JUNIA Grande école d'ingénieurs

HEI · ISEN · ISA

TD4 / C++

Exercice 1 :

- Exécuter et commenter le programme C++ suivant :

```
#include <iostream>
#include <set>
using namespace std;

int main(){
        int a[] = {7, 4, 9, 1, 3, 4, 8, 2, 7, 5, 3, 6, 10, 4, 8, 10, 1, 2};
        multiset<int> s(&a[0], &a[17]);
        multiset<int>::iterator p = s.begin();
        while (p != s.end()) cout << *p++ << " ";
        return 0;
}
```

- Reprendre le même code avec un container de type set.

Exercice 2 :

Vérifier la priorité de résolution d'appel aux fonctions en exécutant le programme C++ suivant :

```
#include <iostream>

using namespace std;

void f(int)              { cout << "f (int)" << endl ;}
template<class T> void  f(T)   { cout << "f (template)" << endl;}
void f(float)                  { cout << "f (float)" << endl;}

int main(){
      f(1);
      f(1.f);
      f(1.);
      return 0;
}
```

Exercice 3 :

Prenez connaissance du code utilisant la classe fraction et des éléments des cours sur la surcharge des opérateurs ainsi que les templates et les STL.

Ce code permet, au delà des fonctions de base comme l'initialisation d'une fraction :

- la modification de ses numérateur et dénominateur,

Egalement, à l'aide de la surcharge des opérateurs, de :

- lire (« >> ») et écrire (« << ») des fractions ;
- de comparer une fraction avec une autre (inférieur avec « < » et égal avec « == »).

En utilisant l'implémentation de la classe « Fraction », le programme (dans le fichier « main.cpp ») permet de lire une liste de fractions à partir du fichier « fraction.txt », de les afficher dans l'ordre croissant, et de supprimer de cette liste les valeurs négatives.

A partir de ces éléments :

1) Compiler le programme et repérer les implémentations décrites ci-dessus. Comprendre leur fonctionnement (poser des questions au besoin).
2) Classer les valeurs des fractions par ordre décroissant.
3) Ajouter dans le programme la possibilité d'afficher la somme totale des fractions et les valeurs minimale et maximale.