|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Garcia Morales Karina |
| *Asignatura:* | Fundamentos de Programación |
| *Grupo:* | 22 |
| *No de Práctica(s):* | Practica 11 |
| *Integrante(s):* | Ramos Orozco Daniel |
|  |  |
|  |  |
| *Semestre:* | 2018-2 |
| *Fecha de entrega:* | 08-05-2018 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Objetivo:**

Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

**¿Qué es un arreglo?**

**R:** Es una secuencia de datos del mismo tipo, que se enumeran sucesivamente 0,1,2,3…etc.

**¿Cuál es la dimensión del arreglo?**

**R:** Unidimensionales o multidimensionales.

**¿Cómo miden la dimensión del arreglo?**

**R:** Para dar respuesta podemos poner unos ejemplos:

Los unidimensionales se miden dando un número entre “[]” y después se le coloca los valores en “{,}”.

Los multidimensionales se miden dando un número entre “[][]” y después se le colocan los valores en “{{,},{,},…}”.

**¿Para qué utiliza un arreglo?**

**R:** Se utilizan para hacer más eficiente el código de un programa.

**¿Cuál es la diferencia entre un arreglo multidimensional y un arreglo unidimensional?**

**R:** la diferencia entre ellas sería sus dimensiones, ya que la unidimensional solo cuenta con una fila de “n” números o caracteres y la multidimensional cuanta con “n” columnas por “n” filas de “n” números o caracteres.

**¿Qué es un apuntador?**

**R:** Un apuntador es una variable que contiene la dirección de una variable, que hace referencia a la localidad de memoria de dicha variable.

**Y defínelo:**

**R:** int \*a;

int b=5;

a=&b;

**Ejercicios vistos**

Primero vimos un arreglo unidimensional con el siclo while que es el siguiente:

#include <stdio.h>

/\*

Este programa genera un arreglo unidimensional de 5 elementos y los

accede a cada elemento del arreglo a través de un ciclo while.

\*/

int main (){

#define TAMANO 5

int lista[TAMANO] = {10, 8, 5, 8, 7};

int indice = 0;

printf("\tLista\n");

while (indice < 5 ){

printf("\nCalificación del alumno %d es %d", indice+1, lista[indice]);

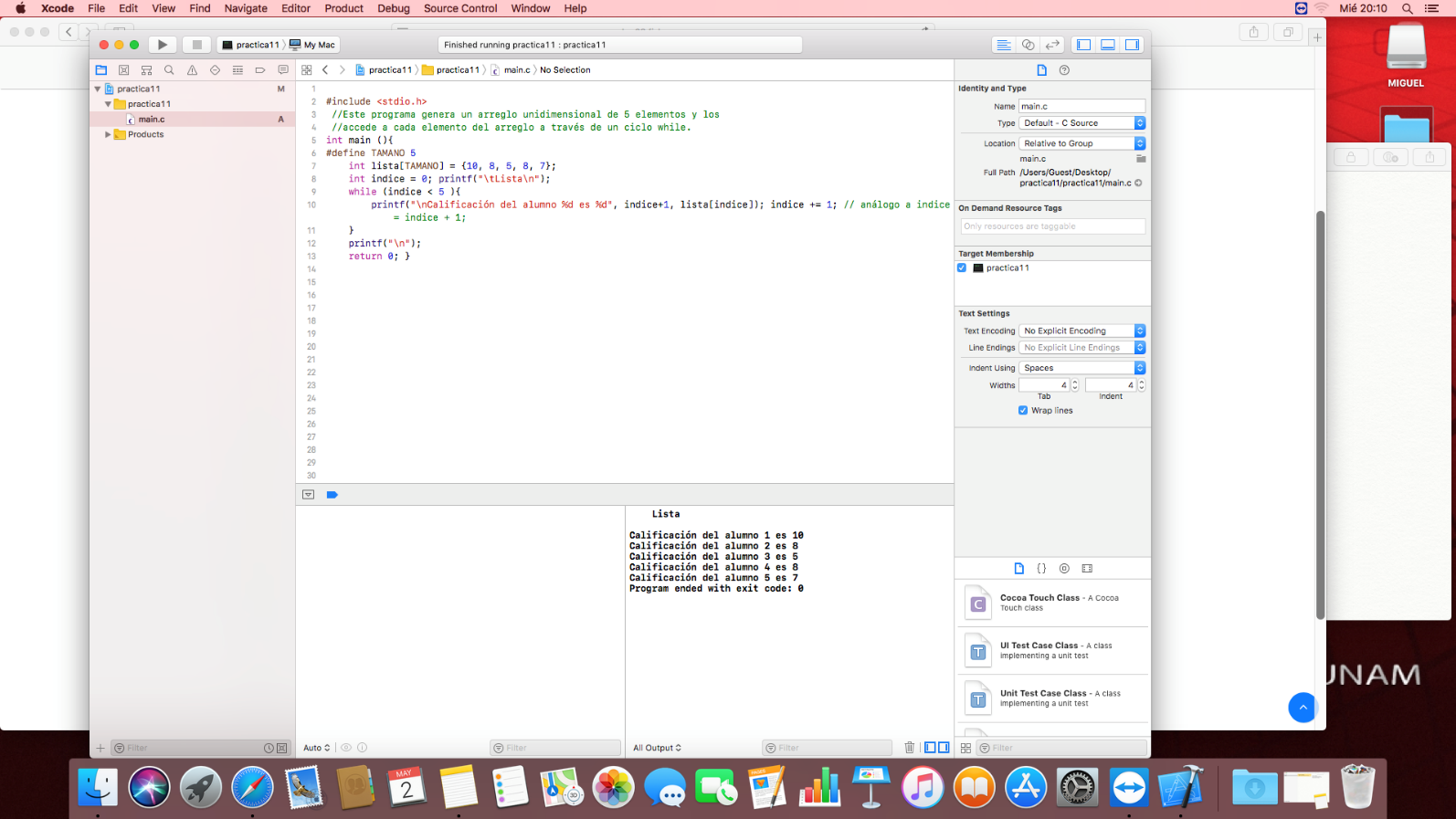
indice += 1; // análogo a indice = indice + 1;

