|  |  |
| --- | --- |
|  | **Caratula para entrega de Prácticas** |
| Facultad de Ingeniería | Laboratorio de docencia |

Laboratorios de computación salas A y B

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Profesor:\_\_\_\_Ing. Marco Antonio Martinez Quintana\_\_\_

Asignatura:\_Estructura de Datos y Algoritmos I \_

Grupo:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_17\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

No. de Práctica(s):\_\_\_\_\_\_\_\_\_\_\_4°\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Integrante(s):Francisco Moises Barrera Guardia

No. de Equipo de  
 cómputo empleado:\_\_\_\_\_\_\_\_\_\_14\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

No. Lista o Brigada:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Semestre:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2°\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fecha de entrega:\_\_\_Marzo del 2020\_\_\_\_\_\_\_\_\_

Observaciones:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CALIFICACIÓN:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Objetivos*

Utilizarás funciones en lenguaje C que permiten reservar y almacenar información de

manera dinámica (en tiempo de ejecución).

*Introducción*

La memoria dinámica se refiere al espacio de almacenamiento que se reserva en tiempo de

ejecución, debido a que su tamaño puede variar durante la ejecución del programa.

El uso de memoria dinámica es necesario cuando a priori no se conoce el número de datos

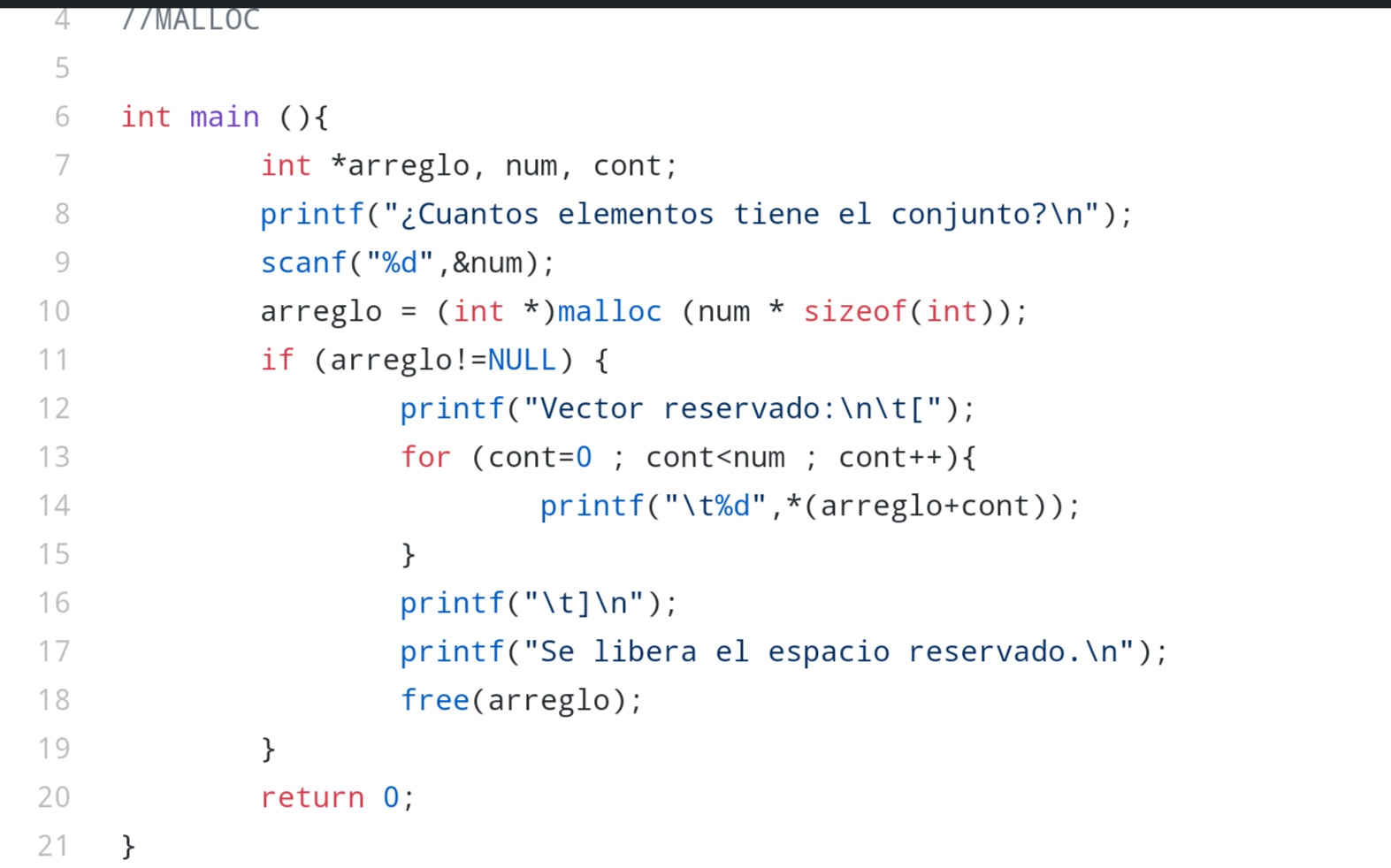
y/o elementos que se van a manejar.

