**UNIVERSIDAD DE GUADALAJARA**

**CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS**

**DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN**

**DEPARTAMENTO DE CIENCIAS COMPUTACIONALES**

**Carrera:** Ingeniería en Computación

**Nombre Materia:** Estructura de datos ll

**SECCIÓN:** D09

**ACTIVIDAD #4**

**Nombre alumno:** López Arellano Ricardo David

**CÓDIGO:** 21713643



**FECHA:** 018/04/2022

**PROF.** Guerrero Segura Ramirez Miguel Angel

**REPORTE DE LA PRÁCTICA**

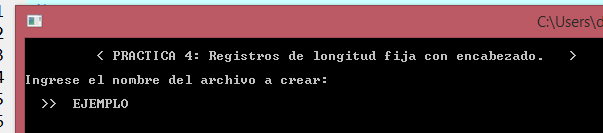
1.- Se busca generar un archivo de los participantes de un maratón organizado por un ayuntamiento, el cual maneja la siguiente información de cada corredor:

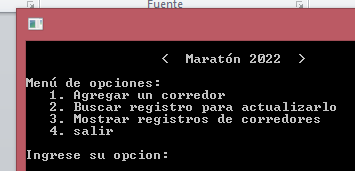
* Número de participación.
* Nombre de atleta.
* Dirección del atleta
* Ciudad.
* Nacionalidad,
* Tiempo de clasificación

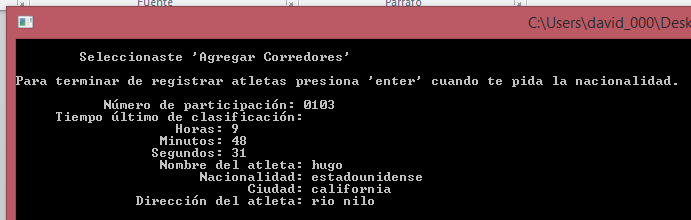
La información se tiene que almacenar en un archivo del que se le solicitará el nombre al usuario,  en el archivo se guardará al inicio un encabezado, este encabezado contendrá una variable entera que será un contador de registros y un relleno de 30 caracteres, después del encabezado se insertarán cada uno de los registros los que medirán 64 bytes cada uno. El número de participación tendrá un formato de 4 caracteres con un límite de 2000 participantes. El tiempo de clasificación es el mejor tiempo registrado en los últimos 3 meses.

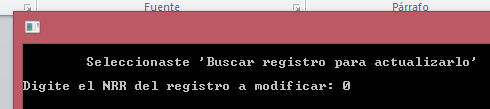
Para consultar la información de cada registro se solicitará al usuario el NRR (número relativo de registro) y en base a él se calculará la posición en donde está ubicado el registro en el archivo.

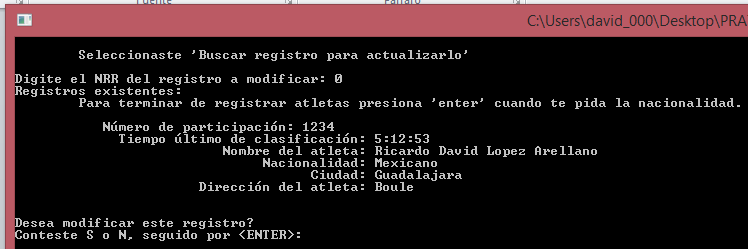
**Esto fue lo que realice para la práctica:**

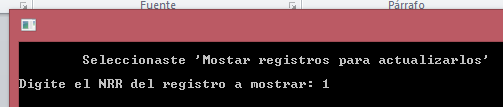


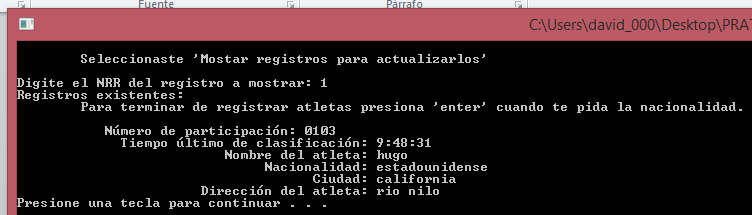












**Código fuente utilizado:**

/\*

Nombre: Ricardo David López Arellano

Código: 217136143

Materia: Estructuras de Datos II

Sección: D09

Pofesor: MIGUEL ANGEL GUERRERO SEGURA RAMIREZ

\*/

/\*LIBRERIAS\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <locale.h>

#include <unistd.h>

#include <conio.h>

#include "Arches.h"

#include "funscads.h"

#include "Tomar.h"

#define LONG\_REG 64

#define campo\_a\_buffreg(br,cad) strcat(br,cad);strcat(br,DELIM\_CAD);

static char \*solicitud[] = {

"Para terminar de registrar atletas presiona 'enter' cuando te pida la nacionalidad.\n\n"

" Número de participación: ",

" Tiempo último de clasificación: ",

" Nombre del atleta: ",

" Nacionalidad: ",

" Ciudad: ",

" Dirección del atleta: ",

""};/\* Cadena nula para terminar el ciclo de solicitud \*/

static struct{

short cont\_reg;

char relleno[30];

}encabezado;

static menu();

static pide\_info(char buffreg[]);

static pide\_nrr();

static muestra();

static lee\_y\_muestra();

static cambio();

int main(){

setlocale(LC\_CTYPE,"Spanish");//acentos

int i,menu\_elec,nrr;

int byte\_pos;

char ARCHIVO[15];

long lseek();

char buffreg[TAM\_MAX\_REG+1];

