



FACULTAD DE CIENCIAS EXACTAS, FÍSICAS Y NATURALES

Algoritmos y Estructuras de Datos

Ingeniería en Computación

Año 2013

Informe:

Trabajo Final

Simulación de una Ciudad

APELLIDO, Nombres:

Aguilar, Mauricio
Tarazi, Pedro Esequiel

Matrícula:

34.496.071
35.035.736

Fecha de Entrega: 08/03/2013

ENUNCIADO

El trabajo consiste en simular el tráfico en una ciudad. Usted conduce un vehículo desde un punto de la ciudad a otro, y desea hacerlo de la manera más rápida posible.

La ciudad esta conformada por manzanas cuadradas, calles de una sola mano y que tienen un solo carril y avenidas de doble mando de dos o tres carriles por mano. En cada esquina, existe un semáforo para evitar colisiones entre autos. Cada turno de semáforo verde permite que pasen n autos por carril, por lo que en una avenida pasaran $2*n$ o $3*n$, según la cantidad de carriles que disponga. El valor de n debe definirse por anticipado, pero una vez definido, no cambia. Cada carril tiene capacidad para alojar solo $3*n$ vehículos. Si al cruzar un semáforo, se desea ir hacia una calle o avenida con el/los carril/es con la capacidad colmada, dicho vehículo no avanza.

En la ciudad también circulan ambulancias y patrulleros de policía, los cuales tienen prioridad de paso, por lo que no deben esperar su turno (en la cola) para cruzar el semáforo, pero si deben esperar que el semáforo este en “verde” para pasar.

Usted deberá definir el “plano” de la ciudad donde simular el trafico de la forma que prefiera. Deberá respetar un mínimo de 64 manzanas, distribuidas como un cuadrado de 8×8 manzanas, con cuatro avenidas, dos por cada dimensión. La configuración deberá leerla desde un archivo donde podrán reflejarse cambios en el plano a simular.

Al comienzo de la simulación, deberá introducir en la ciudad un numero aleatorio de autos, ambulancias y patrulleros, cada cual con su origen y un destino. Una vez que el vehículo arribe a destino, puede sacarlo de la simulación. Puede suponer que los orígenes y destinos son las esquinas.

La forma en que los vehículos circulan es la siguiente: todos los semáforos se habilitan (verde) una vez por cada iteración, es decir, si hay s semáforos, se habilita cada uno de dichos semáforos. En ese momento circulan los vehículos que estaban esperando hacia la siguiente esquina, en la cantidad que corresponda, y se continua con el siguiente semáforo. En cada iteración se habilitan una sola vez todos los semáforos.

El orden en el cual se habilitan dentro de la iteración depende de la “carga” que tengas los semáforos, por lo que aquellos con colas mas largas deberán habilitarse antes que los de menor cola.

Deberá utilizar un grafo para representar la ciudad, una cola de prioridad por semáforo (implementada con una heap en arreglo), y deberá determinar el camino mas corto para cada vehículo, el cual definirá su recorrido. Para su vehículo, deberá redefinir dicho recorrido cada 2 iteraciones, teniendo en cuenta el estado de “saturación” del tráfico. Deberá generar por cada iteración un “ranking” de los semáforos mas saturados a los efectos de habilitarlos según ese orden. Al final de cada iteración deberá mostrar el estado de cada semáforo por pantalla o sobre un archivo, a los fines de realizar el seguimiento de la simulación.

DESCRIPCION DE LO REALIZADO EN EL PROGRAMA

El objetivo de este trabajo es demostrar que lo visto durante el dictado de clases fue comprendido de manera correcta por el alumno y familiarizarse con nuevos contenidos.

Para realizar este programa, usamos la clase Cola, entregada por los profesores en clases. Creamos nuevos miembros para dicha clase, además de nuevas estructuras que se detallaran a continuación:

- Vehículo: Es una estructura que tiene como miembros a dos enteros que sirven para tener un numero de identificación del vehículo y un orden de prioridad, y un char que indica el tipo de vehículo (auto, ambulancia o policía).
- Camino: Es una estructura que contiene a tres enteros, un char y tres miembros del tipo Cola. Los enteros sirven para el numero de identificación del vehículo, y para los vértices inicial y final de cada cuadra. El char indica el tipo de vehículo, y los miembros del tipo cola son punteros que tienen como objetivo identificar la cuadra de origen y destino del vehículo, y otro para guardar el camino mas corto definido por el método de Dijkstra.
- Etiqueta: Tiene como miembros a tres enteros y uno del tipo Cola. Los enteros sirven para indicar el numero de vértice, si fue marcado o no, y la distancia acumulada. El miembro del tipo cola es llamado NodoAnterior, y es un puntero que apunta al nodo anterior al nodo actual.

