Ejercicios Resueltos

Ejercicio 8.1

Escriba un programa que permita al usuario ingresar 15 valores por teclado, los almacene en un vector y luego:

a. Muestre el vector generado utilizando 3 mecanismos de iteración diferentes.

1. utilizando el operador []
2. utilizando iteradores
3. utilizando el for basado en rangos
4. ¿en qué caso es conveniente utilizar cada mecanismo?

b. Calcule y muestre

1. los valores de los elementos máximo y mínimo
2. la suma de todos los elementos del arreglo
3. el promedio y la mediana de los elementos del arreglo

c. Permita al usuario ingresar un valor, e informe si se encuentra en el vector,y en caso afirmativo, en qué posición.

```
//© Programación Orientada a Objetos- F.I.C.H. - U.N.L.
//Sitio web: http://e-fich.unl.edu.ar
//Comisión Viernes: Prof. Gerardo Sas.

…
int main() {
	//Decaro un vector dinamico de 15 posiciones
	vector<int> v(15);
	//Cargo desde teclado sus valores enteros
	cout<<"Ingrese 15 números enteros:"<<endl;
	for(size_t i=0;i<v.size();i++)
		cin>> v[i];
	//***
	cout<<"a.1 utilizando el operador []"<<endl;
	for(size_t i=0;i<v.size();i++)
		cout<< v[i] << "  ";
	cout<< endl;
	//***
	cout<<"a.2 utilizando iteradores"<<endl;
	for( vector<int>::iterator it=v.begin(); it!=v.end(); ++it )
		cout<< *it << "  ";
	cout<< endl;
	//***
	cout<<"a.3 utilizando el for basado en rangos"<<endl;
	for( int x : v )
		cout<< x << "  ";
	cout<< endl;
	//***
	cout<< "b.1 Max: " << *max_element(v.begin(),v.end()) << endl;
	cout<< "b.1 Min: " << *min_element(v.begin(),v.end()) << endl;
	//***
	int sum = accumulate(v.begin(),v.end(),0);
	cout<< "b.2 Suma: " << sum << endl;
	//***
	cout<< "b.3 Media: " << float(sum)/v.size() << endl;
```

```
	cout<< "b.3 Mediana: " << v[v.size()/2] << endl;
	//***
	cout<<"Ingrese un numero que este en el vector: ";
	int x;
	cin>> x;
	/*auto: el compilador determina el tipo que sustituirá a la palabra
	clave auto como si estuviera utilizando las reglas para la deducción
	de argumento de plantilla a partir de una llamada de función.*/
	auto it = find(v.begin(),v.end(),x);
	if (it==v.end())
		cout<< "No esta" << endl;
	else
		cout<< "Esta en la posicion " << it-v.begin() << endl;

	return 0;
}
```

