LABORATORIO 1

1.- Utilizando las siguientes declaraciones C:

```
struct tiempo {
      int minutos;
      int segundos;
};

#define LONGITUD_TABLA 10

struct tiempo tabla[LONGITUD_TABLA];

void mostrar_tiempos() {
      /* Resuelve aqui este ejercicio */

}

int main() {
   printf("...");  /* Muestra el texto '--- Ejercicio 1' */
   mostrar_tiempos();
}
```

Escribe el código de la función **mostrar_tiempos()** que muestra en pantalla todos los valores de la tabla con el siguiente formato:

```
--- Ejercicio 1
Elemento 0: minutos 0 segundos 0
Elemento 1: minutos 0 segundos 0
Elemento 2: minutos 0 segundos 0
Elemento 3: minutos 0 segundos 0
Elemento 4: minutos 0 segundos 0
Elemento 5: minutos 0 segundos 0
Elemento 6: minutos 0 segundos 0
Elemento 7: minutos 0 segundos 0
Elemento 8: minutos 0 segundos 0
Elemento 9: minutos 0 segundos 0
```

2.- Utilizando tu solución del ejercicio anterior, añade al código de **main.c** el código necesario para que la tabla se inicialice con los siguientes valores (y muéstralos en pantalla):

```
--- Ejercicio 2
Elemento 0: minutos 0 segundos 10
Elemento 1: minutos 1 segundos 9
Elemento 2: minutos 2 segundos 8
Elemento 3: minutos 3 segundos 7
Elemento 4: minutos 4 segundos 6
Elemento 5: minutos 5 segundos 5
Elemento 6: minutos 6 segundos 4
Elemento 7: minutos 7 segundos 3
Elemento 8: minutos 8 segundos 2
```

```
#include <stdio.h>

struct tiempo {
int minutos;
int segundos;
};

#define LONGITUD_TABLA 10

struct tiempo tabla[LONGITUD_TABLA];

void mostrar_tiempos() {

  for(int i = 0; i < LONGITUD_TABLA; i++){
     printf("Elemento %d: minutos %d segundos %d\n", i,tabla[i].minutos,tabla[i].segundos);
   }
   printf("--- Ejercicio 2\n");
   for(int i = 0; i < LONGITUD_TABLA; i++){
     int valor_minutos = i;
     tabla[i].minutos = valor_minutos;
     tabla[i].segundos = LONGITUD_TABLA - i ;
     printf("Elemento %d: minutos %d segundos %d\n", i,tabla[i].minutos,tabla[i].segundos);
   }

}

int main() {

 printf("--- Ejercicio 1\n");
 mostrar_tiempos();
```

}

3.- Utilizando el siguiente tipo de dato definido en C:

```
struct tiempo {
      int minutos;
      int segundos;
};
```

Escribe en el fichero **main.c** el código necesario para que se muestre en pantalla una cuenta atrás.

Para realizar los bucles en este ejercicio sólo puedes utilizar bucles **while**; no es necesario que tu programa controle el tiempo entre cada mensaje que se muestra en pantalla puesto que requeriría conocimientos no explicados en esta asignatura.

En esta página de Wikipedia tienes documentación que te permite controlar el formato de salida de **printf()**: https://es.wikipedia.org/wiki/Printf

```
#define MINUTOS 2
#define SEGUNDOS 3
. . .
…
```

Extracto del código

```
Comienza la cuenta atrás
02:03
02:02
02:01
02:00
01:59
. . .
00:01
00:00
Fin de la cuenta atrás
```

Presentación en pantalla tras la ejecución

```
#include <stdio.h>
#define MINUTOS 2
#define SEGUNDOS 3

struct tiempo {
    int minutos;
    int segundos;
};

int main(){
    struct tiempo min, seg;
    min.minutos = MINUTOS;
    seg.segundos = SEGUNDOS;

    int resta =1;
        printf("Comienza la cuenta atrás\n");
    for(int i = SEGUNDOS; i >=0; i--){

            if(i >= 0){
                printf("0%d:0%d\n", min.minutos, i);
            }

   }

    for(int i = 59; i>=0 ; i--){


            if(i==0 && min.minutos >=0){
                printf("0%d:0%d\n", min.minutos-resta, i);
                resta++;

                for(int j =59; j >=0 ; j--){
                    if(j<=9){

                    printf("0%d:0%d\n", min.minutos-resta, j);
                    }else{

                    printf("0%d:%d\n", min.minutos-resta, j);
                    }
                }
```

```
        }else if(i <=9){

            printf("0%d:0%d\n", min.minutos-resta, i);
        }else{
            printf("0%d:%d\n", min.minutos-resta, i);
        }

    }
    printf("Fin de la cuenta atrás\n");
     return 0;
}
```

