

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 APUNTADES

Autor: Ruvalcaba, Barboza Daniel

Funcionalidad: 30 pts
Presentación: 5 pts
Pruebas: 20 pts

31 de mayo de 2018. Tlaquepaque, Jalisco,

No se cumplió con los requerimientos funcionales de la tarea.

Todas las figuras e imágenes deben tener un título y utilizar 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.

SOLUCIÓN DEL ALUMNO, PRUEBAS Y CONCLUSIONES

Código fuente:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

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

Profesor *borrar=NULL;

float averageArray(Profesor *array, int n);

void readArray(Profesor *array, int n);

void mergeArrays(Profesor *arr1, int n1, Profesor *arr2, int n2, Profesor *arrF,
int n3);

void sortArray(Profesor *array, int n3);
void printArray(Profesor *array, int n3);

void main(){
    setbuf(stdout,NULL);
    int n1, n2, n3;
    Profesor arr1[20];
    Profesor arr2[20];
    Profesor arrF[40];
    printf("Escribe el numero de profesores que ingresarás: ");
    scanf("%d",&n1);
    readArray(arr1,n1);
    printf("Excribe el numero de profesor que ingresarás: ");
    scanf("%d",&n2);
    readArray(arr2,n2);
    n3=n1+n2;
    mergeArrays(arr1,n1,arr2,n2,arrF,n3);
    sortArray(arrF,n3);
    printArray(arrF,n3);
}

void readArray(Profesor *array, int n)
{
    Profesor *v;
    Profesor *adios,*adios2;
    v=array;
    int o=1,h;
    for(int i=0;i<n;i++)
    {
        scanf("%s %f",&v[i].nombre,&v[i].calificacion);
    }
}
```

```

//     for(int i=0;i<n;i++)
//     {
//         if(v[i].nombre==v[i+1].nombre)
//         {
//             o++;
//             h+=v[i+1].calificacion;
//             v[i+1]=adios;
//         }
//
//         if(o>1){
//             v[i].calificacion=((v[i].calificacion)+h)/o;
//         }
//         o=1;
//         h=0;
//     }
// }

```

```

void mergeArrays(Profesor *arr1, int n1, Profesor *arr2, int n2, Profesor *arrF,
int n3)

```

```

{
    Profesor *parrF;
    Profesor *parr1;
    Profesor *parr2;
    parrF=arrF;
    parr1=arr1;
    parr2=arr2;
    for(int i=0;i<n1;i++)
    {
        parrF[i]=parr1[i];
    }
    for(int i=0;i<n2;i++)
    {
        parrF[n1+i]=parr2[i];
    }
}

```

```

void sortArray(Profesor *array, int n3){
    Profesor *pnombre;
    Profesor *aux=NULL;
    pnombre=array;
    Profesor swap;
    for (int i = 0 ; i < n3 - 1; i++)
    {
        for (int j = 0 ; j < n3 - i - 1; j++)
        {
            if (array[j].calificacion < array[j+1].calificacion)
            {
                swap=array[j];
                array[j]=array[j+1];
                array[j+1]=swap;
            }
        }
    }
}

```

```

}

void printArray(Profesor *array, int n3){
    printf("RESULTADOS\n");
    for(int i=0;i<n3;i++)
    {
        printf("%s %.1f\n",array[i].nombre,array[i].calificacion);}
}

float averageArray(Profesor *array, int n){

    return n;
}

```

Ejecución:

Primer arreglo mayor:

```

<terminated> (exit value: 10) d:\didi.exe [C/C++ Application] C:\User
Escribe el numero de profesores que ingresarás: 2
Juan 10
Pepe 4
Escribe el numero de profesor que ingresarás: 2
Laura 0
Jimena 3
RESULTADOS
Juan 10.0
Pepe 4.0
Jimena 3.0
Laura 0.0

```

Segundo arreglo mayor:

```

Escribe el numero de profesores que ingresarás: 2
Paco 4
Ronaldo 6.7
Escribe el numero de profesor que ingresarás: 2
Daniel 8.9
Eugenia 7
RESULTADOS
Daniel 8.9
Eugenia 7.0
Ronaldo 6.7
Paco 4.0

```

Conclusiones (obligatorio):

Pude empezar a darme que cuenta sobre cómo podemos manipular los apuntadores, por ejemplo, en esta práctica lo que utilicé mucho es crear apuntadores que apuntaban a arreglos, para de esa manera usarlos como arreglos. También en los pases por referencia de las funciones, me di cuenta de que aunque tengamos definido un tipo de dato, podemos pasarlo por un función, siempre y cuando se trate de una variable o de un arreglo, con su respectiva sintaxis.

Algo que se me complicó mucho fue que no supe como hacer para que cuando tuviésemos un nombre repetido, eliminar los necesarios para que solo quedara un solo nombre, y sacarle su promedio, estuve intentando mucho y buscando en internet, pero la verdad no supe cómo hacerlo.

En todo lo demás, creo que pude ejercitar y darme cuenta sobre el uso de los apuntadores, y de cómo es su sintaxis.

También aprendí, aunque no va mucho con el tema, pero en las investigaciones que estuve haciendo para realizar el programa, pude aprender que utilizando un .1 antes del %f, podía decidir el numero de decimales que quería que salieran.

Pude entender mejor los tipos de datos estructurados, ya que los vi el semestre pasado pero la verdad no me había quedado muy claros, ahora sé que podemos utilizar un apuntador para obtener la dirección de memoria de una variable con un tipo de dato estructurado de varias maneras, pero una de las que más se me facilitaron fue la de ->.

Espero ir mejorando con la práctica, y que, si ahora no pude resolver satisfactoriamente esta tarea, espero que, en un futuro, con las demás habilidades que vaya adquiriendo, pueda resolverlo sin ningún problema.