Ejercicio 8.2

Escriba un programa que defina un vector dinámico de 30 enteros aleatoriosmenores que 25. Luego:
a. Ordene en forma descendente los elementos ubicados entre las posiciones10 y 20 inclusive, y muestre el vector.
b. Inserte en un nuevo vector los elementos que fueron ordenados en elapartado anterior, y quítelos del vector original.
c. Permita ingresar un valor por teclado, y muestre cuántas veces aparecedicho valor en el vector.
d. Elimine todas las ocurrencias del valor ingresado en el punto c, utilizando lafunción remove. Responda: ¿Pueden las funciones globales de la STLeliminar realmente los elementos (liberando la memoria de un contenedor)?(NO)

```
//© Programación Orientada a Objetos- F.I.C.H. - U.N.L.
//Sitio web: http://e-fich.unl.edu.ar
//Comisión Viernes: Prof. Gerardo Sas.
using namespace std;
//Funcion que muestra el vector
void muestra(string nom, const vector<int>&v, int c, int f) {
	for(size_t i=0;i<v.size();i++) {
		cout<< nom << "[" << right << setw(2) << i <<"]=" << setw(2) << v[i]<<", ";
		if(i== (c-1) || i == (f-1)) cout<<"/";
	}
	cout<< endl; //cin.get();
}

int main() {
//Declaro vector y asigno valores
	const int largo=30;//30
	int ini= 10;//10
	int fin= 21;//21

	vector<int> v1(largo);
	for(size_t i=0;i<v1.size();i++)
		v1[i] = rand()%25;
	muestra("v1",v1,ini,fin);//LLamo a la funcion muestra

	// a- Utilizo los algoritmos sort y reverse
	cout<<endl<<"Ordeno con sort de  "<<ini<<" a "<<fin-1<<", Tamanio: "<<v1.size()<<endl;
	sort(v1.begin()+ini,v1.begin()+fin);
	muestra("v1",v1,ini,fin);
	cout<<endl<<"Reordeno con reverse de "<<ini<<" a "<<fin-1<<", Tamanio: "<<v1.size()<<endl;
	reverse(v1.begin()+ini,v1.begin()+fin);
```

```
	muestra("v1",v1,ini,fin);

	// b - Creo otro vector con los elementos de 10 a 20
	vector<int> v2;
	v2.insert(v2.begin(),v1.begin()+ini,v1.begin()+fin);
	//Borro de v1 esos elementos.
	v1.erase(v1.begin()+ini,v1.begin()+fin);
	cout<<endl<<"Vector v1 sin los elementos de "<<ini<<" a "<<fin<<", Tamanio: "<<v1.size()<<endl;
	muestra("v1",v1,0,v1.size());
	cout<<endl<<"Vector v2 con los elementos de "<<ini<<" a "<<fin<<", Tamanio: "<<v2.size()<<endl;
	muestra("v2",v2,0,v2.size());

	// c- utilizo algoritmo count
	cout<<endl<<"Ingrese un valor a buscar en v2: "<<endl;
	int x;
	cin>> x;
	cout<< "Se encontró " << count(v2.begin(),v2.end(),x) << " veces." <<endl<< endl;

	/*d - auto: el compilador determina el tipo que sustituirá a la palabra
	clave auto como si estuviera utilizando las reglas para la deducción
	de argumento de plantilla a partir de una llamada de función.
	Tambien utilizare el algoritmo remove y el metodo vector::erase*/
	auto it = remove(v2.begin(),v2.end(),x);
	//Observer que remove coloco un 0 en donde estaba el numero
	cout<<endl<<"Vector v2, despues de remove "<<", Tamanio: "<<v2.size()<<endl;
	muestra("v2",v2,0,v2.size());
	//std::vector:: erase
	v2.erase(it,v2.end());//El metodo erase elimina la posicion de memoria
	cout<<endl<<"Vector v2, despues de erase "<<", Tamanio: "<<v2.size()<<endl;
	muestra("v2",v2,0,v2.size());
	//Comisión Viernes: Prof. Gerardo Sas.
}
```

Ejercicio 8.3

Cree un programa que lea valores flotantes por teclado y los inserte en una lista.
Luego:
a. Muestre dicha lista.
b. Inserte en medio de cada par de elementos contiguos el promedio dedichos elementos y guarde la lista resultante en un archivo de texto llamado *"listafloat.txt"*.
c. Responda: ¿es posible ordenar solamente una porción de la lista de la misma manera que se hizo con el vector en el ejercicio 8.2?(NO)

```
//© Programación Orientada a Objetos- F.I.C.H. - U.N.L.
//Sitio web: http://e-fich.unl.edu.ar
//Comisión Viernes: Prof. Gerardo Sas.
using namespace std;
void muestra(list<float>&lista){
	for(list<float>::iterator p=lista.begin();p != lista.end();p++){
		cout<< *p<<", ";
	}
	cout<< endl; cin.get();
}
//*****************************
int main(int argc, char *argv[]) {
```

```
	//(std::list) Declaro la lista y reservo 15 posiciones
	list<float> valores(15);
	//std:: iterator, utilizado para recorrer la lista
	list<float>::iterator p;
	//Para ingresar por teclado
	cout<<"Ingrese 15 números flotantes:"<<endl;
	for( p = valores.begin(); p != valores.end(); p++)
		//cin >> *p;
		*p= (float)(rand()%10);
	//a) mostrar por pantalla
	cout<<"\na) Valores(): tamaño:"<<valores.size()<<endl;
	muestra(valores);
	//b) Inserto cada 2 posiciones el promedio.
	//El iterador p apunta al primer elemento del vector
	p = valores.begin();
	list<float>::iterator a , b;
	for (unsigned i=1; i< valores.size(); i+=3){
		//Busco los dos datos
		a= p;
		b= ++p;
		valores.insert( ++p,(*(a) + *(b))/(-2));//calculo e inserto promedio
	};
	cout<<"\n\nb) Valores con los promedios. Tamaño: "<<valores.size()<<endl;
	muestra(valores);
	//abrir archivo para escritura
	ofstream archi("listafloat.txt");
	if(!archi.is_open()){cout<<"ERROR DE ARCHIVO";cin.get();exit(1);}
	// grabar los datos de la lista en el archivo
	p= valores.begin();
	cout<<"\nGrabar los datos de la lista en el archivo : "<<valores.size()<<endl;
	for( unsigned i=0; i<valores.size(); ++i, p++ ){
		archi<< *p<<endl;
		cout<<*p<<endl;
	}
	archi.close();
	return 0;
	//Comisión Viernes: Prof. Gerardo Sas.
}
```

