



TD #3

Rédacteur: Stéphane Vialette

Exercice 1

Écrire une classe template `MaxPlus` paramétrée par un type `T`. Un objet de type `MaxPlus<T>` contient une valeur numérique de type `T` et la classe contient

- un constructeur qui prend une valeur initiale en argument, mais qui ne permet pas le transtypage implicite ;
- une surcharge de l'opérateur `+=` qui remplace la valeur courante par le membre droit s'il est plus grand ;
- une surcharge de l'opérateur `*=` qui incrémente la valeur courante de la valeur du membre droit ;
- une surcharge de l'opérateur `+` qui retourne le maximum de deux valeurs ;
- une surcharge de l'opérateur `*` qui retourne la somme de deux valeurs ;
- une surcharge de l'opérateur de transtypage en une valeur de type `T`.

Par ailleurs, on désire que les objets de type `MaxPlus<T>` se comportent correctement vis-à-vis des flots.

On veut maintenant que le type `MaxPlus<T>` fournisse une méthode statique `get_zero()` retournant un objet de valeur $-\infty$. Cette valeur sera en fait représentée par la plus petite valeur représentable dans le type `T`. On utilisera pour cela la méthode statique `min()` définie dans la classe template `numeric_limits` de la stl (header `limits`).

Exercice 2

On rappelle que $binomial(n, k)$ est calculable récursivement selon les formules suivantes :

- $binomial(n, k) = 0$ si $k > n$,
- $binomial(n, k) = 1$ si $k = 0$ ou $k = n$,
- $binomial(n, k) = binomial(n - 1, k) + binomial(n - 1, k - 1)$ sinon.

Écrire la classe `Binomial` de sorte que `Binomial<N, K>::val` est une constante égale à $binomial(n, k)$.

Écrire un programme `binomial.cc` de sorte que la ligne `g++ -DN=n binomial.cc` produise un programme qui affiche $binomial(n, k)$ pour k compris entre 0 et n .

Exercice 3

Écrire une fonction paramétrée `instance_of` retournant un booléen, dont l'argument est une référence constante de type quelconque et telle que `instance_of<T>(x)` retourne `true` si et seulement l'objet `x` est d'un type compatible avec `T` (pour autant que les mécanismes du C++ permettent d'en décider).