*Desarrollo y Resultados*

**Actividades:**

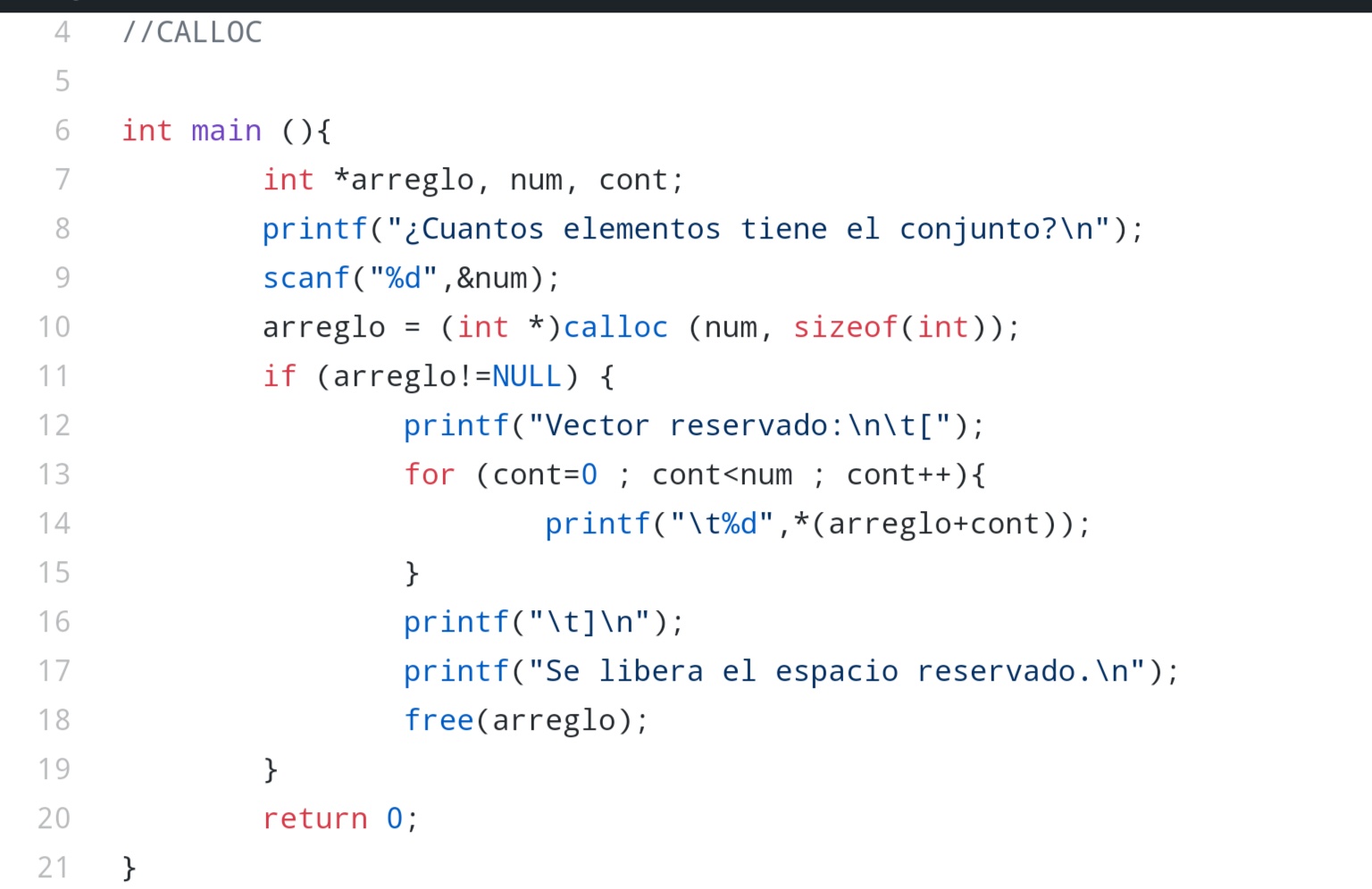
* **Utilizar funciones para reservar memoria dinámica en lenguaje C.**
* **Almacenar información en los elementos reservados con memoria dinámica.**