Ejercicio 8.4
Declare un arreglo estático de 20 elementos enteros (*int v[20];*) y luego:

a. Implemente una función *int rand10()* que genere un entero aleatorio entre-10 y 10, y utilícela como argumento para *generate* para inicializar el arreglocon valores aleatorios.

b. Implemente una función *bool es_par(int x)* que retorne true si el entero querecibe es par; y utilícela en combinación con *count_if* para contar cuantoselementos pares hay en el contenedor generado.

c. Implemente una función *bool menor_abs(int a, int b)* que reciba dos enterosy retorne verdadero cuando el valor absoluto del primero sea menor que elvalor absoluto del segundo; y utilice esta función en como argumento de *sort* para ordenar el vector por valor absoluto.

d. Elimine los elementos repetidos utilizando los algoritmos genéricos de laSTL, y luego muestre el arreglo resultante.

```
//Sitio web: http://e-fich.unl.edu.ar
//Comisión Viernes: Prof. Gerardo Sas.

using namespace std;
//Salida por pantalla
void muestra(int v[], int n) {
	for(int i=0;i<n;i++)
		cout<< setw(5) << right << v[i];
	cout<< endl;
}

/* a) Implemente una función int rand10() que genere un entero aleatorio entre
	-10 y 10, y utilícela como argumento para generate para inicializar el
	arreglo con valores aleatorios.*/
int rand10() { return -10+rand()%21; }

/* b) Implemente una función bool es_par(int x) que retorne true si el entero
	que recibe es par; y utilícela en combinación con count_if para contar
	cuantos elementos pares hay en el contenedor generado.*/
bool es_par(int x) { return x%2==0; }

/* c) Implemente una función bool menor_abs(int a, int b) que reciba dos enteros
	y retorne verdadero cuando el valor absoluto del primero sea menor que el
	valor absoluto del segundo; y utilice esta función en como argumento de
	sort para ordenar el vector por valor absoluto.*/
bool menor_abs(int a, int b) { return abs(a)<abs(b); }
//*************************************************************************
int main() {
// Declare un arreglo estático de 20 elementos enteros (int v[20];)
	int v[20];

	// a
	generate(v,v+20,rand10);
	muestra(v,20);

	/* b) std:: count(p,u), std:: count_if(p,u,p)
	Devuelve el número de elementos en el rango [primero, último) que cumplan
	criterios específicos. La primera versión cuenta los elementos que son
	iguales a valor, la segunda versión cuenta con elementos para los cuales el
	predicado p retorna verdadero.*/
	cout<< "Cant pares: " << count_if(v,v+20,es_par) << endl;

	/* c) std::sort   Ordena los elementos en el rango [primero, último)
	en orden ascendente.
	La primera versión utiliza el operador < para comparar los elementos,
	la segunda versión utiliza la comparación de objetos función de los dado.
	*/
	sort(v,v+20,menor_abs);
	cout<<"Despues de sort(v,v+20,menor_abs)"<<endl;
	muestra(v,20);

	/* d) Elimine los elementos repetidos utilizando los algoritmos genéricos de la
		STL, y luego muestre el arreglo resultante.
	std:: unique
	unique( ForwardIt first, ForwardIt last ) ;		(1)
	unique( ForwardIt first, ForwardIt last, BinaryPredicate p ) ; (2)
	Elimina todos los elementos duplicados consecutivos desde el rango
```

```
        [primero, último) y devuelve un iterador past-the-end para el nuevo final
        lógico de la gama. La primera versión utiliza el operador == para comparar
        los elementos, la segunda versión utiliza el predicado binario dado p.
        Extracción es realizado por el desplazamiento de los elementos en el rango
        de tal manera que los elementos a ser borrados se sobrescriben.
        orden relativo de los elementos que permanecen se conserva y el tamaño
        físico del recipiente es sin cambios. Los iteradores que apuntan a un elemento
        entre el nuevo final lógico y el final físico de la gama son todavía Dereferenceable,
        pero los propios elementos tienen valores no especificados. Una llamada a la
        únique es seguido por una llamada al método de borrado de un contenedor, que
        borra los valores especificados y reduce el tamaño físico del contenedor
        para que coincida con su nuevo tamaño lógico.
        */
        //Para que funcione el unique hay que hacer un sort
        sort(v,v+20);
        int *vend = unique(v,v+20);
        int n = vend-v;
        cout<<"Despues de unique"<<endl;
        muestra(v,n);
        //Comisión Viernes: Prof. Gerardo Sas.
        return 0;
}
```

