C++ (M2)

TD #2

Rédacteur: Stéphane Vialette

Exercice 1

Écrire une classe `bit_reference` permettant de manipuler un bit donné en mémoire. Cette classe possède un constructeur prenant une référence de `unsigned char` ainsi qu'un entier `i` entre 0 et 7. L'objet ainsi construit permet de manipuler le i-ème bit de l'octet passé comme premier argument du constructeur. Pour cela, cette classe est dotée de :

- une surcharge de l'opérateur d'affectation qui permet d'affecter une valeur booléenne au bit ;
- une surcharge de l'opérateur de transtypage qui permet de convertir ce bit en valeur booléenne ;
- une méthode `swap()` qui permet de changer la valeur du bit.

Exercice 2

Écrire une classe `bool_vector_t` permettant de manipuler des vecteurs de booléens. Cette classe comporte :

- un constructeur dont l'argument de type unsigned int indique la taille initiale du vecteur. On utilisera **correctement** la fonction `calloc` de la bibliothèque standard pour allouer l'espace requis. Pour gagner de la place, on stockera 8 valeurs par octets (on pourra utiliser l'exercice précédent) ;
- une surcharge de l'opérateur `[]` permettant d'accéder en lecture et en écriture aux éléments. Cet opérateur produira une exception `out_of_range` (bibliothèque stdexcept) en cas d'indice incorrect ;
- des méthodes `size()` et `capacity()` indiquant respectivement le nombre d'éléments et la place réservée pour le vecteur ;
- une méthode `reserve()` permettant de réserver de l'espace supplémentaire (penser à `realloc` ; la taille est exprimée en nombre d'éléments, donc de bits, mais la taille réellement allouée sera un nombre entier d'octets), et une méthode `resize()` permettant d'augmenter ou diminuer le nombre d'éléments (avec allocation si besoin) ;

– un type `iterator` et des méthodes `begin()` et `end()` retournant un iterator utilisable comme suit :

```
bool_vector_t v(12);
for(bool_vector_t::iterator it = v.begin(); it != v.end(); ++it)
{
  *it = true;
}
```