Para hacer mas fácil la lectura del programa, cada función utilizada fue guardada en librerías cuyo nombre hace alusión a la función que guarda. Ellas son:

- Asignación.h: En ella se encuentra la función **asignación**, que recibe tres enteros, un elemento del tipo Cola y otro del tipo Camino. En ella se asignan aleatoriamente los orígenes y destinos de cada vehículo, colocándolos en sus respectivos orígenes.
- Estructuras.h: En ella se encuentran las estructuras Cola, Vehiculo, Camino y Etiqueta antes explicadas.
- Matriz.h: Aquí se define la **Matriz de Adyacencias**, que sirve para identificar que nodos adyacentes tiene un nodo x.
- MostrarCalles.h: Aquí se define la función **MostrarCalles**, que recibe un elemento del tipo Cola, uno del tipo char, y dos enteros, y cuya función es imprimir las calles que se encuentran habitadas por al menos un auto.
- Mover.h: En ella se define la función **Mover**, cuyo argumento contiene a un elemento del tipo Cola, otro del tipo Camino y dos enteros. Aquí, primero se reordenan las calles de mas ocupadas a menos ocupadas, y luego se realizan los movimientos de los vehículos, hasta llegar a su destino. Esta función cumple con el camino mas corto entregado por la función Dijkstra, que se definirá mas adelante. Por cada llamada a esta función, se realizan 2 movimientos;
- Dijkstra.h: Aquí se define la función **Dijkstra**, cuyo objetivo es encontrar el camino mas corto para llegar desde el origen hasta el destino para cada vehículo. Tiene como argumentos a un elemento del tipo Cola, otro del tipo Camino, y tres enteros.
- Heapsort.h: Dentro de esta librería, se define la función **Heapsort**, que tiene como argumento a un elemento del tipo Cola y otro del tipo entero. Su objetivo es darle prioridad a ambulancias y policías por sobre los autos.

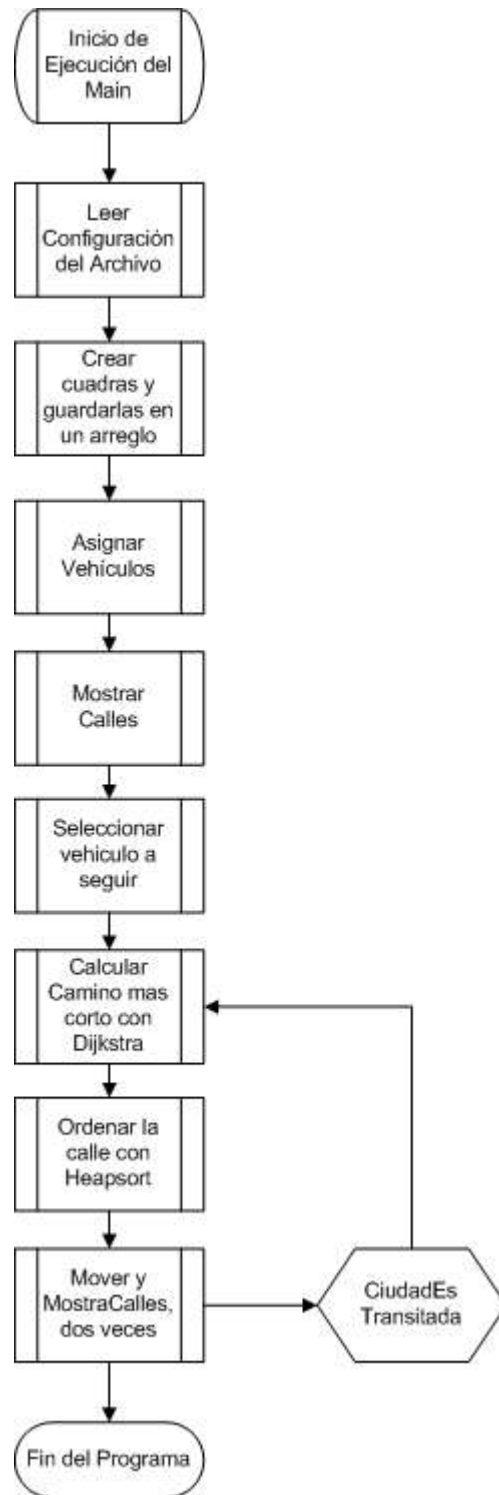
Por ultimo, se creo la función **main()**, que es la encargada de llevar a cabo el programa. En ella se lee la configuración de la ciudad de un archivo de texto y se crean las cuadras que definirán a la misma. Luego, dentro de ella, se llaman a las funciones definidas anteriormente con el objetivo de cumplir con lo pedido en el enunciado.