Ejercicio 8.5

Diseñe y programe una clase para manipular datos del tipo que se muestra en elrecuadro.

```
struct FichaMedico{
string nombreMedico;
vector < long > dniPacientes;
};
```

La clase será utilizada para administrar los pacientes de una clínica. La clínica estáconformada por 6 médicos y cada uno de ellos atiende a un número variable depacientes.

Crear una clase llamada *Clinica* que posea funciones para:

a. Incorporar nuevos pacientes indicando el número de médico y el DNI delpaciente.

b. Listar el nombre de cada médico y los DNI de los pacientes que atiende.

c. Eliminar un paciente indicando solamente su DNI.

Utilice la clase desarrollada desde un programa cliente.

```
//© Programación Orientada a Objetos- F.I.C.H. - U.N.L.
//Sitio web: http://e-fich.unl.edu.ar
//Comisión Viernes: Prof. Gerardo Sas.
using namespace std;
typedef struct{
        string nomMedico;
        list<long> dniPacientes;
}FichaMedico;

class Clinica{
protected:
        FichaMedico f[20];
        int largo;
public:
Clinica(){      largo =6;
```

```
                        f[0].nomMedico= "Peralta Rudecinda";
                        f[1].nomMedico="Mayo Juan";
                        f[2].nomMedico="Carnero Juan";
                        f[3].nomMedico="Perez Joaquin";
                        f[4].nomMedico="Barcelo Carlos";
                        f[5].nomMedico="Rodrigues Ramona";
                for(int k=0;k<6;k++)
                        f[k].dniPacientes.resize(10);}
//-------------------------------------------------
void mostrar(){
                for (int x=0; x< largo; x++)
                        cout<<x+1<<" Medico: "<< f[x].nomMedico <<endl;
        }
//-------------------------------------------------
void mostrarPaciente(int nm){
list<long>::iterator it;
int x=0;
        cout<<nm<<" Medico: "<< f[nm].nomMedico <<endl;
        it= f[nm-1].dniPacientes.begin();
        while(it != f[nm-1].dniPacientes.end()){
                if(*it!=0)
                        cout<<*it<<endl;
                else
                        cout<<"----------"<<endl;
                it++;
        }//while
}
//-------------------------------------------------
void modMedico(int nm, char *nomb){
                f[nm-1].nomMedico.clear();//borro nombre viejo
                f[nm-1].nomMedico += nomb;//agrego nombre nuevo
        }
//-------------------------------------------------
void agrPaciente(int nm, long num_dni){
                f[nm-1].dniPacientes.push_back(num_dni);//agrego dni al final
        }
//-------------------------------------------------
void modPaciente(int nm, long np, long dnip){//Numero de medico, dniP viejo, dniP nuevo
                list<long>::iterator it,principio, fin;
                principio= f[nm-1].dniPacientes.begin();
                fin= f[nm-1].dniPacientes.end();
                it= find( principio, fin, np);//busco y borro nombre viejo
                if(it != f[nm-1].dniPacientes.end())
                        *it= dnip;//modifico valor
                }
//-------------------------------------------------
//-------------------------------------------------
//-------------------------------------------------
};//Class Clinica
//*********************************************************

int main(int argc, char *argv[]) {
        Clinica C;
        int nummed;
        char op;
        while (op!='0'){
                system("cls");
```

```
                cout<<"         MENU DE OPCIONES"<<endl<<endl;
                cout<<"1 - MODIFICACIONES DATOS DE MEDICOS."<<endl;
                cout<<"2 - ALTA DE PACIENTES."<<endl;
                cout<<"3 - MODIFICACIONES DE PACIENTES."<<endl;
                cout<<"4 - LISTADO DE MEDICOS."<<endl;
                cout<<"5 - LISTADO DE PACIENTES POR MEDICO."<<endl;
                cout<<"6 - GUARDAR FICHAS EN DISCO."<<endl;
                cout<<"7 - RECUPERAR FICHAS DESDE DISCO."<<endl<<endl;
                cout<<"0 - SALIR"<<endl<<endl;
                cout<<"Ingrese su opcion:  ";  cin>> op;

                switch (op){
                case '1':{system("cls");
                cout<<"Ingrese numero de medico (1 - 6): ";
                cin>>nummed;
                cout<<"Medico ("<<nummed<<"): "<<endl;
                cout<<"Reingrese el nombre del medico: ";
                char nombre[30];
                cin.ignore();
                cin.getline(nombre,50);
                C.modMedico(nummed, nombre);
                break;};
                case '2':{cout<<"Ingrese numero de medico (1 - 6): ";
                cin>>nummed;
                cout<<"Medico ("<<nummed<<"): "<<endl;
                cout<<"Ingrese el DNI del paciente: ";
                long num_dni;
                cin>>num_dni;
                C.agrPaciente(nummed, num_dni);
                break;};
                case '3':{cout<<"Ingrese numero de medico (1 - 6): ";
                cin>>nummed;
                cout<<"Medico ("<<nummed<<"): "<<endl;
                cout<<"Ingrese el DNI del paciente: ";
                long num_dni;
                cin>>num_dni;
                //f[nummed-1].dni.push_back(num_dni);//agrego dni al final
                C.agrPaciente(nummed,num_dni);
                break;};
                case '4':{C.mostrar();
                system("pause");

                break;};
                case '5':{system("cls");
                cout<<"Ingrese numero de medico (1 - 6): ";
                cin>>nummed;
                C.mostrarPaciente(nummed);
                system("pause");
                break;};
                };//del switch
        };//del while

        return 0;
}//FICH - PROGRAMACIÓN ORIENTADA A OBJETOS - C++ - Prof. Gerardo Sas.
```

