Práctica 7 Manejo de Estructuras

# Objetivo

Durante el desarrollo de esta práctica se buscará la utilización de estructuras, apuntadores a estructuras, arreglos de caracteres, apuntadores a arreglos, y la codificación de funciones que devuelven apuntadores a arreglos de caracteres.

# Introducción

Como punto de partida se presentará una función X,

X:Ω→{0,1,4}

medible discreta definida sobre el conjunto

Ω = {DOMINGO,LUNES,MARTES,MIERCOLES,JUEVES,VIERNES,SABADO},

de acuerdo a como se indica en la siguiente tabla

|  |  |
| --- | --- |
| ω | X(ω) |
| DOMINGO | 0 |
| LUNES | 4 |
| MARTES | 0 |
| MIERCOLES | 4 |
| JUEVES | 1 |
| VIERNES | 0 |
| SABADO | 0 |

A continuación se presentan algunas líneas de código con las que se puede codificar la función X. En este caso, la función se codificó con el nombre funcion\_LintroProg. Primero se definió un tipo de dato compuesto a través de la siguiente declaración de estructura:

/\*\*PO\_DiaNumDLab.h\*/

struct PO\_DiaNumDLab {

short NumDLab;

char cadena[];

};

Después, en el archivo LintroProg.c se instanciaron siete estructuras de tipo struct PO\_DiaNumDLab:

/\*\*LintroProg.c\*/

struct PO\_DiaNumDLab PO\_DNdL\_0 = {0,"DOMINGO"};

struct PO\_DiaNumDLab PO\_DNdL\_1 = {4,"LUNES"};

struct PO\_DiaNumDLab PO\_DNdL\_2 = {0,"MARTES"};

struct PO\_DiaNumDLab PO\_DNdL\_3 = {4,"MIERCOLES"};

struct PO\_DiaNumDLab PO\_DNdL\_4 = {1,"JUEVES"};

struct PO\_DiaNumDLab PO\_DNdL\_5 = {0,"VIERNES"};

struct PO\_DiaNumDLab PO\_DNdL\_6 = {0,"SABADO"};

estas estructuras se usaron para inicializar en el mismo archivo, el siguiente arreglo de apuntadores a estructura:

/\*\*LintroProg.c\*/

/\*\*Arreglo de apuntadores a estructura\*/

struct PO\_DiaNumDLab \*LIntroProg[] = {

/\*0\*/ &PO\_DNdL\_0,

/\*1\*/ &PO\_DNdL\_1,

/\*2\*/ &PO\_DNdL\_2,

/\*3\*/ &PO\_DNdL\_3,

/\*4\*/ &PO\_DNdL\_4,

/\*5\*/ &PO\_DNdL\_5,

/\*6\*/ &PO\_DNdL\_6

};

El arreglo de apuntadores a estructura LintroProg se utilizó en el siguiente código para codificar la función short funcion\_LIntroProg(char dia[16])como se muestra a continuación:

/\*\*LintroProg.c\*/

/\*\*

funcion\_LIntroProg - Devuelve el n\'umero de laboratorio

correspondiente al d\'ia que se recibe como argumento.

@pre: El arreglo dia debe contener el nombre de uno de

los d\'ias de la semana en espa\~nol.

\*/

short funcion\_LIntroProg(char dia[16])

{

short index;

char DIA[][16] = {"DOMINGO","LUNES","MARTES","MIERCOLES",

"JUEVES","VIERNES","SABADO"};

for(index=0;index<(sizeof(DIA)/sizeof(DIA[0]));index++){

if(!strcmp(dia,DIA[index])){/\*\*``Los d\'ias son iguales''\*/

return LIntroProg[index]->NumDLab;

}

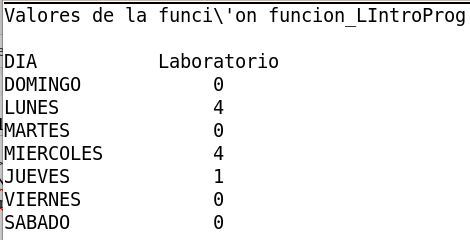
}

return -1;

}

Ya con esta función codificada, se pudo obtener la tabla que muestra los valores de la función X para cada uno de los elementos de su dominio

Ω = {DOMINGO,LUNES,MARTES,MIERCOLES,JUEVES,VIERNES}.



