

PRÀCTIQUES D'ESTRUCTURES DE DADES

Robert Benavente i Jordi González
Març 1999

Pràctica 2: Utilització dels algorismes de la STL

1.- Descripció general de la pràctica

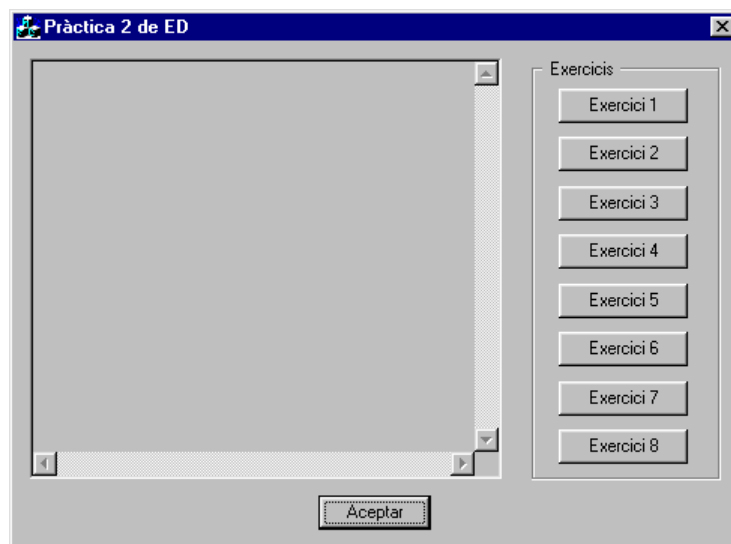
En les pròximes dues sessions s'aplicaran una sèrie d'algorismes de transformació i manipulació de seqüències de la STL, sobre l'estructura del vector genèric que es va crear en les sessions anteriors.

Haureu de realitzar cada una de les operacions que s'enumeren a continuació, mostrant en cada pas un text indicant quin pas heu fet i el vector resultant obtingut.

Abans d'aquesta sessió, és imprescindible haver-se llegit la descripció del funcionament dels algorismes STL del manual del *Stepanov*.

2.1. Què se us dona fet

El programa que se us passa en aquesta sessió és una aplicació basada en diàleg, que consisteix en 8 botons i un control CEdit (on escriureu els resultats). Hi han dos fitxers .cpp : **Practica2** (que inicialitza el diàleg i el destrueix) i **Practica2Dlg** (que és on posarem tot allò relatiu amb la interacció amb el diàleg, en el nostre cas pitjar buttons i escriure al control CEdit)



Per a utilitzar la llibreria STL, heu d'insertar al capdamunt del fitxer **Practica2Dlg.cpp** les següents instruccions:

```
#include <vector>
#include <algorithm>
#include <numeric>
using namespace std;
```

El primer és per a utilitzar totes les funcions relacionades amb la classe Vector que la STL incorpora, el segon per a utilitzar els algorismes de la STL i el tercer serveix per a que funcionin totes les crides a funcions STL en l'entorn Visual.

Dins del codi corresponent a la funció que es crida quan es pitja cada botó heu d'afegir el codi que realitza els exercicis que es descriuen al següent apartat.

2.2.- Què s'ha de fer

Cada botó té associat un event (codi que s'executa quan s'apreta cada botó):

```
void CPractica2Dlg::OnExercici1()
{
    // TODO: Add your control notification handler code here
    Escriure("Exercici 1");
}
```

La funció **Escriure**(" ") escriu un CString al CEdit del diàleg. Per tant, haureu d'utilitzar una variable CString (on posareu la informació) per presentar-ho per pantalla. Així, hi ha 8 tasques a fer:

- a) Per poder fer servir els algorismes STL sobre la nostra classe vector, primer haurem de definir-ne una funció membre **begin()** i **end()**, la primera ens retornarà un apuntador al primer element del vector, mentre que la funció **end()** ens retornarà un apuntador a una posició després de l'últim element del vector.
- b) **Exercici 1:** Genereu un vector de 10 sencers i feu servir la funció *generate* per posar-hi números aleatoris entre 0 i 100. Per fer-ho haureu de definir una funció que us generi un número aleatori i passar-se-la com a paràmetre a la funció *generate* (per incloure la funció *rand*, haureu d'incloure la llibreria **stdlib.h**).
- c) **Exercici 2:** Indiqueu quin és el valor màxim i mínim (*max_element*, *min_element*), i quan val la suma de tots els elements del vector (*accumulate*).
- d) **Exercici 3:** Substituiu els elements senars del vector per zeros (*replace_if*).
- e) **Exercici 4:** Conteus quants elements del vector són 0 (*count_if*).
- f) **Exercici 5:** Genereu un nou vector que només contingui els elements parells del vector inicial (*remove_copy*).
- g) **Exercici 6:** Definiu un array de 10 sencers amb els valors {1,2,3,4,5,6,7,8,9,0} i copieu-lo sobre un vector (*copy*). Resteu aquest vector al vector aleatori que havieu generat en el pas 2 (*transform*). Per fer-ho haureu de definir una funció que donats dos elements els resti. A més del vector resultat de la resta torneu el mateix vector ordenat de menor a major (*sort*).
- h) **Exercici 7:** Generar la sèrie de Fibonacci. Aquesta té la següent forma: {1,1,2,3,5,8,13,21,34,55}, és a dir:
 - h.1) Haureu de crear una classe *Fibonacci* amb dos atributs **v1** i **v2**
 - h.2) El constructor inicialitzarà **v1** a **0** i **v2** a **1**.

h.3) Sobrecarregar l'operador () per a quan utilitzeu el *generate* generi la seqüència de Fibonacci:

```
int r = v1 + v2;
v1 = v2;
v2 = r;
return v1;
```

h.4) Ara cal crear un vector de per exemple 10, i cridar a la funció *generate* passant-li com a tercer paràmetre un element de la classe Fibonacci (que serà el generador dels números).

- i) **Exercici 8:** Donats dos vectors, heu de mostrar per pantalla els elements comuns (*set_intersection*) i després els diferents (*set_difference*) que hi ha entre ells.

Heu d'implementar els exercicis dins del tros de codi associat a cada botó.

2.3.- Què heu de mostrar

Haureu de treure pel Dialog Box els resultats de les operacions anteriors amb un format similar al següent:

```
Vector aleatori generat: { 95 74 26 88 86 5 8 90 71 3 }
Valor Maxim = 95
Valor Mínim = 3
Suma Total = 546

Vector amb senars substituïts per zero: { 0 74 26 88 86 0 8 90 0 0 }
Elements Zero = 4

Vector Només Parells : { 74 26 88 86 8 90 }

Vector predefinit : { 1 2 3 4 5 6 7 8 9 0 }
Vector inicial: { 95 74 26 88 86 5 8 90 71 3 }
Vector resta: { 94 72 23 84 81 -1 1 82 62 3 }
Vector resta ordenat: { -1 1 3 23 62 72 81 82 84 94 }

Fibonacci: { 1 1 2 3 4 8 13 21 34 55 }

Vector 1: { 0 1 2 3 4 5 7 99 }
Vector 2: { -2 1 2 7 33 }

Intersecció: { 1 2 7 }
Diferència: { 0 3 4 5 99 }
```