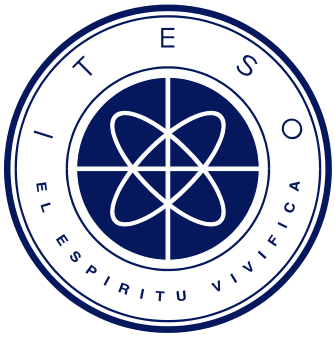
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Departamento de Electrónica, Sistemas e Informática

INGENIERÍA EN SISTEMAS COMPUTACIONALES



PROGRAMACIÓN CON MEMORIA DINÁMICA

TAREA 1. MANEJO DE APUNTADORES

Autor: Franco García Fernando

31 de mayo de 2018. Tlaquepaque, Jalisco,

No se cumplió con los requerimientos funcionales de la tarea.

Todas las figuras e imagenes deben tener un título y utiilizar una leyenda que incluya número de la imagen ó figura y una descripción de la misma. Adicionalmente, debe de existir una referencia a la imagen en el texto.

La documentación de pruebas implica:
1) Descripción del escenario de cada prueba
2) Ejecución de la prueba
3) Descripción y análisis de resultados.

Instrucciónes para entrega de tarea

Es ***IMPRESCINDIBLE*** apegarse a los formatos de entrada y salida que se proveen en el ejemplo y en las instrucciones.

Esta tarea, como el resto, se entregará de la siguiente manera:

- **Reporte:** vía *moodle* en **un archivo PDF**.
- **Código:** vía su repositorio **Github**.

La evaluación de la tarea comprende:

- 10% para la presentación 10 pts
- 60% para la funcionalidad 30 pts
- 30% para las pruebas 20 pts

Es necesario responder el apartado de conclusiones, pero no se trata de llenarlo con paja. Si no se aprendió nada al hacer la práctica, es preferible escribir eso.

Objetivo de la actividad

El objetivo de la tarea es que el alumno aplique los conocimientos y habilidades adquiridos en el tema de apuntadores para la resolución de problemas utilizando el lenguaje ANSI C.

Descripción del problema

Denisse estudia una ingeniería en una universidad de excelencia, donde constantemente invitan a sus estudiantes a evaluar el desempeño académico de los profesores. Cuando Denisse esta inscribiendo asignaturas para su próximo semestre, descubre que tiene diversas opciones con profesores que no conoce, entonces, decide crear un aplicación que le ayude a ella, y a sus compañeros a seleccionar grupos acorde a los resultados de las evaluaciones de los profesores.

Para iniciar, Denisse solicitó apoyo a traves de Facebook para que sus compañeros de toda la Universidad le apoyaran en la asignación de calificaciones de los profesores. Esto en base a sus experiencias previas en los diversos cursos. La respuesta que obtuvo fue 2 listas de profesores evaluados, la primer lista correspondia a profesores que imparten clases en Ingenierías y la segunda contenia a todos los profesores que imparten clases en el resto de las carreras.

Debido a que Denisse, le gusta programar, decidio crear una pequeña aplicación que le permitiera capturar los datos de los profesores y posteriormente le imprimiera una sola lista con todos los profesores ordenados acorde a su calificación. Lamentablemente, debido a que Denisse salio de viaje, no pudo terminar el programa. Tu tarea es ayudar a Denisse para completar el código.

Código escrito por Denisse

Importante: no modificar el código escrito por Denisse, solamente terminar de escribir el código e implementar las funciones.

```
typedef struct{
        char nombre[15];
        float calificacion;
} Profesor;

float averageArray(Profesor ______  , int _____);
void readArray(Profesor ________, int _____);
void mergeArrays(Profesor _____ , int _____, Profesor ____, int _____, Profesor _____, int ____);
void sortArray(Profesor _____, int ____);
void printArray(Profesor ____ , int _____);


void main(){
        Profesor arr1[20];  //Primer arreglo
        Profesor arr2[20];  //Segundo arreglo
        Profesor arrF[40];  //Arreglo final, con elementos fusionados y ordenados
```

```
        int n1, n2; //Longitud de los arreglos

        readArray(_____________); //leer el primer arreglo

        readArray(_____________); //leer el segundo arreglo

        mergeArrays(_____________________________); //Fusionar los dos arreglos en un tercer arreglo

        sortArray(_____________); //Ordenar los elementos del tercer arreglo, recuerde que pueden
                                  //existir profesores repetidos

        printArray(___________); //Imprimir el resultado final

        return 0;
}
```

Descripción de la entrada del programa

El usuario ingresara dos listas con máximo 20 elementos (profesores: nombre y califación). Antes de indicar, uno por uno los datos de los profesores, el usuario debe indicar la cantidad de elementos de la respectiva lista. Así lo primero que introducirá será la cantidad (n1) de elementos de la primer lista (arr1), y en seguida los datos de los profesores de la lista; posteriormente, la cantidad (n2) de elementos de la segunda lista (arr2), seguida por los profesores de los profesores correspondientes.

