

TP5 – C++ - Vecteurs à 2 dimensions

Facebook... les amis... (2 séances)

NB : Les codes sources utiles pour ce TP sont disponibles sur Moodle

Objectifs

1. Créer et remplir des vecteurs à deux dimensions
2. Parcourir et lire le contenu de vecteurs 2D
3. Utiliser des booléens
4. Manipuler différents types de boucles

Consignes générales pour les exercices sur machine

1. Lire **entièrement** les questions avant tout.
2. Préparer un jeu d'essai avant d'écrire du code.
3. Pour écrire vos programmes, regardez ce qui existe déjà.
4. Tester en utilisant le jeu d'essai. Vos tests doivent être en commentaire à la fin de vos fichiers.

TRAVAIL à RENDRE : Déposer sur Moodle, votre travail accompagné de vos jeux d'essais.

Dans cet exercice, nous allons simuler le réseau Facebook par un vecteur `amis` à 2 dimensions (ce qui est bien loin de la réalité !). Les lignes et les colonnes représentent des personnes inscrites sur le réseau, chaque `case[i][j]` contient un booléen indiquant si `i` et `j` sont des amis (`true` si `i` et `j` sont des amis, `false` sinon). Les personnes sont représentées par des numéros (en commençant par 0).

Exemple d'affichage du réseau ci-contre

Nombre de personnes : 4

La personne 0 est amie avec : 0 2 3

La personne 1 est amie avec : 1 2

La personne 2 est amie avec : 0 1 2 3

La personne 3 est amie avec : 0 2 3

true	false	true	true
false	true	true	false
true	true	true	true
true	false	true	true

Question 1 Recopier le programme **facebook-TP5.cpp** sur Moodle et compléter le pour initialiser le réseau `amis`, avec 4 personnes, de la façon suivante : chaque personne n'est amie qu'avec elle-même.

Question 2 Afficher le contenu du réseau sous la forme présentée ci-dessus (une ligne par personne contenant : "la personne X est amie avec : Z X Y").

Question 3 (optionnelle) Modifier pour n'afficher que les amis autres que soi-même.
ATTENTION : préférez faire 2 boucles successives plutôt qu'une boucle avec un test sur l'indice...

- Question 4** Compléter le programme pour permettre ensuite de mettre à jour le vecteur **Amis** afin de mémoriser que 2 personnes deviennent amies. Les numéros des personnes sont donnés par l'utilisateur. Combien de cases doivent être mises à jour ? Vérifier la validité des numéros par une boucle de saisie. Vérifier le résultat en affichant.
Ajouter plusieurs couples amis et vérifier (vous pouvez faire une boucle pour la saisie : « voulez vous continuer o/n ? »).
- Question 5** Compléter le programme pour qu'il affiche le numéro de la personne qui a le plus d'amis (dans l'exemple, c'est la personne 2). En cas d'égalité, afficher le numéro de la 1^{ère} personne.
- Question 6 (optionnelle)** Compléter le programme pour qu'il affiche le nombre d'amis communs entre deux personnes dont les numéros seront donnés par l'utilisateur. Vérifier la validité des numéros. Dans le réseau donné en exemple, le nombre d'amis communs entre 1 et 3 est 3 (car les amis communs entre 1 et 3 sont 1 2 et 3).
- Question 7 (optionnelle)** Le vecteur des amis n'est correct que s'il est symétrique (si *i* est ami avec *j* cela veut dire que *j* est ami avec *i*). Modifier le programme pour qu'il vérifie puis affiche si le vecteur donné est correctement constitué.
- Question 8 (optionnelle)** On veut pouvoir désigner les personnes par leurs noms. Complétez le programme en ajoutant un vecteur `noms` qui contient le nom de toutes les personnes (de même taille que le vecteur `amis`). L'association entre nom et numéro de personne se fera par l'intermédiaire des indices (le numéro d'une personne correspond à son indice dans le tableau `noms`). Modifiez les instructions d'affichage de la question 1 (et des autres questions si vous avez le temps) pour afficher les noms des personnes au lieu de leurs numéros.