Cuestionario

1. ¿Qué es STL?¿Para qué sirve?¿Qué ventajas trae su uso?
2. ¿Qué es un contenedor?¿Cuáles son los tipos de contenedores?
3. ¿Qué es un iterador?¿Para qué sirve?
4. ¿En qué se diferencian los algoritmos de la STL de otros algoritmosconvencionales?
5. ¿Cuáles son las diferencias entre un vector y una lista? ¿En que casosconviene usar uno o el otro?
6. ¿Por qué la clase list implementa sus propios algoritmos como *sort()* enlugar de utilizar los algoritmos genéricos de la STL?

Ejercicios adicionales

Ejercicio 1

Diseñe una clase que permita manejar un archivo de configuración con el formatoque se muestra en el recuadro. Las opciones deberán ser almacenadas en unmapeo de tipo string a string, y los valores de las opciones serán convertidos aotros formatos (int, float, etc) cuando el usuario lo solicite.

```
unaopcion=5
otraopcion=hola
fuerzagravedad=9.8
```

La clase debe proveer funcionalidades para:

a. Leer un archivo de texto dado por el usuario y poblar el mapa con lasopciones encontradas.

b. Agregar una opción con su respectivo valor, si dicha opción ya existe suvalor debe actualizarse. Se recomienda que esta función tenga variassobrecargas para soportar distintos tipos de datos (int, float, etc) y los convierta a string para poder almacenarlos en el mapa.

c. Devolver por referencia el valor asociado a una determinada opción y unbooleano indicando si dicho valor se encontró o no en el mapa. Serecomienda que esta función tenga varias sobrecargas para soportar los diversos tipos de datos (int, float, etc).

d. Guardar la configuración modificada en un archivo de texto.