LABORATORIO 2

Clase main

```
#include <stdio.h>
#include "mi_cadena.h"

#include <string.h>
#include <assert.h>

/* ------------------------------------------------
                TESTS DE LONGITUD
   ------------------------------------------------ */

void test_longitud_vacia() {
    char texto1[] = "";

    printf("test longitud vacia ..... ");
    assert(strlen(texto1)==longitud(texto1));
    printf("ok\n");
}

void test_longitud_no_vacia() {
    char texto1[] = "Programación";

    printf("test longitud no vacia ..... ");
    assert(strlen(texto1)==longitud(texto1));
    printf("ok\n");
}

/* ------------------------------------------------
                TESTS DE COPIA
   ------------------------------------------------ */


void test_copia_vacia() {
    char texto1[] = "";
    char texto2[] = " ";

    printf("test copia vacia ..... ");
    copia(texto2, texto1);
    assert(strcmp(texto1,texto2) == 0);
    printf("ok\n");
}

void test_copia_tamano_igual() {
    char texto1[] = "1234";
```

```
    char texto2[] = "     ";

    printf("test copia a igual tamaño ..... ");
    copia(texto2, texto1);
    assert(strcmp(texto1,texto2) == 0);
    printf("ok\n");
}

void test_copia_tamano_mayor() {
    char texto1[] = "1234";
    char texto2[50];

    printf("test copia a tamaño mayor ..... ");
    copia(texto2, texto1);
    assert(strcmp(texto1,texto2) == 0);
    printf("ok\n");
}


/* ------------------------------------------------
                TESTS DE COMPARA
   ------------------------------------------------ */

/*
void test_compara_vacias() {
    char texto1[] = "";
    char texto2[] = "";
    assert(strcmp(texto1,texto2)==compara(texto1,texto2));
}

void test_compara_iguales() {
    char texto1[] = "Programación";
    char texto2[] = "Programación";
    assert(strcmp(texto1,texto2)==compara(texto1,texto2));
}

int check_strcmp(int value1, int value2) {
    if (value1 == value2)
      return 1;

    if (value1<0 && value2<0)
      return 1;

    return (value2>0 && value2>0);
}

void test_compara_distintas() {
    char texto1[] = "PrograMación";
    char texto2[] = "Programación";
    assert(check_strcmp(strcmp(texto1,texto2),
                    compara(texto1,texto2)));
}

void test_compara_menor_tamano() {
    char texto1[] = "Progra";
    char texto2[] = "Programación";
    assert(check_strcmp(strcmp(texto1,texto2),
                    compara(texto1,texto2)));
}
```

```
void test_compara_mayor_tamano() {
    char texto1[] = "Programación";
    char texto2[] = "Progra";
    assert(check_strcmp(strcmp(texto1,texto2),
                        compara(texto1,texto2)));
}

void test_compara_menor() {
    char texto1[] = "A";
    char texto2[] = "B";
    assert(check_strcmp(strcmp(texto1,texto2),
                        compara(texto1,texto2)));
}

void test_compara_mayor() {
    char texto1[] = "B";
    char texto2[] = "A";
    assert(check_strcmp(strcmp(texto1,texto2),
                        compara(texto1,texto2)));
}
*/

/* ------------------------------------------------
                TESTS DE CONCATENA
   ------------------------------------------------ */

/*
void test_concatena_dos_strings_vacias() {
    char texto1[] = "";
    char texto2[] = "";
    concatena(texto2, texto1);
    assert(strcmp(texto1,texto2) == 0);
}

void test_concatena_una_string_vacia() {
    char texto1[] = "";
    char texto2[] = "Programación";
    concatena(texto2, texto1);
    assert(strcmp(texto2,"Programación") == 0);
}

void test_concatena() {
    char texto1[]  = "Programación";
    char texto2[]  = " EITE";
    char frase[50] = "";

    concatena(frase, texto1);
    concatena(frase, texto2);
    assert(strcmp(frase,"Programación EITE") == 0);
}
*/

int main() {
    test_longitud_vacia();
    test_longitud_no_vacia();

    test_copia_vacia();
    test_copia_tamano_igual();
    test_copia_tamano_mayor();
```

```
    /* Tests para los ejercicios avanzados ********
       Aviso: estos tests no generan salida a pantalla; sólo se
       muestra un error en caso de fallo.
       ********************
    */

/*  test_compara_vacias();
    test_compara_iguales();
    test_compara_distintas();
    test_compara_menor_tamano();
    test_compara_mayor_tamano();
    test_compara_menor();
    test_compara_mayor();

    test_concatena_dos_strings_vacias();
    test_concatena_una_string_vacia();
    test_concatena();
*/

    return 0;
}
```

