

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)

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

(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)

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

(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)

Annexe A pour la Question 1

Cette page peut être détachée.

```

1  /* fraction.h */
2  class Fraction{
3  public:
4      Fraction(int n=0, int d=1);
5      ~Fraction();
6
7      Fraction& operator+=(const
8          Fraction& f);
9      Fraction operator+(const
10         Fraction& f) const;
11     bool operator<(const
12         Fraction& f) const;
13
14 private:
15     int numérateur,
16         dénominateur;
17     void simplifier();
18 };

```

```

1  /* fraction.cpp */
2  #include "fraction.h"
3  #include <iostream>
4  using namespace std;
5  Fraction::Fraction(int n, int d)
6      : numérateur(n)
7  {
8      dénominateur = d;
9      cout << "F" << n << "/" << d << " ";
10     simplifier();
11 }
12 Fraction::~Fraction(){
13     cout << "K"
14         << numérateur << "/"
15         << dénominateur << " ";
16     numérateur = dénominateur = 0;
17 }
18 void Fraction::simplifier(){/*...*/}

```

```

1  /* prog1.cpp */
2  #include <iostream>
3  #include "fraction.h"
4  void test(Fraction a, Fraction& b){
5      Fraction* c = new Fraction(a);
6      a = b;
7      b = *c;
8      c = NULL;
9      return;
10 }
11 int main(){
12     Fraction f1(1,2), f2(2,3), f3(5,4);
13     Fraction* tab = new Fraction[5];
14     std::cout << std::endl;
15     test(f1, tab[2]);
16     test(tab[3], tab[4]);
17     f3 = tab[5];
18     std::cout << std::endl;
19     return 0;
20 }

```