MALLOC

****

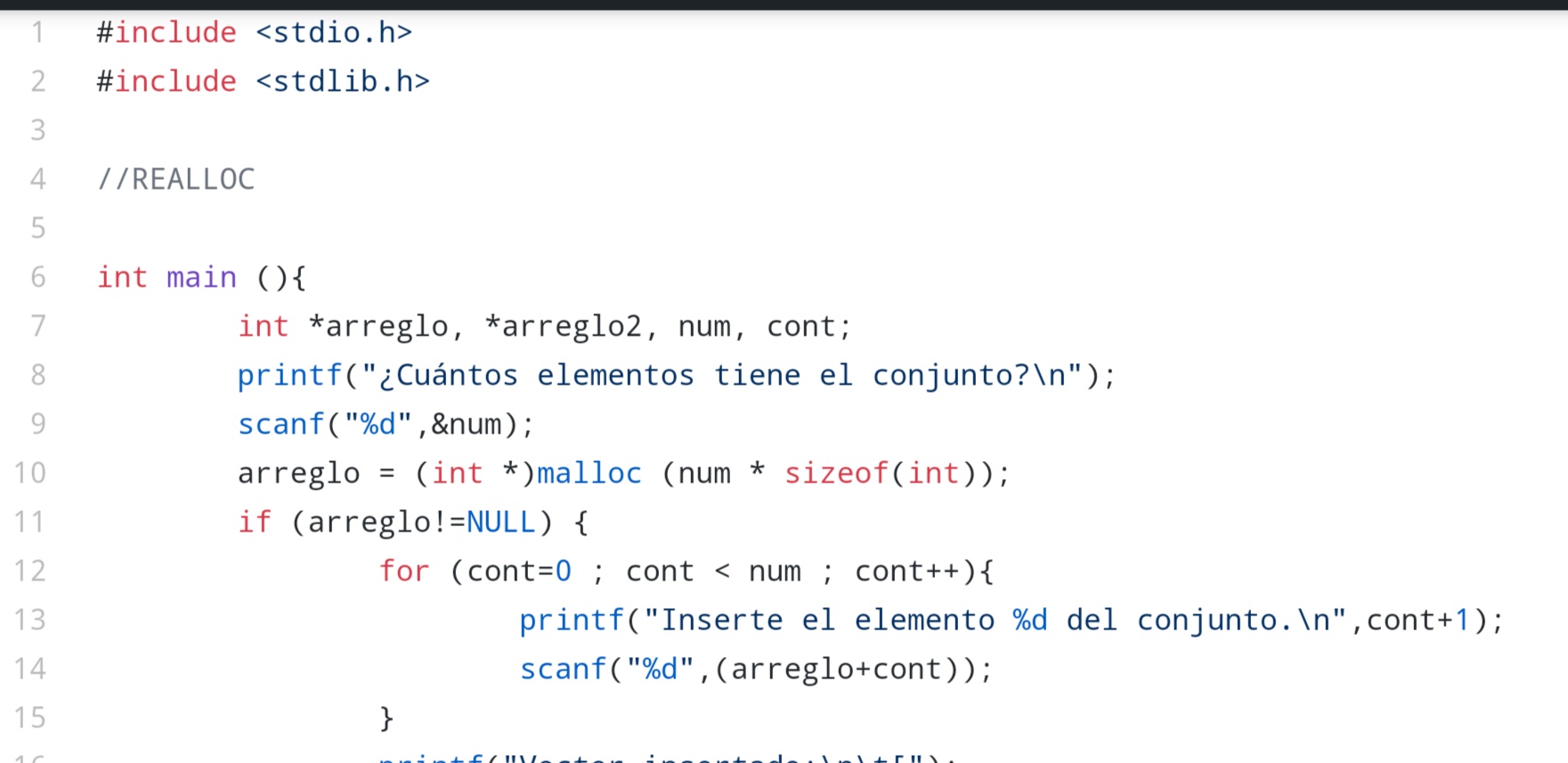
**Para el primer programa que es MALLOC tenemos un arreglo de tipo apuntador a entero,con 2 variables enteras aparte los cuales servirán para saber el tamaño de números del conjunto que imprimirá la sucesión de 1 hasta el número dado, para lo cual cuando el vector a sido reservado nos lo dirá y al final se liberaran los apuntadores con la función free.**