Los siguientes son los archivos de un proyecto con el cual se obtendrá una tabla con los pares ordenados correspondientes a la función X (codificada con el nombre de función funcion\_LintroProg).

/\*\*PO\_DiaNumDLab.h\*/

#ifndef PO\_DIANUMDLAB\_H\_INCLUDED

#define PO\_DIANUMDLAB\_H\_INCLUDED

struct PO\_DiaNumDLab {

short NumDLab;

char cadena[];

};

#endif // PO\_DIANUMDLAB\_H\_INCLUDED

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

/\*\*LintroProg.c\*/

#include <string.h>

#include "PO\_DiaNumDLab.h"

struct PO\_DiaNumDLab PO\_DNdL\_0 = {0,"DOMINGO"};

struct PO\_DiaNumDLab PO\_DNdL\_1 = {4,"LUNES"};

struct PO\_DiaNumDLab PO\_DNdL\_2 = {0,"MARTES"};

struct PO\_DiaNumDLab PO\_DNdL\_3 = {4,"MIERCOLES"};

struct PO\_DiaNumDLab PO\_DNdL\_4 = {1,"JUEVES"};

struct PO\_DiaNumDLab PO\_DNdL\_5 = {0,"VIERNES"};

struct PO\_DiaNumDLab PO\_DNdL\_6 = {0,"SABADO"};

/\*\*Arreglo de apuntadores a estructura\*/

struct PO\_DiaNumDLab \*LIntroProg[] = {

/\*0\*/ &PO\_DNdL\_0,

/\*1\*/ &PO\_DNdL\_1,

/\*2\*/ &PO\_DNdL\_2,

/\*3\*/ &PO\_DNdL\_3,

/\*4\*/ &PO\_DNdL\_4,

/\*5\*/ &PO\_DNdL\_5,

/\*6\*/ &PO\_DNdL\_6

};

/\*\*

funcion\_LIntroProg - Devuelve el n\'umero de laboratorio

correspondiente al d\'ia que se recibe como argumento.

@pre: El arreglo dia debe contener el nombre de uno de

los d\'ias de la semana en espa\~nol.

\*/

short funcion\_LIntroProg(char dia[16])

{

short index;

char DIA[][16] = {"DOMINGO","LUNES","MARTES","MIERCOLES",

"JUEVES","VIERNES","SABADO"};

for(index=0;index<(sizeof(DIA)/sizeof(DIA[0]));index++){

if(!strcmp(dia,DIA[index])){/\*\*``Los d\'ias son iguales''\*/

return LIntroProg[index]->NumDLab;

}

}

return -1;

}

(\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*)

El archivo que contiene la función main se muestra a continuación:

#include <stdio.h>

#include <stdlib.h>

#include "PO\_DiaNumDLab.h"

short funcion\_LIntroProg(char dia[16]);

int main(int argc,char \*argv[])

{

short i;

/\*\*Dominio de la funci\'on funcion\_LIntroProg\*/

char day[][16] = {"DOMINGO","LUNES","MARTES","MIERCOLES",

"JUEVES","VIERNES","SABADO"};

printf("\nValores de la funci\\'on funcion\_LIntroProg\n\n");

printf("%-14s%-14s\n","DIA","Laboratorio");

for(i=0;i<(sizeof(day)/sizeof(day[0]));i++){

printf("%-14s%6i\n",day[i],funcion\_LIntroProg(day[i]));

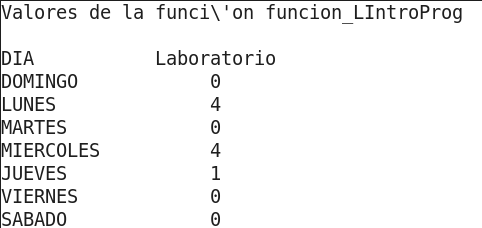
}

return 0;

}/\*end main()\*/

Construyendo el proyecto 07Practica07\_00 y corriendo el ejecutable se obtiene la tabla de

Valores de la función funcion\_LIntroProg (función Laboratorio de Introducción a la Programación)



Función char (\*funcion\_LIntroProg\_Inv(short ndl))[]

En el archivo PO\_DiaNumDLab.h se agregó

struct PO\_NumDLab\_Days {

short num\_d\_lab;

char char\_arr[];

};