/\* MENU INICIAL PARA DAR NOMBRE AL ARCHIVO \*/

printf("\n\t < PRACTICA 4: Registros de longitud fija con encabezado. >\n");

printf("\nIngrese el nombre del archivo a crear: ");

printf("\n\n >> ");

gets(ARCHIVO);

if((fd=open(ARCHIVO,LECTESCRIT))<0)

{

fd=creat(ARCHIVO,PMODE);

encabezado.cont\_reg=0;

write(fd,&encabezado,sizeof(encabezado));

}

else

read(fd,&encabezado,sizeof(encabezado));

while((menu\_elec=menu())<4) {

switch(menu\_elec){

case 1:

system("cls");/\* OPCION 1 PARA REGISTRAR CORREDORES \*/

printf("\n\tSeleccionaste 'Agregar Corredores'\n\n");

pide\_info(buffreg);

byte\_pos=encabezado.cont\_reg \*LONG\_REG+sizeof(encabezado);

lseek(fd,(long)byte\_pos,0);

write(fd,buffreg,LONG\_REG);

encabezado.cont\_reg++;

break;

case 2:

nrr=pide\_nrr();

if(nrr>=encabezado.cont\_reg)

{

system("cls");

printf("El número de registro ingresado es demasiado grande...\n\n");

system("pause");

break;

}

else {

byte\_pos=nrr\*LONG\_REG+sizeof(encabezado);

lseek(fd,(long)byte\_pos,0);

lee\_y\_muestra();

if(cambio()){

printf("\n\nProporcione los valores revisados:\n\n");

pide\_info(buffreg);

lseek(fd,(long)byte\_pos,0);

write(fd,buffreg,LONG\_REG);

}

}

break;

case 3:

nrr=muestra();

if(nrr>=encabezado.cont\_reg)

{

system("cls");

printf("El número de registro ingresado es demasiado grande...\n\n");

system("pause");

}

else {

byte\_pos=nrr\*LONG\_REG+sizeof(encabezado);

lseek(fd,(long)byte\_pos,0);

lee\_y\_muestra();

system("pause");

break;

}

}

}

lseek(fd,0L,0);

write(fd,&encabezado,sizeof(encabezado));

close(fd);

}

static menu(){

int eleccion;

char respuesta[10];

system("cls");

printf("\n\t\t < Maratón 2022 >\n");

printf("\nMenú de opciones: \n");

printf(" 1. Agregar un corredor\n");

printf(" 2. Buscar registro para actualizarlo\n");

printf(" 3. Mostrar registros de corredores\n");

printf(" 4. salir\n");

printf("\nIngrese su opcion: ");

gets(respuesta);

eleccion=atoi(respuesta);

return(eleccion);

};

static pide\_info(char buffreg[]){

int cont\_campos,i;

char respuesta[50];

char hora[1];

char minutos[2];

char segundos[2];

for(i=0;i<LONG\_REG;buffreg[i++]='\0'); //limpia el buffer del registro

for(i=0; i<2 ;i++)

{

printf("%s", solicitud[i]);

if(i==1)

{

printf("\n\t\t Horas: ");

gets(respuesta);

strcat(buffreg,respuesta);

printf("\t\t Minutos: ");

gets(respuesta);

strcat(buffreg,respuesta);

printf("\t\t Segundos: ");

gets(respuesta);

strcat(buffreg,respuesta);

break;

}

gets(respuesta);

strcat(buffreg,respuesta); //pasar las variables al buffer sin delimitador

}

for(i=2;\*solicitud[i] !='\0';i++)

{

printf("%s",solicitud[i]);

gets(respuesta);

campo\_a\_buffreg(buffreg,respuesta);

}

}

static pide\_nrr()

{

int nrr;

char respuesta[10];

system("cls");/\* OPCION 2 PARA BUSCAR CORREDORES \*/

printf("\n\tSeleccionaste 'Buscar registro para actualizarlo'\n\n");

printf("Digite el NRR del registro a modificar: ");

gets(respuesta);

nrr=atoi(respuesta);

return(nrr);

}

static muestra()

{

int nrr;

char respuesta[10];

system("cls");/\* OPCION 3 PARA MOSTRAR CORREDORES \*/

printf("\n\tSeleccionaste 'Mostar registros para actualizarlos'\n\n");

printf("Digite el NRR del registro a mostrar: ");

gets(respuesta);

nrr=atoi(respuesta);

return(nrr);

}

static lee\_y\_muestra()

{

char buffreg[TAM\_MAX\_REG + 1], campo[TAM\_MAX\_REG + 1];

int pos\_busca, long\_reg;

pos\_busca = 9;

read(fd,buffreg,LONG\_REG);

printf("Registros existentes: \n");

buffreg[LONG\_REG] = '\0';

long\_reg = strlen(buffreg);

int i=2;

int j =0;

printf("\t%s",solicitud[0]," :");

for(j=0; j<4; j++)

{

printf("%c",buffreg[j]);

}

printf("\n\t%s",solicitud[1]," :");

printf("%c%s%c%c%s%c%c",buffreg[4],":",buffreg[5],buffreg[6],":",buffreg[7],buffreg[8]);

printf("\n");

while ((pos\_busca = toma\_campo(campo,buffreg,pos\_busca,long\_reg)) >0)

{

printf("\t%s%s\n",solicitud[i],campo);

++i;

}

}

static cambio(){

char respuesta[10];

printf("\n\nDesea modificar este registro?\n");

printf("Conteste S o N, seguido por <ENTER>: ");

gets(respuesta);

mayusculas(respuesta,respuesta);

return((respuesta[0]=='S') ? 1:0);

}