Se propone el siguiente prototipo para la clase:

```
class  Config {
public :
// constructores Config(); Config(string filename);
// cargar y guardar en archivo de texto
Load(string filename);
 Save(string filename);
 // pedir un valor
bool  GetValue(string option, float &value, float def_val=0);
bool  GetValue(string option, int &value, int  def_val=0);
bool  GetValue(string option, string &value,
string def_val="");
// agregar o modificar un valor bool bool  SetValue(string option, float newValue);  SetValue(string option, int newValue);
bool  SetValue(string option, string newValue);
private :
map<string,string> entries;};
```

Analice: ¿Podría reemplazar las sobrecargas de GetValue y SetValue por funciones genéricas?

```
#include <iostream>
#include <map>
#include <fstream>
#include <cstdlib>
#include <sstream>
using namespace std;
class  Config {
	public :
		// constructores Config();
		Config(string filename);
		// cargar y guardar en archivo de texto int int
		int Load(string filename);
		int Save(string filename);
		// pedir un valor
		bool  GetValue(string option, float& , float);
		bool  GetValue(string option, int& , int );
		bool  GetValue(string option, string&, string );
		// agregar o modificar un valor
		bool  SetValue(string option, float newValue);
		bool  SetValue(string option, int newValue);
		bool  SetValue(string option, string newValue);
		//Mostrar
		void Mostrar(){
			for(auto it=entries.begin(); it!= entries.end(); it++){
				cout<<it->first<<"="<<it->second<<endl;}
		}
	private :
		map<string,string> entries;
};
Config::Config(string filename){
	Load(filename);//LLamo al metodo Load() que es miembro de la clase
}
int Config::Load(string filename){
	ifstream lectura(filename);//Declaro objeto y abro archivo para lectura
	if(!lectura.is_open()){cout<<"Error"; return 1;}//Si da error sale
	string cadena,cad1,cad2;//Defino 3 cadenas
	int pos;
	while(lectura >> cadena){//Leo la cadena
		pos=cadena.find("=");//Busco el "="
		cad1=cadena.substr(0,pos);//corto miembro 1
		cad2=cadena.substr(pos+1);//corto membro 2
		entries[cad1]=cad2;//Agrego al mapa
	}
	return 0;
}
int Config::Save(string filename){
	ofstream escritura(filename);//Creo el objeto de salida al archivo
	if(!escritura.is_open()){cout<<"Error"; return (1);}//Si da error salgo
	for(auto it=entries.begin(); it!= entries.end(); it++){//Recorro el mapa
		escritura<<it->first<<"="<<it->second<<endl;}//Guardo los miembros en archivo
	return 0;
}
// pedir un valor
```

```
bool  Config::GetValue(string option, float &value, float def_val=0){
	stringstream contenedor;//Para guardar el par encontrado
	auto it = entries.find(option);//Busco option y si lo encuentra
	if (it!=entries.end()){
		contenedor<<it->second;//Guardo en el contenedor el segundo string
		contenedor>>value;//Convierto al tipo de value
		return true;
	}else
		return false;
}
bool  Config::GetValue(string option, int &value, int  def_val=0){
	stringstream contenedor;//Para guardar el par encontrado
	auto it = entries.find(option);//Busco option y si lo encuentra
	if (it!=entries.end()){
		contenedor<<it->second;//Guardo en el contenedor el segundo string
		contenedor>>value;//Convierto al tipo de value
		return true;
	}else
		return false;

}
bool  Config::GetValue(string option, string &value, string def_val=""){
	stringstream contenedor;//Para guardar el par encontrado
	auto it = entries.find(option);//Busco option y si lo encuentra
	if (it!=entries.end()){
		contenedor<<it->second;//Guardo en el contenedor el segundo string
		contenedor>>value;//Convierto al tipo de value
		return true;
	}else
		return false;
}
// agregar o modificar un valor
bool  Config::SetValue(string option, float newValue){
	stringstream contenedor;//Para guardar el par encontrado
	contenedor<<newValue;//Guardo en el contenedor parametro
	auto it = entries.find(option);//Busco option y si lo encuentra
	entries[option]=contenedor.str();
	return true;
}
bool  Config::SetValue(string option, int newValue){
	stringstream contenedor;//Para guardar el par encontrado
	contenedor<<newValue;//Guardo en el contenedor parametro
	auto it = entries.find(option);//Busco option y si lo encuentra
	entries[option]=contenedor.str();
	return true;
}
bool  Config::SetValue(string option, string newValue){
	stringstream contenedor;//Para guardar el par encontrado
	contenedor<<newValue;//Guardo en el contenedor parametro
	auto it = entries.find(option);//Busco option y si lo encuentra
	entries[option]=contenedor.str();
	return true;
}
//************************************************************************
int main(int argc, char *argv[]) {
	Config M("datosMap.txt");
	M.Mostrar();
```

```
	float v1;
	string o="fuerzagravedad";
	if(M.GetValue(o, v1))
	cout<< "Busco: "<<o<<"="<< v1 <<endl;
	else
		cout<< "No encuentro a "<<o<<endl;
	int v2;
	o = "unaopcion";
	if(M.GetValue(o, v2))
		cout<< "Busco: "<<o<<"="<< v2 <<endl;
	else
		cout<< "No encuentro a "<<o<<endl;
	string v3;
	o = "otraopcion";
	if(M.GetValue(o, v3))
		cout<< "Busco: "<<o<<"="<< v3 <<endl;
	else
		cout<< "No encuentro a "<<o<<endl;
	string campo1;
	int campo2;
	cout<<"Ingrese campo 1(string): ";cin>>campo1;
	cout<<"Ingrese campo 2(int): ";cin>>campo2;
	M.SetValue(campo1,campo2);
	float campo3;
	cout<<"Ingrese campo 1(string): ";cin>>campo1;
	cout<<"Ingrese campo 2(float): ";cin>>campo3;
	M.SetValue(campo1,campo3);
	string campo4;
	cout<<"Ingrese campo 1(string): ";cin>>campo1;
	cout<<"Ingrese campo 2(string): ";cin>>campo4;
	M.SetValue(campo1,campo4);
	M.Mostrar();
	M.Save("datosMap.txt");
	return 0;
}
```