A continuación se mostrara el código de las estructuras creadas:

```
struct vehiculo{
    int numID, prioridad;
    char tipo; // a=auto ; p=policia ; h=ambulancia
};
class Cola{
private:
    Lista *l;
    Nodo *czoq;
    void update_start(Lista *l,Nodo *p);
    void do_desencolar(Nodo *p,Nodo *s);
public:
    Cola(void){l=new Lista();update_start(l,NULL);};
    ~Cola(void) {delete(l);};
    vehiculo * tope();
    int colavacia() { return l->esvacia();};
    void encolar(vehiculo *a) ;
    void desencolar();
    int Ultimo();
    int tamano(){return l->get_tam();};
    void Imprimir(char ts, int ns){l->toPrint(ts, ns);};
    int numero(){l->num();};
    int vi(){l->numvi();};
    int vf(){l->numvf();};
    int arista(){l->verArista();};
    void modArista(int numero) {l->ModArista(numero);};
    void addnum(int numero){l->addN(numero);};
    void addvi(int numero){l->addVI(numero);};
    void addvf(int numero){l->addVF(numero);};
};
struct Camino {
    int numID, Destino;
    char tipo;
    Cola *ori, *CamCorto[250];
    bool NoSeMovio;
};
struct Etiqueta {
    int D, iteracion;
    bool marcado;
    Cola *NodoAnterior;
    Etiqueta () {iteracion=0;};
};
```

