Travaux dirigés C++ n°7

Informatique

—IMAC 2e année—

La STL - partie 1

Au cours de ce TP vous apprendrez à utiliser la Standard Template Library. Il est conseillé de se référer à la documentation en ligne : http://cplusplus.com/reference/ et notamment http://cplusplus.com/reference/stl/.

▸ **Exercice 1.** *Le conteneur* ***vector***

Objectifs : La construction et l'affichage des propriétés élémentaires d'un vecteur.

Outils : empty(), size(), max_size(), push_back().

1. *déclarer d'un vecteur encore vide contenant des entiers*
2. *afficher si le vecteur est vide en testant avec empty()*
3. *afficher la taille du vecteur*
4. *afficher la taille maximale d'un vecteur d'entiers*
5. *introduire quelques éléments dans le vecteur*
6. *afficher la nouvelle taille*
7. *afficher le contenu du vecteur*

▸ **Exercice 2.** ***vector et swap()***

La fonction swap() permet d'échanger les contenus de deux vecteurs. Utiliser cette fonction pour échanger deux vecteurs v1 et v2 de type std :: vector < std :: string > contenant "Test" et "Swap" respectivement.

▸ **Exercice 3.** ***Algorithmes: sort, count***

La bibliothèque standard ne se réduit pas à la définition de conteneurs. Elle définit également un jeu de fonctions appelées algorithmes. Pour ces algorithmes, l'accès aux données contenues dans un conteneur s'effectue exlusivement par le biais d'itérateurs.

Ecrire un code qui permet de :

1. *définir un vecteur v de 20 entiers*
2. *remplir le vecteur v de manière aléatoire au moyen de la fonction rand() avec des valeurs entre 0 et 20*
3. *afficher ce vecteur*
4. *le trier*
5. *l'afficher à nouveau*
6. *compter le nombre de fois la valeur 7 existe dans ce vecteur*

▸ **Exercice 4.** ***deque***
Objectifs : comprendre le fonctionnement du conteneur deque.
Outils : utiliser push_front(), pop_front(), push_back(),front_back().

Le conteneur deque (doubly ended queue) est assez similaire au conteneur vector, à ceci près que sa finalité de donner un rôle équivalent au début et à la fin de la suite d'éléments qu'il contient.

1. *définir une file à double extrémité (la deque) de 5 éléments,*
2. *initialiser la deque par des nombres au hazard (fonction rand())*
3. *afficher la deque*
4. *on y fait entrer successivement 5 nombres au hazard placés en tête, le dernier élément de la file étant chaque fois retiré. la deque est affichée après chaque opération.*

▸ **Exercice 5.** ***Le conteneur list***
Les listes sont implémentées comme des listes chaînées, cela permet des insertions rapides au début et à la fin de la liste. Grâce aux itérateurs, des éléments peuvent être insérés au milieu de listes.
Une liste prend en charge un certain nombre d'opérations :

- *merge() : Fusionner les listes*
- *reverse() : Inverser l'ordre des éléments*
- *unique() : supprimer les doublons d'une liste triée*

1. *définir une liste l_philo de philosophes : Platon, Aristote, Descartes, et Kant*
2. *définir une deuxième liste l_math de mathématiciens : Gauss, Laplace, Poincaré, Descartes*
3. *afficher les deux listes triées (sort()) et les afficher*

4. *fusionner les deux listes et stocker le résultat dans une nouvelle liste*
5. *supprimer les répétitions dans la liste l_all*
6. *inverser l'ordre de la liste l_all puis l'afficher*