Templatizando métodos propios

```
using namespace std;
class  Config {
	public :
		// constructores Config();
		Config(string filename);
		// cargar y guardar en archivo de texto
		int Load(string filename);
		int Save(string filename);
		// pedir un valor
		template<class T>
		bool  GetValue(string option, T& , float);
		// agregar o modificar un valor
		template<class T>
		bool  SetValue(string option, T newValue);
		//Mostrar
		void Mostrar(){
			for(auto it=entries.begin(); it!= entries.end(); it++){
				cout<<it->first<<"="<<it->second<<endl;}
```

```
		}
	private :
		map<string,string> entries;
};
Config::Config(string filename){
	Load(filename);//LLamo al metodo Load() que es miembro de la clase
}
int Config::Load(string filename){
	ifstream lectura(filename);//Declaro objeto y abro archivo para lectura
	if(!lectura.is_open()){cout<<"Error"; return 1;}//Si da error sale
	string cadena,cad1,cad2;//Defino 3 cadenas
	int pos;
	while(lectura >> cadena){//Leo la cadena
		pos=cadena.find("=");//Busco el "="
		cad1=cadena.substr(0,pos);//corto miembro 1
		cad2=cadena.substr(pos+1);//corto membro 2
		entries[cad1]=cad2;//Agrego al mapa
	}
	return 0;
}
int Config::Save(string filename){
	ofstream escritura(filename);//Creo el objeto de salida al archivo
	if(!escritura.is_open()){cout<<"Error"; return (1);}//Si da error salgo
	for(auto it=entries.begin(); it!= entries.end(); it++){//Recorro el mapa
		escritura<<it->first<<"="<<it->second<<endl;}//Guardo los miembros en archivo
	return 0;
}
// pedir un valor
template<class T>
bool  Config::GetValue(string option, T &value, float def_val=0){
	stringstream contenedor;//Para guardar el par encontrado
	auto it = entries.find(option);//Busco option y si lo encuentra
	if (it!=entries.end()){
		contenedor<<it->second;//Guardo en el contenedor el segundo string
		contenedor>>value;//Convierto al tipo de value
		return true;
	}else
		return false;
}

// agregar o modificar un valor
template<class T>
bool  Config::SetValue(string option, T newValue){
	stringstream contenedor;//Para guardar el par encontrado
	contenedor<<newValue;//Guardo en el contenedor parametro
	auto it = entries.find(option);//Busco option y si lo encuentra
	entries[option]=contenedor.str();
	return true;
}
//*************************************************************************
int main(int argc, char *argv[]) {
	Config M("datosMap.txt");
	M.Mostrar();
	float v1;
	string o="fuerzagravedad";
	if(M.GetValue(o, v1))
	cout<< "Busco: "<<o<<"="<< v1 <<endl;
```

```
        else
                cout<< "No encuentro a "<<o<<endl;
        int v2;
        o = "unaopcion";
        if(M.GetValue(o, v2))
                cout<< "Busco: "<<o<<"="<< v2 <<endl;
        else
                cout<< "No encuentro a "<<o<<endl;
        string v3;
        o = "otraopcion";
        if(M.GetValue(o, v3))
                cout<< "Busco: "<<o<<"="<< v3 <<endl;
        else
                cout<< "No encuentro a "<<o<<endl;
        string campo1;
        int campo2;
        cout<<"Ingrese campo 1(string): ";cin>>campo1;
        cout<<"Ingrese campo 2(int): ";cin>>campo2;
        M.SetValue(campo1,campo2);
        float campo3;
        cout<<"Ingrese campo 1(string): ";cin>>campo1;
        cout<<"Ingrese campo 2(float): ";cin>>campo3;
        M.SetValue(campo1,campo3);
        string campo4;
        cout<<"Ingrese campo 1(string): ";cin>>campo1;
        cout<<"Ingrese campo 2(string): ";cin>>campo4;
        M.SetValue(campo1,campo4);
        M.Mostrar();
        M.Save("datosMap.txt");
        return 0;
}
```

