

# 1 Connaissances techniques et C++ (5 points)

Pour répondre à cette question, référez-vous au code fourni à l'Annexe A (page 8).

(a) Expliquez la différence entre «numérateur (n)» et «dedominateur=d» aux lignes 6 et 8 de `fraction.cpp`. (1 point)

Ligne 6 : il s'agit d'une **initialisation** de l'objet `numérateur` de type `int`. Cela équivaut à **construire** `numérateur` en lui copiant la valeur `n` ;

Ligne 8 : il s'agit d'une **affectation**. L'objet n'a pas été préalablement initialisé auparavant, car il n'y a pas d'initialisation pour les types de base.

(b) La ligne 17 de `prog1.cpp` est-elle un suspecte ? Si oui, expliquez pourquoi ? Sinon, écrivez simplement «Non». (1 point)

**Oui.** La ligne 17 accède à la position [5] de `tab` alors que ce tableau n'a qu'une taille de 5, c'est-à-dire des positions [0] à [4]. Le contenu de `tab[5]` étant indéterminé, on ne sait pas ce qui aura dans `f3` après la ligne 17.

(c) Ce programme se compile-t-il (sans erreur avec `g++ prog1.cpp fraction.cpp`) et s'exécute-t-il sans «plantage» ? Si oui, écrivez simplement «Oui». Sinon, (1) indiquez la nature du problème, (2) expliquez brièvement et (3) indiquez la(les) correction(s) minimale(s) et nécessaire(s). (1 point)

**Oui.**

(d) Qu'affiche ce programme ? Considérez les sauts de ligne et les modifications en (c) s'il y a lieu. (1 point)

```
F1/2 F2/3 F5/4 F0/1 F0/1 F0/1 F0/1 F0/1
K0/1 K0/1
K?/? K2/3 K1/2
```

où ? est un nombre indéterminé (mémoire précédente).

En pratique, on pourrait probablement observer `K0/0` au lieu de `K?/?`.

(e) Ce programme a au moins une fuite de mémoire. Évaluez la quantité de mémoire (en octets) qui n'a pas été libérée correctement à la fin du programme. (1 point)

1. Le tableau pointé par `tab` n'est jamais libéré, donc  $5 * \text{sizeof}(\text{Fraction})$ .

2. À chaque appel de la fonction `test`, `*c` n'est pas libéré, donc  $5 * \text{sizeof}(\text{Fraction})$ .

Donc,  $7 * \text{sizeof}(\text{Fraction}) = 7 * \text{sizeof}(\langle \text{int}, \text{int} \rangle) = 7 * 2 * 4 = 7 * 8 = 56$  octets.