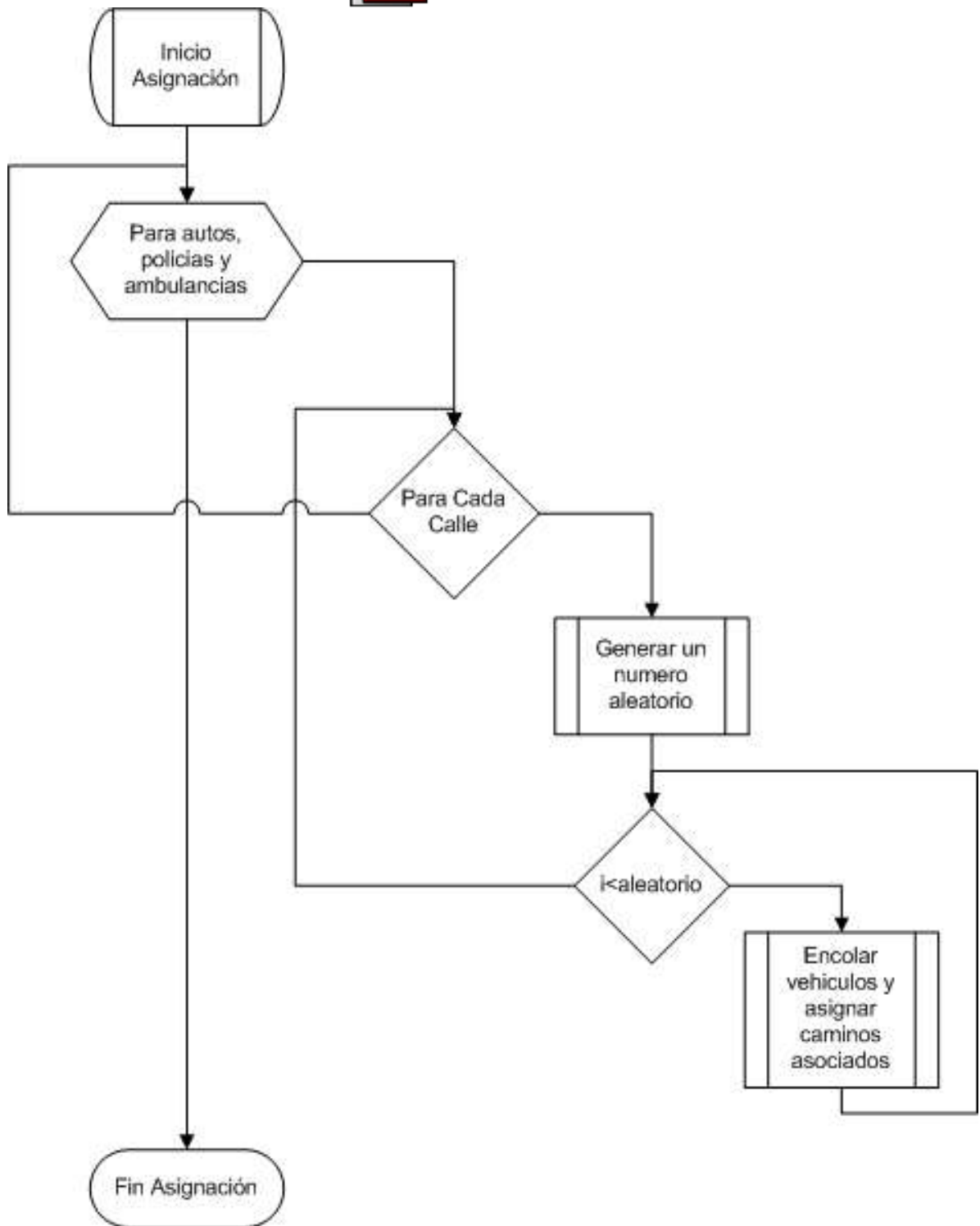


main
C++ Source File
9 KB



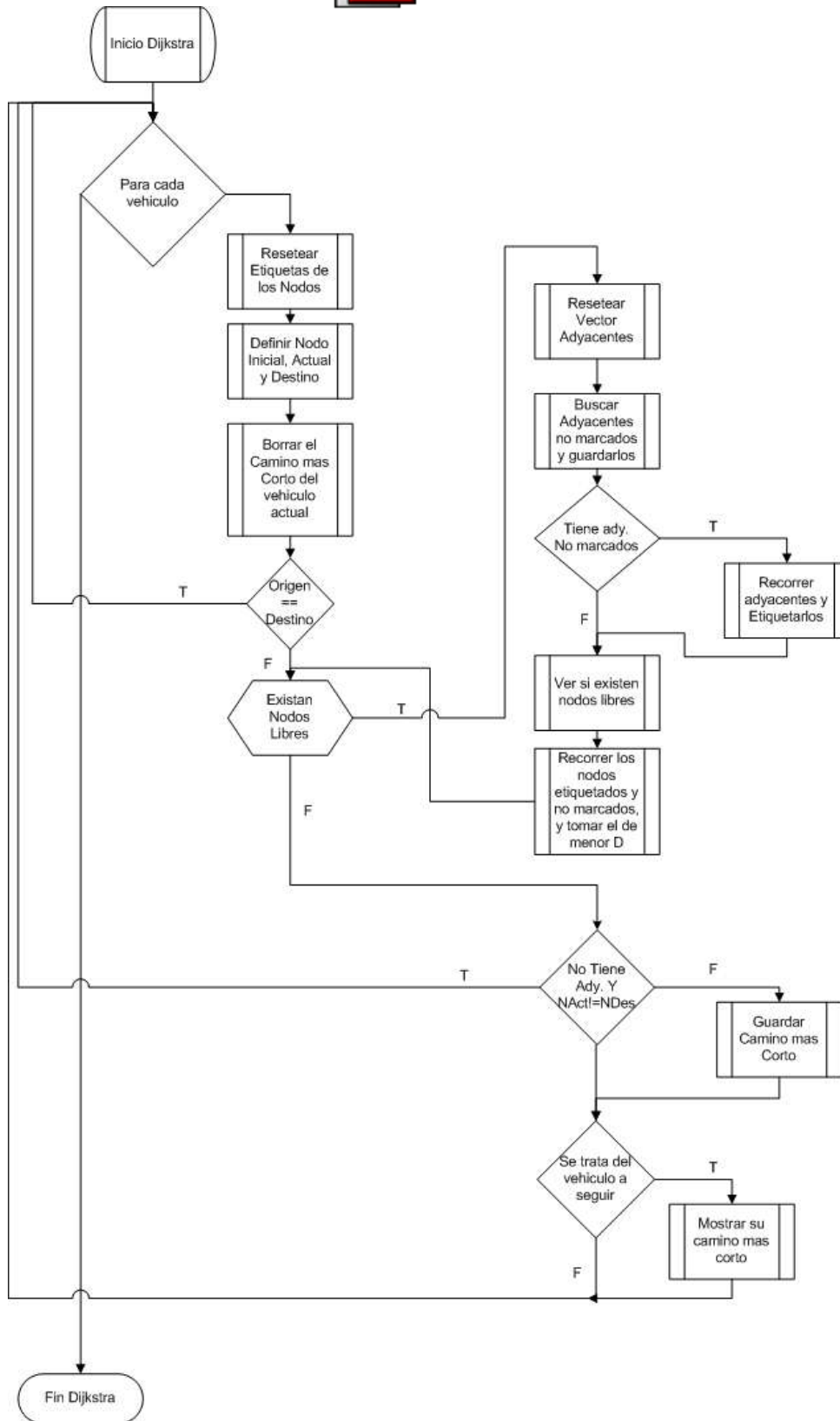