Ejercicio 2

El archivo *"datos.txt"* contiene una lista de valores flotantes dispuestos uno por línea. Lea el contenido del archivo y almacénelo en una lista utilizando el algoritmo copy.

Para esto haga lo siguiente:

1. Abra el archivo para lectura y cree un iterador de lectura para leer valoresflotantes que apunte al principio del archivo de la siguiente forma:

stream_iterator<float>p(archi).

2. Cree otro iterador que apunte al final del archivo e indique el fin del archivo.

3. Cree un contenedor para almacenar lo que se leerá.

4. Utilice el algoritmo *copy* para copiar los datos del archivo al contenedor.

Luego, realice las siguientes operaciones:

a. Calcule el promedio de los elementos del contenedor.

b. Reste dicho valor a cada uno de los elementos.

c. Guarde los datos modificados en el archivo utilizando nuevamente el algoritmo *copy*

```
using namespace std;

int main(int argc, char *argv[]) {
        //**************************************************************************
        //Como no tengo el archivo, lo genero y lo guardo.-
        // crear un contenedor para almacenar los flotantes
```

```
	list<float> lista(20);
	// (STL: generate) generar una lista de valores al azar
	generate( lista.begin(), lista.end(), rand );
	// abrir el archivo para escritura
	ofstream archi1( "datos.txt" );
	// crear un iterador de escritura para guardar los datos nuevos
	ostream_iterator <float>q( archi1, "\n" );
	// grabar los datos de la lista en el archivo
	cout<<"\nGenero Archivo con Nº total elementos: "<<lista.size()<<endl;
	//crear iterador para recorrer la lista
	list<float>::iterator p = lista.begin();
	for( unsigned i=0; i<lista.size(); ++i, q++,p++){
		*q = *p;
		cout<<*p<<endl;
	}
	archi1.close();
	//*************************************************************************
	//1-  abrir el archivo para lectura
	ifstream archi2( "datos.txt" );
	lista.clear();//Vacio el lista
	//Items 2, 3 y 4
	copy( istream_iterator<float>(archi2),istream_iterator<float> (), back_inserter(lista) );
	//cierro el archivo
	archi2.close();
	//a) calculo promedio
	float sum,prom;
	sum = accumulate(lista.begin(),lista.end(),0.0);
	prom = sum /lista.size();
	cout<<"\na) El promedio : "<<prom<<", total elementos: "<<lista.size()<<endl;
	p= lista.begin();
	//b) Resto de los elementos el promedio
	while(p != lista.end()){
		*p -= prom;
		p++;
	}
	// mostrar por pantalla (utilizando copy / ostream_iterator)
	cout<<"\nlista con el promedio restado, total elementos: "<<lista.size()<<endl;
	copy( lista.begin(), lista.end(),
		ostream_iterator<float>( cout, "\n" ) );
	archi1.open("datos.txt" );
	// c) grabar los datos de la lista en el archivo
	copy( lista.begin(), lista.end(), ostream_iterator<float>(q) );
	archi1.close();
	return 0;
}
```