}

printf("\n");

return 0;

}



En donde con un simple siclo while podemos hacer que nos de cada uno de los valores contenidos en un arreglo de manera ordenada

Después vimos un arreglo unidimensional pero ahora con el siclo for:

#include <stdio.h>

/\*

Este programa genera un arreglo unidimensional de 5 elementos y

accede a cada elemento del arreglo a través de un ciclo for.

\*/

int main (){

#define TAMANO 5

int lista[TAMANO] = {10, 8, 5, 8, 7};

printf("\tLista\n");

for (int indice = 0 ; indice < 5 ; indice++){

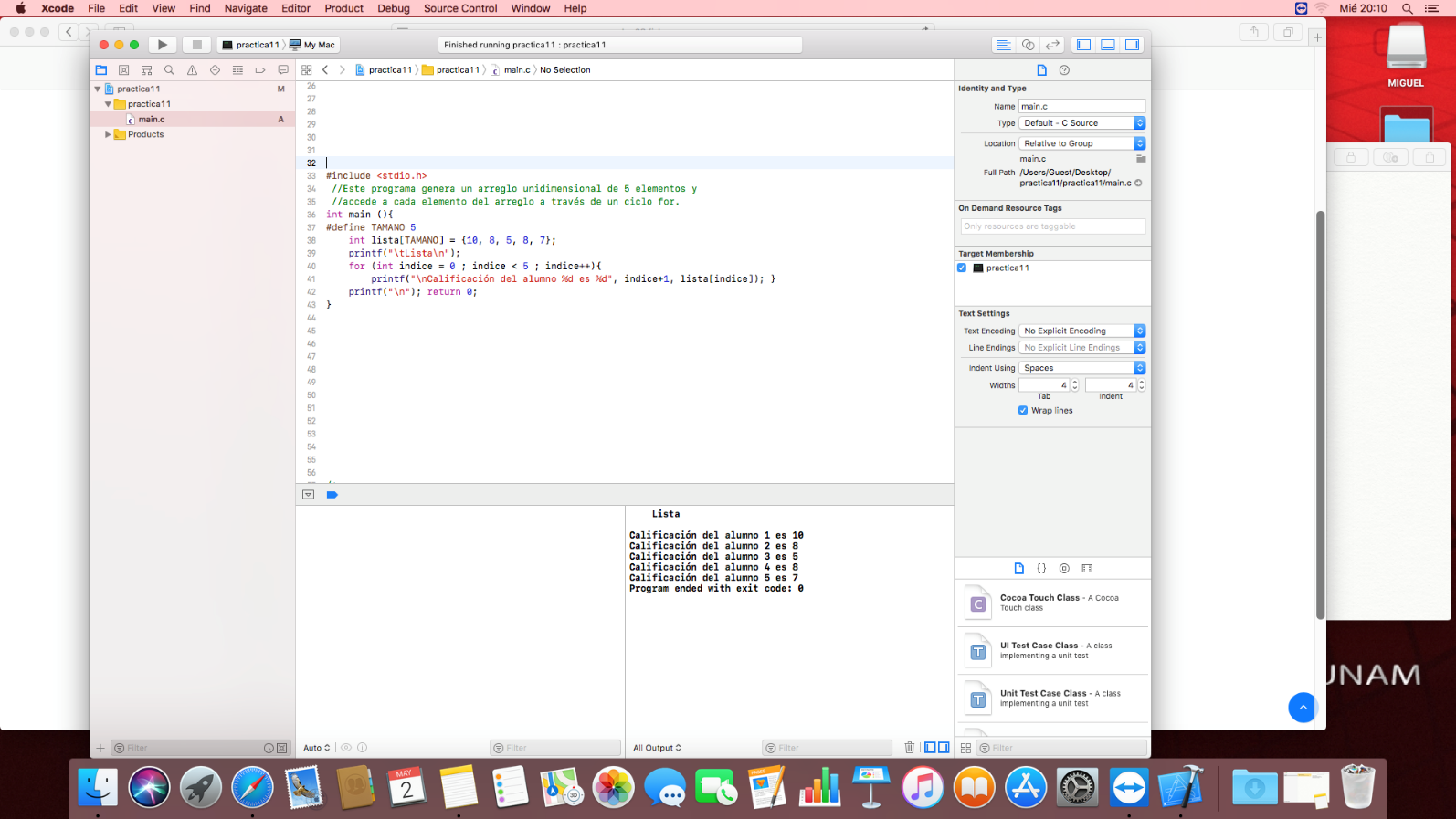
printf("\nCalificación del alumno %d es %d", indice+1, lista[indice]);

