

Las sucesivas coordenadas de un móvil se dan como números complejos, que serán almacenadas mediante un programa.

Se usarán las estructuras siguientes:

```
struct HORA {  
    unsigned hora,minuto,segundo;  
};
```

```
struct COMPLEJO {  
    float real,imaginaria;  
    struct HORA tiempo;  
};
```

```
struct lista_COMPLEJOS {  
    unsigned total;  
    struct COMPLEJO lista[N];  
};
```

```
POSICIONES DE UN MOVIL CON COORDENADAS COMPLEJAS  
1 Aniadir numero complejo (maximo 8)  
2 Eliminar ultimo numero complejo  
3 Listar numeros complejos  
0 Terminar  
  
posicion de un movil (numero complejo)  
    parte real? 1.2  
    parte imaginaria? -3.4  
hora de la posicion  
    hh:mm:ss ? 22:17:54  
  
Numero complejo insertado  
PULSE cualquier tecla _
```

```
POSICIONES DE UN MOVIL CON COORDENADAS COMPLEJAS  
1 Aniadir numero complejo (maximo 8)  
2 Eliminar ultimo numero complejo  
3 Listar numeros complejos  
0 Terminar  
  
complejo: 1.20-3.40i  
22:17:54  
  
Numeros complejos listados  
PULSE cualquier tecla
```

Figura 1. Ejemplos de ejecución del programa

Se definirán, y **se usarán, todas y cada una** de las funciones cuyos prototipos se dan en el siguiente recuadro.

//includes

```
#include "stdio.h"
#include <stdlib.h>
#include <conio.h>
```

// defines

```
#define N 8
```

// prototipos de las funciones que se deben definir en esta práctica

```
void print_HORA(struct HORA );
void scan_HORA(struct HORA *);
void scan_COMP (struct COMPLEJO *);
void print_COMP(struct COMPLEJO);
void print_menu();
void cualquier_tecla(); // espera que se pulse cualquier tecla, sin hacer echo
void inicializar_lista(struct lista_COMPLEJOS *);
void aniadir_COMPLEJO(struct lista_COMPLEJOS *,struct COMPLEJO);
void eliminar_COMPLEJO(struct lista_COMPLEJOS *);
// elimina el ultimo complejo añadido a la lista
void print_lista_COMPLEJOS(struct lista_COMPLEJOS);
```

int main(){

```
    char character;
    struct COMPLEJO c;
    struct lista_COMPLEJOS lista;
    inicializar_lista(&lista);
    do{
        printf("POSICIONES DE UN MOVIL CON COORDENADAS COMPLEJAS");
        printf("\n1 aniadir numero complejo (maximo 8) \n2 Eliminar ultimo complejo\n3
Listar numeros complejos\n0 Terminar\n");
        fflush(stdin);
        character = getchar();
        //character = getch();
        switch(character){
            case '1':
                printf("\nposicion de un movil (numero complejo)\n");
                scan_COMP(&c);
                aniadir_COMPLEJO(&lista,c);
                break;
            case '2':
                eliminar_COMPLEJO(&lista);
                break;
            case '3':
                print_lista_COMPLEJOS(lista);
                break;
            default:
                printf(" ");
        }
    }
```

```
    cualquier_tecla());

    } while(caracter!='0');
    return 0;
}
```

// funciones que se deben definir en esta práctica

```
void print_HORA(struct HORA h){
    printf("%u:%u:%u", h.hora, h.minuto, h.segundo);
}

void scan_HORA(struct HORA *c){
    unsigned hour, minute, second;
    do{
        printf("HORA? ");
        scanf("%u",&hour);
        c->hora = hour;

        printf("MIN? ");
        scanf("%u",&minute);
        c->minuto = minute;

        printf("SEG? ");
        scanf("%u",&second);
        c->segundo = second;
    }while(c->hora>24 && c->minuto>60 && c->segundo>60);
}

void scan_COMP (struct COMPLEJO *c){
    float real,ima;
    struct HORA new_hora;
    printf("Parte real? ");
    scanf("%f",&real);
    printf("Parte imaginaria? ");
    scanf("%f",&ima);
    scan_HORA(&new_hora);
    c->real = real;
    c->imaginaria = ima;
    c->tiempo = new_hora;
}

void print_COMP(struct COMPLEJO c){
    printf("\ncomplejo: %.2f%+.2fi\n",c.real,c.imaginaria);
    print_HORA(c.tiempo);
}

/*
void print_menu(){
    printf("\n1 Anadir numero complejo (maximo 8)");
    printf("\n2 Eliminar ultimo numero complejo");
    printf("\n3 Listar numeros complejos");
    printf("\n0 Terminar\n");
}*/
```

```
void cualquier_tecla(){
    //char x;
    printf("\nPULSA CUALQUIER TECLA PARA CONTINUAR");
    getch();
}

void inicializar_lista(struct lista_COMPLEJOS *c){
    c->total = 0;
    struct COMPLEJO complejo;
    struct HORA tiempo;
    complejo.imaginaria = 0.0;
    complejo.real = 0.0;
    tiempo.hora = 0;
    tiempo.minuto = 0;
    tiempo.segundo = 0;
    complejo.tiempo = tiempo;
    for(int i=0; i<8; i++){
        c->lista[i] = complejo;
    }
}

void aniadir_COMPLEJO(struct lista_COMPLEJOS *list, struct COMPLEJO c){
    list->lista[list->total] = c;
    list->total = list->total + 1;
    printf("\nNumero complejo insertado");
}

void eliminar_COMPLEJO(struct lista_COMPLEJOS *list){
    struct COMPLEJO actual;
    struct HORA h;
    actual.imaginaria = 0.0;
    actual.real = 0.0;
    h.hora = 0;
    h.minuto = 0;
    h.segundo = 0;
    actual.tiempo = h;
    list->lista[list->total] = actual;
    list->total=list->total - 1;
}

void print_lista_COMPLEJOS(struct lista_COMPLEJOS list){
    int max = list.total;
    for(int i=0; i<max; i++){
        print_COMP(list.lista[i]);
        printf("\n");
    }
}
```