POLYTECH – ELSE3-FISE 2025–2026

Contrôle de Info C

Durée : 1h Aucun document autorisé

Nom : Prénom :

Notez que les affirmations (antécédents, conséquents, rôles, et invariants) dans vos codes C entreront pour partie dans la note finale.

▶ 1. Combien de mode(s) transmission des paramètres existe(nt) en C ? Cochez une seule réponse !

- ☐ un seul, la transmission par référence.
- ☐ un seul, la transmission par valeur.
- ☐ deux, transmissions par valeur et par référence.
- ☐ zéro.

▶ 2. Dans l'appel de procédure `lire(x)`, `x` est

- ☐ un paramètre *donné* ;
- ☐ un paramètre *résultat* ;
- ☐ un paramètre *donné* et *résultat*.
- ☐ zéro.

▶ 3. Expliquez de façon claire et synthétique la différence entre les énoncés *tantque*, *répéter* et *pourtout*.

► 4. À partir de l'antécédent suivant :

```
// jour, mois et année : 3 variables de type int qui représentent une date valide
```

écrivez en C le fragment de code qui calcule la date de la veille. Vous pouvez utiliser la fonction `joursDansMois` qui renvoie le nombre de jours d'un mois d'une année donnée.

▶ 5. Écrivez en C une fonction `somme` qui lit sur l'entrée standard `n` entiers (**int**) et renvoie leur somme. Cette fonction possède un seul paramètre, le nombre `n` d'entiers > 0.

▶ 6. Écrivez la fonction booléenne `estPremier` qui teste si son paramètre `n` (un **int** $\geqslant 2$) est premier ou non. *Rappel* : un nombre premier admet uniquement deux diviseurs *distincts* 1 et lui-même. Pensez à minimiser le nombre d'itérations !

En C, un *entier non signé* peut être dénoté de façon décimale (une suite de chiffres de 0 à 9), de façon octale (une suite de chiffres de 0 à 7 préfixée par 0) ou de façon hexadécimale (une suite de chiffres de 0 à 9 et de lettres de A à F ou de a à f préfixée par $0x$ ou $0X$). Les notations suivantes sont valides : 8764 0765 $0xFF$ $0xFa3$ et correspondent aux entiers 8764 501 255 4003.

▶ 7. Écrivez la fonction `lireEntier` qui calcule et renvoie le prochain entier non signé lu sur l'entrée standard. Cet entier peut être précédé par des espaces. Vous pourrez utiliser les fonctions `isspace`, `isdigit` et `isxdigit` qui testent si leur paramètre (un caractère) est, respectivement, un espace, un chiffre décimal (`'0'-'9'`) et un chiffre hexadécimal (`'0'-'9'-'A'-'F'/'a'-'f'`). Utilisez la fonction `assert` pour traiter les cas d'erreur. Cette fonction `lireEntier` possède l'en-tête suivant :

```
/* Antécédent : le caractère courant de l'entrée standard est un espace ou un chiffre [0-9]
 * Rôle : renvoie l'entier non signé lu au format C
 */
int lireEntier(void) {
```