Ejemplo de entrada:

2
Roberto 7.8
Carlos 8.3

4
Oscar 8.3
Miguel 9.4
Diana 9.5
Oscar 8.5

Descripción de la salida

La salida del programa deberá ser sencillamente la impresión de una lista de profesores y su respectiva calificación (ordenados en orden descendiente, separados por un salto de línea). ¿Qué sucede si tenemos dos o más veces el registro de un profesor? La lista final, deberá mostrar sólo una vez a ese profesor y el promedio de sus calificaciones.

Ejemplo de la salida:

Diana 9.5
Miguel 9.4
Oscar 8.4
Carlos 8.3
Roberto 7.8

SOLUCIÓN DEL ALUMNO, PRUEBAS Y CONCLUSIONES

Código fuente:

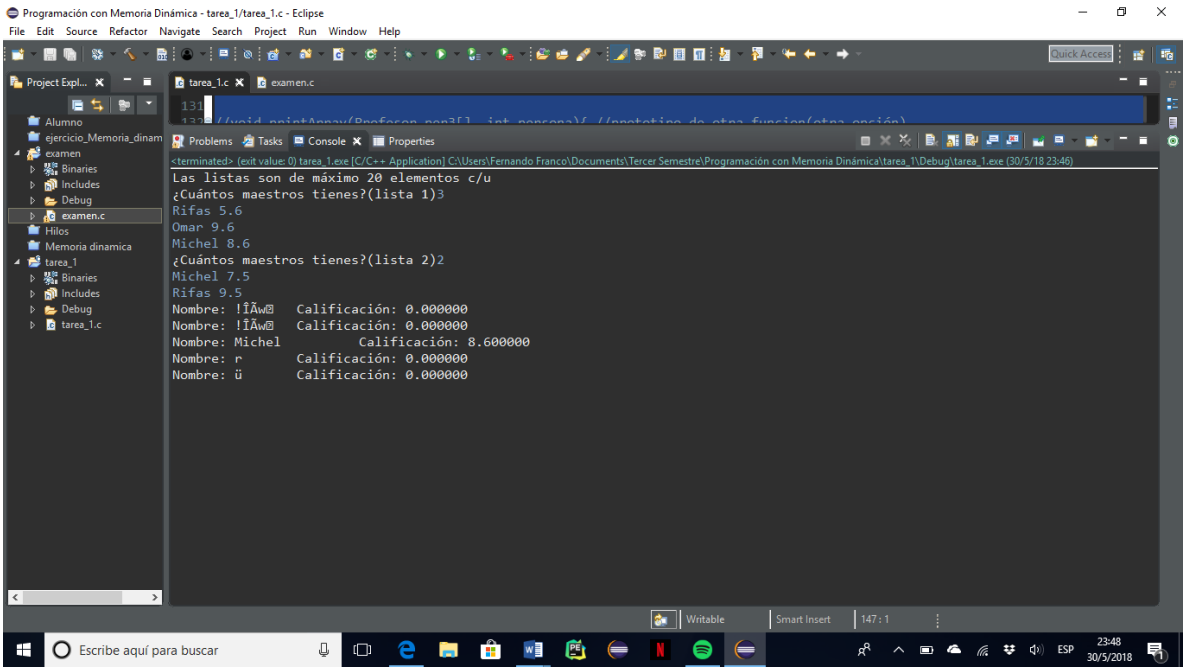
```
/*
 * tarea_1.c
 *
 *  Created on: 24 may. 2018
 *      Author: Fernando Franco
 */
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

typedef struct{
	char nombre[15];
	float calificacion;
} Profesor;
float averageArray(Profesor per3[], int persona3);
void readArray(Profesor per[], int persona);
void sortArray(Profesor per3[], int persona3);
void mergeArrays(Profesor per[] , int persona, Profesor per2[],
int persona2, Profesor per3[], int persona3);
void printArray(Profesor per3[], int persona3);


int main(){
	setvbuf(stderr, NULL, _IONBF, 0);
	setvbuf(stdout, NULL, _IONBF, 0);
	printf("Las listas son de máximo 20 elementos c/u\n");
	Profesor per[20];
	Profesor per2[20];
	Profesor per3[40];
	int n, n1;
	printf("¿Cuántos maestros tienes?(lista 1)");
	scanf("%d",&n);
```

