

Laboratoire 13: Matrices binaires en C++

Durée du laboratoire: 4 périodes. A rendre le jeudi 10 mars 2016, au début de la séance de laboratoire.

Environnement C++

Installer CodeBlocks, Netbeans, Eclipse, Visual C++ ou GNU g++.

Matrices binaires

Définir une classe permettant de représenter des matrices binaires carrées de taille quelconque (NxN), telle que:

- A sa création (une fois sa taille définie) le contenu d'une matrice soit généré aléatoirement.
- Il soit possible d'afficher le contenu d'une matrice.
- Il soit possible d'appliquer les opérateurs logiques `ou`, `et` et `ou exclusif` entre deux matrices de même taille de deux manières différentes:
 - Soit en modifiant la matrice sur laquelle est invoquée la méthode,
 - Soit en retournant, par valeur (pourquoi par valeur et non par référence?) une nouvelle matrice résultat allouée statiquement,
 - Soit en retournant, un pointeur sur une nouvelle matrice résultat allouée dynamiquement.

Remarques

- Représenter une matrice binaire au moyen d'un tableau de tableaux de booléens.
- Prendre garde à ce que toute la mémoire allouée dynamiquement soit libérée correctement.
- Prendre garde à ce que la construction d'une matrice résultat s'effectue correctement.
- Pour générer aléatoirement un nombre utiliser le générateur aléatoire défini dans l'en-tête `<stdlib.h>`
 - initialisation du générateur aléatoire: `srand(time(0))` (où `time()` est défini dans l'en-tête `<time.h>`)
 - génération d'un nombre aléatoire entre 1 et n: `1 + rand() / (RAND_MAX + 1.0) * n`
- Factoriser le code commun aux différentes opérations logiques au moyen de fonctions représentant l'opération à effectuer sur les éléments des matrices opérands et ceci sans utiliser de structures de contrôle.

Exemple d'exécution pour l'opérateur `ou exclusif` modifiant la matrice sur laquelle est invoquée la méthode:

```
one:
1 1 1 0
0 1 1 1
1 1 0 0
1 1 0 0

two:
1 0 1 1
1 1 0 1
0 1 0 1
1 1 1 0

one, apres one xor two:
0 1 0 1
1 0 1 0
1 0 0 1
0 0 1 0
```