CALLOC

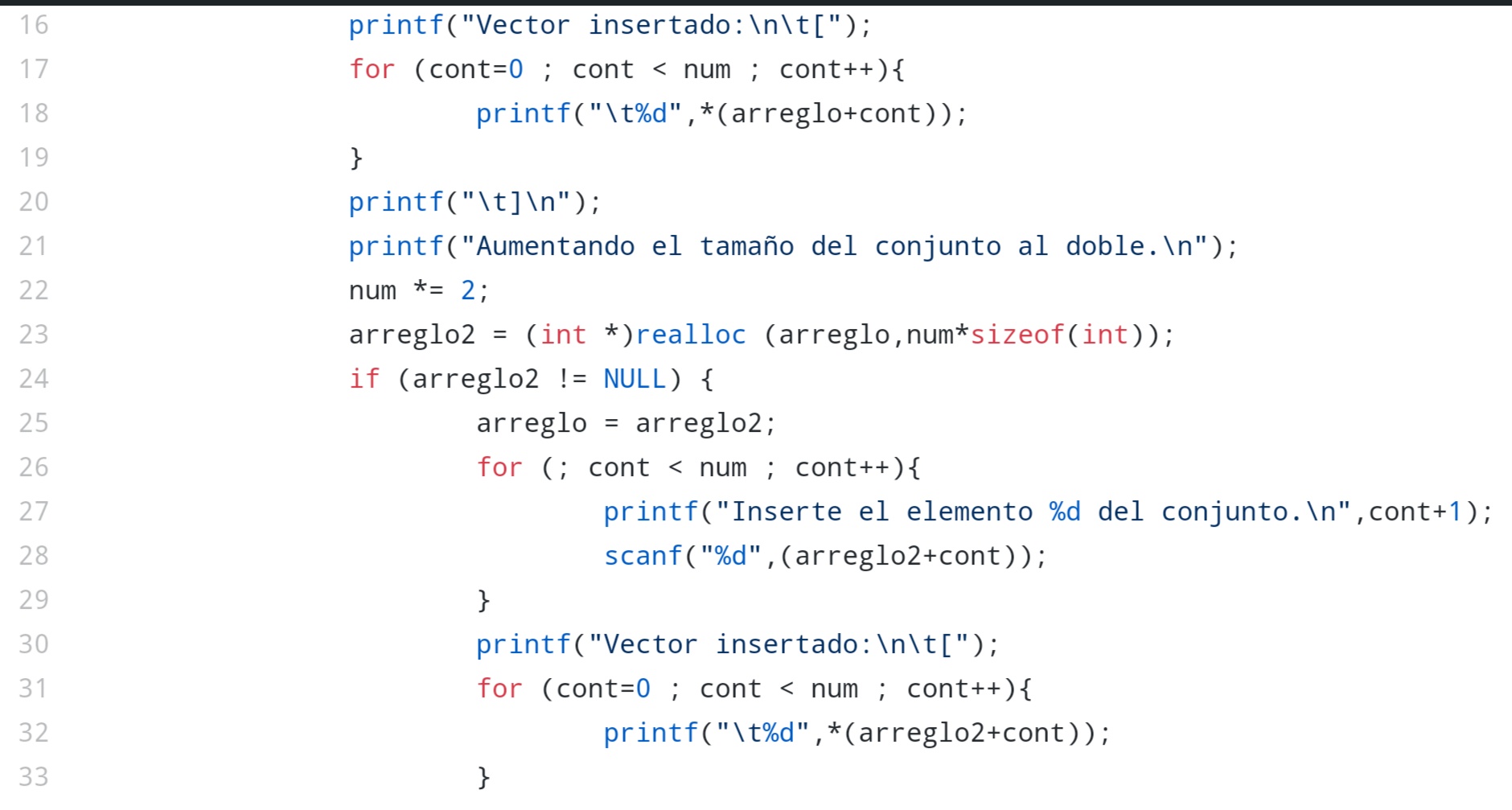
****

**En el segundo programa tenemos a Calloc el cual igual va inicializarse como apuntador a entero, junto con otras 2 variables del tipo entero y con ello pide al usuario la cantidad de elementos del arreglo para poder reservar memoria y a cada dato le asigna un numero para mostrarlo al usuario, después de ello libera la memoria. Lo que cabe resaltar de calloc es su inicialización en 0 de cada dirección a la que apunta, en diferencia con malloc.**

REALLOC

****

**Para el tercer y último programa de la función REALLOC tenemos 2 apuntadores a enteros y 2 variables enteras las cuales cuando inicia el programa le piden al usuario el número de elementos de un arreglo para reservar su memoria con malloc, después con un ciclo for tenemo que digitar el valor de cada dato del arreglo.**

****

**Despues el programa nos imprimirá todos los datos de cada valor reservado que le dimos anteriormente, para después aumentar el tamaño del arreglo al doble con la función realloc y volvernos a pedir valores para los demás espacios reservados e imprimirlos todos los datos al final.**

*Conclusiones*

Los objetivos si fueron cumplidos, ya que .se pudo comprender el funcionamiento de los trabajos que realizan malloc, calloc y realloc, ademas, de que se pudieron ver los codgigos punto a punto para corroborar que no tuvieran ningun error y compilaran de una manera agradable, y asi tener el conocimiento de su funcionamiento.

*Bibliografía*

[**http://lcp02.fi-b.unam.mx/**](http://lcp02.fi-b.unam.mx/)