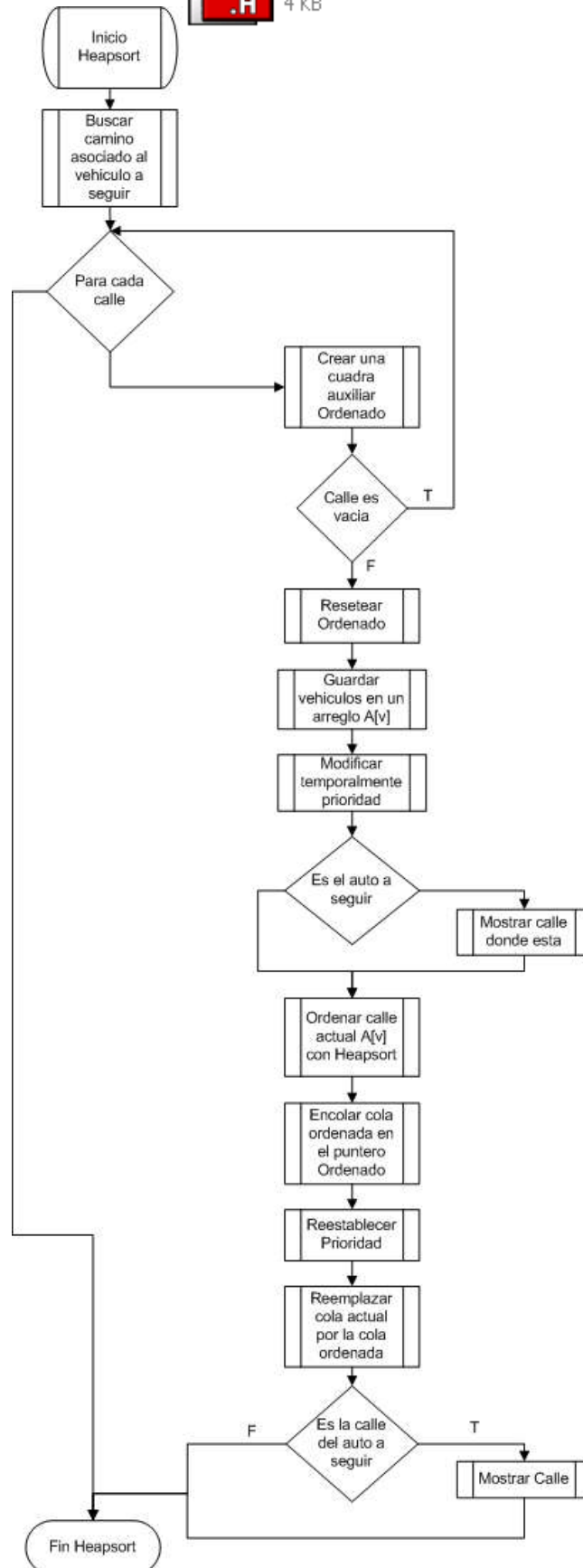
asignacion
C Header File
3 KB





Dijkstra
C Header File
7 KB







Mover
C Header File
5 KB

