Std::vector

Obtener iteradores:

Vector<int>::iterator itbeg = v.begin();

Vector<int>::iterator itend = v.end(); //itend esta despues del ultimo elemento

Vector<int>::iterator it5 = v.begin()+5;

Modificar el vector

V[0] = 25; //guarda “25” en la posición 0

\*it = 18 //”18” en la posicion apuntada por it

v.push\_back(10); //insertar “10” al final

it = v.insert(it,20) //insertar “20” en la posición it

v.pop\_back(); //eliminar el ultimo;

it = v.erase(it); //eliminar el de la posición it

v.erase(it1, it2); //eliminar rango [it1, it2)

v.clear(); //borrar todo

Cargar datos en (o recorrer) un vector

int main(){

vector<int> v(15,0);

for(vector<int>::iterator it = v.begin();it!=v.end(); ++it) { // != o <

\*it = 1+rand()%100;

}

vector<int>v(15,0)

for(auto it = v.begin(); it != v.end(); ++it)

auto x = 0; //deduce de acuerdo al tipo de elemento al que le asigne

//.-------------------------------------------------------------

vector<int>v(15,0);

for( int &x : v){ //"Para cada elemento del vector"

x = 1 + rand()%100;

}

//es igual a . . .

for( vector<int>::iterator it=v.begin(); it!=v.end(); ++it) {

int x = \*it;

cout << x<< endl;

}

//.-------------------------------------------------------------

Vector vs list (comparacion teorica)

Clase list, se organiza en memoria de forma no contigua, cada elemento tiene, además , un puntero que apunta al siguiente elemento. En cambio, en la clase vector los datos se almacenan de forma contigua.

Tipo de acceso:

en vector: aleatorio en list: secuencial

inserción y eliminación:

en vector : lenta en list: rápida.

List tiene una interfaz muy similar a la de vector, pero no permite acceso aleatorio:

* No tiene sobrecarga para el operador []
* Sus iteradores solo pueden “moverse” con ++ y --.

Movimiento de iteradores

Una posición modificando el iterador:

++it it++ --it it—

Una pos, sin modificar el iterador:

It1 = prev(it); it2 = next(it);

Multiples pos, modificando el iterador:

Advance(it,N); advance(it,-N);

Multiples pos, sin modificar el iterador:

It1 = it-N; it2 = it+N;

It = prev(it, N);

Biblioteca <algorithm>

Ejemplos de funciones de la biblioteca:

* [Agregar valores a un vector y a una lista (incompleto).cpp](Agregar%20valores%20a%20un%20vector%20y%20a%20una%20lista%20(incompleto).cpp)
* [Cambiar los -1 por el promedio entre el anterior y el siguiente.cpp](Cambiar%20los%20-1%20por%20el%20promedio%20entre%20el%20anterior%20y%20el%20siguiente.cpp)
* [Función insert.cpp](Función%20insert.cpp)
* [Funciones ya definidas para arreglos.cpp](Funciones%20ya%20definidas%20para%20arreglos.cpp)
* [generar valores.cpp](generar%20valores.cpp)
* [Mezclar valores de un vector.cpp](Mezclar%20valores%20de%20un%20vector.cpp)