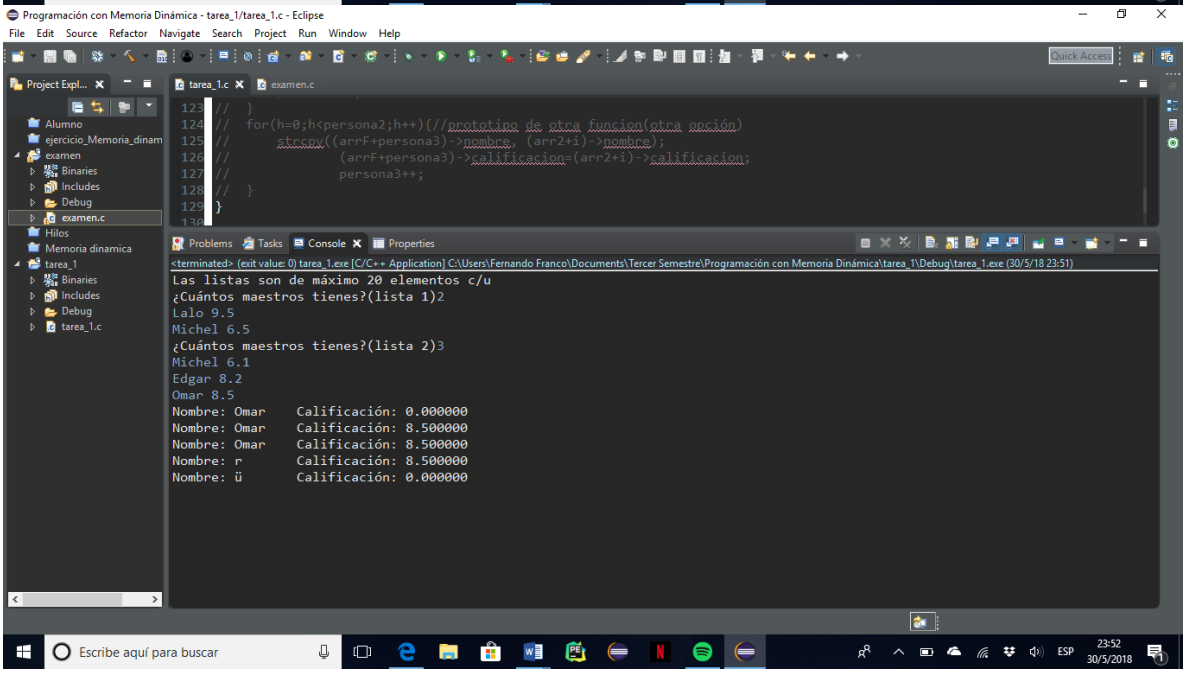
```
	fflush(stdin);
	readArray(per, n);
	printf("¿Cuántos maestros tienes?(lista 2)");
	scanf("%d",&n1);
	fflush(stdin);
	readArray(per2,n1);
	int i=n+n1;
	mergeArrays(per, n, per2,n1, per3, i);
	averageArray(per3, i);
	sortArray(per3, i);
	printArray(per3, i);

	return 0;
}
float averageArray(Profesor per3[]  , int persona3){
	int i, j;
	int av;
	Profesor *pointer=per3;
	int acum1, repetidos;
	for(i=0; i<persona3;i++){
		acum1=0;
		repetidos=1;
		for(j=0;j<persona3;j++){

	if(strcmp((*(pointer)).nombre,(*(pointer+j)).nombre)==0){

	acum1=(*(pointer)).calificacion+(*(pointer+j)).calificacion;
				repetidos++;
			}
		}
		av=acum1/repetidos;
	}
	return av;
}

void readArray(Profesor per[], int persona){
	Profesor *p=per;
	for (int i=0; i<persona; i++){
		//scanf("%[^\n] %f",*(p+i+1).nombre,
&(*(p+i+1)).calificacion );
		scanf("%s[^\n] ", (*(p+i)).nombre);
		scanf(" %f", &(*(p+i)).calificacion);
	}
}
//void readArray(Profesor per[20], int persona){//prototipo de
otra funcion(otra opción)
```

```
//    for( int i =0; i<persona;i++){
//         printf("¿Cuál es el nombre del profesor?¿Cuál es su
calificación?\n");
//         gets(per[i].nombre);
//         fflush(stdin);
//         scanf(" %f", &per[i].calificacion);
//    }
//}

//void sortArray(Profesor per[20] , int i){//prototipo de otra
funcion(otra opción)
//    Profesor sort;
//         for(int x=0; x<i;x++){
//              if(per[x].calificacion>per[x+1].calificacion){
//                   sort=per[x];
//                   per[x]=per[x+1];
//                   per[x+1]=sort;
//              }
//         }
//}
void sortArray(Profesor per3[40], int persona3){
     Profesor *sort=per3;
     float x;
     int i;
     for (i=0; i<persona3;i++){
          if(sort->calificacion>(sort+i+1)->calificacion){
               x=sort->calificacion;
               sort->calificacion=(sort+i+1)->calificacion;
               (sort+i+1)->calificacion=x;
          }
     }
}

void mergeArrays(Profesor per[] , int persona, Profesor per2[],
int persona2, Profesor per3[], int persona3){
     Profesor *arr=per;
     Profesor *arr2=per2;
     Profesor *arrF=per3;
     int i, h;
     //persona3=0;
     for(i=0; i<persona;i++){
          strcpy((arrF+i)->nombre, (arr+i)->nombre);
          (arrF+i)->calificacion=(arr+i)->calificacion;
     }
     for(h=0;h<persona2;h++){
          strcpy((arrF+h)->nombre, (arr2+i)->nombre);
```

```
            (arrF+h)->calificacion=(arr2+i)->calificacion;
      }
//    for (i=0;i<persona;i++){//prototipo de otra funcion(otra
opción)
//          strcpy((arrF+persona3)->nombre, (arr+i)->nombre);
//          (arrF+persona3)->calificacion=(arr+i)->calificacion;
//          persona3++;
//    }
//    for(h=0;h<persona2;h++){//prototipo de otra funcion(otra
opción)
//          strcpy((arrF+persona3)->nombre, (arr2+i)->nombre);
//                      (arrF+persona3)->calificacion=(arr2+i)-
>calificacion;
//                      persona3++;
//    }
}


//void printArray(Profesor per3[], int persona){ //prototipo de
otra funcion(otra opción)
//    int i;
//    for(i=0; i<persona;i++){
//          printf("Nombre: %s Calificación: %f\n", per3[i].nombre,
per3[i].calificacion);
//
//    }
//}

void printArray(Profesor per3[], int persona3){
      Profesor *p=per3;
      for(int i=0; i<persona3;i++){
            printf("Nombre: %s  \t",(*(p+i)).nombre);
            printf("Calificación: %f\n",(*(p+i)).calificacion );
      }
}
}
```

