

## Contrôle continu n° 99 FAKE

Tout document, calculatrice, ordinateur ou moyen de communication est interdit.

Les exercices sont indépendants. Les codes demandés doivent être écrits en C.

Vous répondrez sur les feuilles placées à la fin de l'énoncé, en renseignant pour chacune d'elle vos prénom, nom et numéro d'étudiant.

**Durée de l'épreuve : 24h.**

---

### Exercice 1

- 1) 😎 On veut écrire une fonction `sqrt_int` prenant en paramètre un entier de type `int`, positif ou nul et renvoyant la partie entière de la racine carrée de cet entier.

- a) Dans quel cas on ne peut pas calculer cette valeur ?
- b) Écrire la partie commentaire que vous allez placer avant le prototype de la fonction. En particulier, donner l'assertion d'entrée et l'assertion de sortie.
- c) Si ce n'est pas le cas, réécrire l'assertion de sortie en utilisant le fait que la partie entière de la racine carrée de  $n$  est l'unique entier positif  $i$  tel que  $i^2 \leq n$  et  $(i + 1)^2 > n$ .
- d) Écrire le prototype de `sqrt_int`.

- 2) 🤔 Écrire la définition de la fonction `sqrt_int`.

**Indication :** Vous pourrez utiliser une variable `i` et écrire une boucle respectant l'invariant

IB :  $(i - 1) * (i - 1) < n$

- 3) 🔥 Écrire une fonction de prototype

```
bool scan_pos (int*p);
```

qui demande à l'utilisateur d'entrer un entier positif et renvoie celui-ci. Si l'utilisateur donne en entrée un entier strictement négatif, la fonction lui redemande jusqu'à ce que l'entier soit positif. En cas d'erreur de saisie (autre que le signe), la fonction renvoie `false`, sinon elle renvoie `true`.

- 4) 🤔 Donnez un corps à la fonction `main` afin de tester les deux fonctions que vous avez écrite.

## Exercice 2

Le calendrier grégorien est celui que nous utilisons depuis le 15 Octobre 1582.

Les mois de Janvier, Mars, Mai, Juillet, Aout, Octobre et Décembre comptent 31 jours.

Les mois d'Avril, Juin, Septembre et Novembre comptent 30 jours.

Le mois de Février compte 29 jours si l'année est bisextile et 28 sinon.

Une année est bisextile si elle est divisible par 4 mais pas par 100 ou si elle est divisible par 400.

Par exemple : L'année 2003 n'est pas une année bisextile car elle n'est pas divisible par 4.

L'année 1900 n'est pas une année bisextile car elle est divisible par 100. L'année 2004 est bisextile car elle est divisible par 4 mais pas par 100. L'année 2000 est bisextile car elle est divisible par 400.

Le but de cet exercice est d'écrire un programme permettant de tester si une date fournie par l'utilisateur sous la forme jj/mm/aaaa est valide dans le calendrier grégorien. Les mois sont représentés par des entiers Janvier = 1, Février = 2 etc.

- 1) 😬 Une date jj/mm/aaaa est valide si le jour (jj) est un entier positif, le mois (mm) est un entier positif inférieur ou égal à 12, la date se situe après le 15 octobre 1582 et qu'elle respecte les règles énoncées ci-dessus.

Décrire, à l'aide d'arbres la structure de tests permettant de décider si une date est valide. Il est conseillé d'utiliser plusieurs arbres successifs afin de simplifier la méthode.

- 2) 😊 En déduire une fonction de prototype  
`bool date_valide(int jj, int mm, int aaaa);`  
renvoyant true si la date est valide et false sinon. Vous vous efforcerez de suivre l'arbre de test que vous avez décrit dans la question précédente. Vous pourrez supprimer les mots clef else si ils vous semblent inutiles afin de ne pas alourdir l'indentation.
- 3) 🐢 Donnez une version de cette fonction qui n'utilise pas d'instruction if ni d'instruction switch. On s'efforcera de préserver la lisibilité du programme.