Clase micadena.h

```
#ifndef MI_CADENA_H_
#define MI_CADENA_H_
int longitud(const char *str);
void copia(char *destino, const char *origen);

#endif
```

Clase micadena.c

```
#include "mi_cadena.h"
#include <stdio.h>
```

1.- Añade al fichero **mi_cadena.h** la declaración de la siguiente función y programa en el fichero **mi_cadena.c** el cuerpo de la función.

```
/* Longitud de una cadena de caracteres

        int longitud(const char *str);

@param str Puntero al comienzo de la string C. Está declarado
           'const' para indicar que es un puntero que no puede
           modificar el contenido al que apunta (pero si que se
           puede mover para recorrer los caracteres de la string
           a la que apunta).

@return Longitud en caracteres de la string
*/
```

En C la longitud de una *string* se calcula recorriendo todos los caracteres de la *string* desde el principio de la *string* hasta encontrar el carácter de terminación '\0' (sin contar el '\0').

No debe confundirse con el tamaño en bytes de una *string* (*size*) que es el tamaño total de la *string*. Por ejemplo:

```
char mystr[100]="test string";
```

... define un *array* de caracteres con un tamaño (*size*) de 100 caracteres que está inicializado con una *string* de longitud 11 caracteres. Por lo tanto, **sizeof(mystr)** retorna 100, y **longitud(mystr)** retorna 11.

Para comprobar tu solución descomenta las siguientes líneas en el fichero **main.c**:

```
int main() {
     /* test_longitud_vacia();       */
     /* test_longitud_no_vacia();    */
      ...
}
```

Cuando finalices este ejercicio sube los ficheros (mi_cadena.h, mi_cadena.c y main.c) al VPL y comprueba que tu código es correcto picando en el botón "Evaluar" del VPL.

```
int longitud(const char *str){
```

```
    int contador = 0;
    while(*str != '\0' ){
       str++;
      contador++;
   }
   return contador;
}
```

2.- Añade al fichero **mi_cadena.h** la declaración de la siguiente función y programa en el fichero **mi_cadena.c** el cuerpo de la función.

```
/* Copia de cadenas de caracteres

   void copia(char *destino, const char *origen);

   @param origen  Puntero al comienzo de la string C que vamos a copiar.
   @param destino      Puntero al comienzo del array de caracteres donde
                       copiamos todo el contenido de la cadena origen
                       (incluyendo el terminador '\0').
*/
```

Para comprobar tu solución descomenta los tests de copia en el fichero **main.c**.

```
 ...
/* test_copia_vacia();          */
/* test_copia_tamano_igual();   */
/* test_copia_tamano_mayor();   */
 ...
```

```
void copia(char *destino, const char *origen){

   while(*origen != '\0'){
     *destino=*origen;
      origen++;
      destino++;

   }
    *destino=*origen;
}
```

3.- Crea el fichero **.h** de una biblioteca (**mi_tiempo.h**) que contenga las siguientes declaraciones:

```
struct tiempo {
      int minutos;
      int segundos;
};

#define LONGITUD_TABLA 10

struct tiempo tabla[LONGITUD_TABLA];

void mostrar_tiempos (struct tiempo *p);
/* @param p Puntero al comienzo de la tabla de tiempos

   Esta función muestra el contenido de la tabla; el formato de salida
   es el formato del ejercicio básico 2 de la práctica C1.
*/

void inicializar_tiempos (struct tiempo *p);
/* @param p Puntero al comienzo de la tabla de tiempos

   Esta función inicializa el contenido de la tabla con los mismos
   valores de minutos y segundos que el ejercicio básico 3 de la
   práctica C1.
*/
```

Añade el fichero .c de esta biblioteca (**mi_tiempo.c**)

Cuando finalices estos ejercicios sube los ficheros (mi_tiempo.h, mi_tiempo.c y main.c) al VPL y comprueba que tu código es correcto picando en el botón "Evaluar" del VPL.

```c
#include <stdio.h>
#include "mi_tiempo.h"

int main() {
   printf("--- Tabla no inicializada:\n");
   mostrar_tiempos(tabla);

   printf("--- Tabla inicializada:\n");
   inicializar_tiempos(tabla);
   mostrar_tiempos(tabla);

   /* Ejercicios avanzados ******* */

   /*  mostrar_n_tiempos(tabla, 5);     */

   /*  invertir_contenido_tabla(tabla); */

   /*  mostrar_tabla(tabla); */

}
```