Ejecución:





Conclusiones (obligatorio):

- ✓ Lo que aprendí con esta práctica. Lo que ya sabía.
- ✓ Lo que me costó trabajo y cómo lo solucioné.
- ✓ Lo que no pude solucionar.

Lo que aprendí con esta tarea es hacer un mejor uso de los apuntadores, nunca antes había trabajado con apuntadores más que para la apertura de archivos, pero aun así no fue de una manera tan precisa como ahora, ciertamente tuve muchos problemas, puesto que mi código sigue estando

mal, aun no estoy muy seguro de cómo utilizar de manera correcta los apuntadores, puesto que su sintaxis cambia conforme a quieras utilizarlo, y aun que me las sé, no sé cuándo con exactitud se deba usar cada ya que me confunden bastante, debo de repasar más eso.

Realmente todo el código me costó trabajo al no saber si lo estaba haciendo de manera correcta.

Hay varias cosas que no pude solucionar, pero lo que más me causo inquietud fue el print, ya que a la hora de imprimir la lista final no me la imprimía de manera correcta (como se muestra en los screenshots), revise varias veces esas partes y ciertamente no logre encontrar el error e intente también solo usando las funciones de printArray, readArray y aun así se imprimía mal, no logre encontrar el error, así realmente no podía cerciorarme de que otras partes tuvieran un gran error ya que no podía ver como se imprimía, incluso pregunte a compañeros y tampoco lograron encontrarlo.