 3 vaut false |
| comparaison | >, ≥, <, ≤ | 1 > 3 vaut false | 5 ≤ 5 vaut true |
| addition et soustraction | +, - | 3 + 2 vaut 5 | 1 - 9 vaut -8 |
| multiplication | × | 2 × 7 vaut 14 | -2 × 3 vaut -6 |
| division entière | / | 10/3 vaut 3 | 9/3 vaut 3 |
| reste de la division entière | % | 10%3 vaut 1 | 9%3 vaut 0 |
| « non » logique | not | not 3 = 2 vaut true | not 1 = 1 vaut false |
| « et » logique | and | 3 = 3 and 5 ≤ 9 vaut true | 1 ≥ 3 and 2 = 2 vaut false |
| « ou » logique | or | 3 = 3 or 12 ≤ 9 vaut true | 1 ≥ 3 or 2 ≠ 2 vaut false |
| accès à un élément de tableau | [] | soit tab valant [1, 2, 3], alors $tab[1]$ vaut 2 | |
| appel à une fonction retournant une valeur | size | ($list$) | min (12, 5) |

Les fonctions que vous pouvez utiliser sont, en plus de celles définies dans l'énoncé et celles définies sur les tableaux ci-dessous, les fonctions $\max(a, b)$, $\min(a, b)$ et $\text{pow}(a, b)$ qui permettent respectivement de calculer le maximum entre deux nombres, le minimum entre deux nombres et la puissance (a^b).

Le tableau suivant reprend les instructions de base :

| opération | notation | exemple |
|--|--------------------|---------------------------------|
| affectation | ← | $x \leftarrow 20 + 11 \times 2$ |
| renvoi | return | return 42 |
| appel à une fonction (qui ne renvoie rien) | nom de la fonction | sort ($list$) |

En plus, il y a également des *instructions de contrôle* qui sont au nombre de trois : **if-else**, **while** et **for**. Ces instructions vont faire en sorte d'exécuter un groupe d'instructions ou non, en fonction de conditions. La notation **for** ($i \leftarrow a$ **to** b **step** k) { [...] } indique une boucle qui se répète tant que $i \leq b$ pour i initialisé à a et incrémenté de k à la fin de chaque itération.

Dans un algorithme, outre les variables permettant de stocker une valeur, on peut également utiliser des *tableaux* pour stocker plusieurs valeurs. Un tableau tab de taille n est indicé de 0 à $n - 1$. La notation $tab[i]$ permet d'accéder au $(i - 1)^e$ élément du tableau. Ainsi, le premier élément du tableau est $tab[0]$. On peut créer un nouveau tableau tab de taille n , dont les éléments sont initialisés à 0, avec la notation $tab \leftarrow \text{newArray}(n)$. On peut faire une copie d'un tableau tab de taille n vers un nouveau tableau $newTab$ avec la notation $newTab \leftarrow tab$.

À propos du cout des opérations

Les questions XX vous demandent d'écrire des algorithmes efficaces afin d'obtenir le score maximum.

Lorsqu'on s'intéresse à la complexité temporelle d'un algorithme (une estimation du temps qu'il va prendre pour s'exécuter), une manière de procéder consiste à compter le nombre d'opérations élémentaires qu'il va effectuer. Toutes les instructions de base coutent une unité de temps. La création d'un tableau de taille n , et également la copie d'un tel tableau, coutent n unités de temps. Pour les instructions **if-else**, **while** et **for**, il faut compter une unité de temps pour le calcul de la condition. Puis, bien sûr compter les instructions qui seront exécutées.