}

printf("\n");

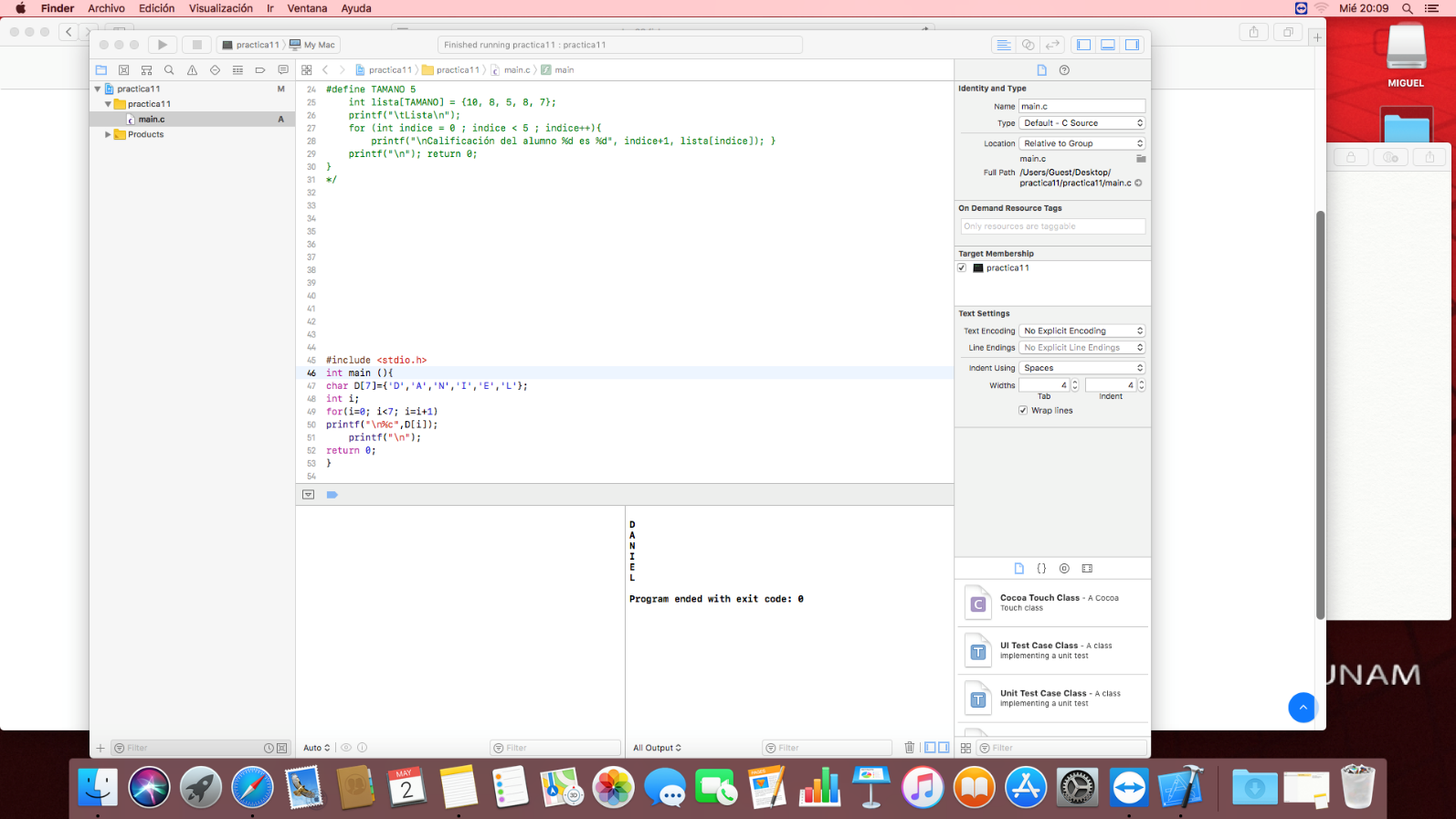
return 0;

}



En donde de igual manera, pero con un ciclo for podemos hacer que nos de los valores contenidos en un arreglo y de manera ordenada.

Después se nos pidió hacer un programa con un ciclo for pero que en vez que nos diera las calificaciones de cada alumno, nos diera nuestro nombre:



Y como se muestra nos da nuestro nombre con un comando algo simple de for.

Después vimos un ejemplo con apuntadores donde lo definimos:

#include <stdio.h>

/\*

Este programa crea un apuntador de tipo carácter.

\*/

int main () {

char \*ap, c = 'a';

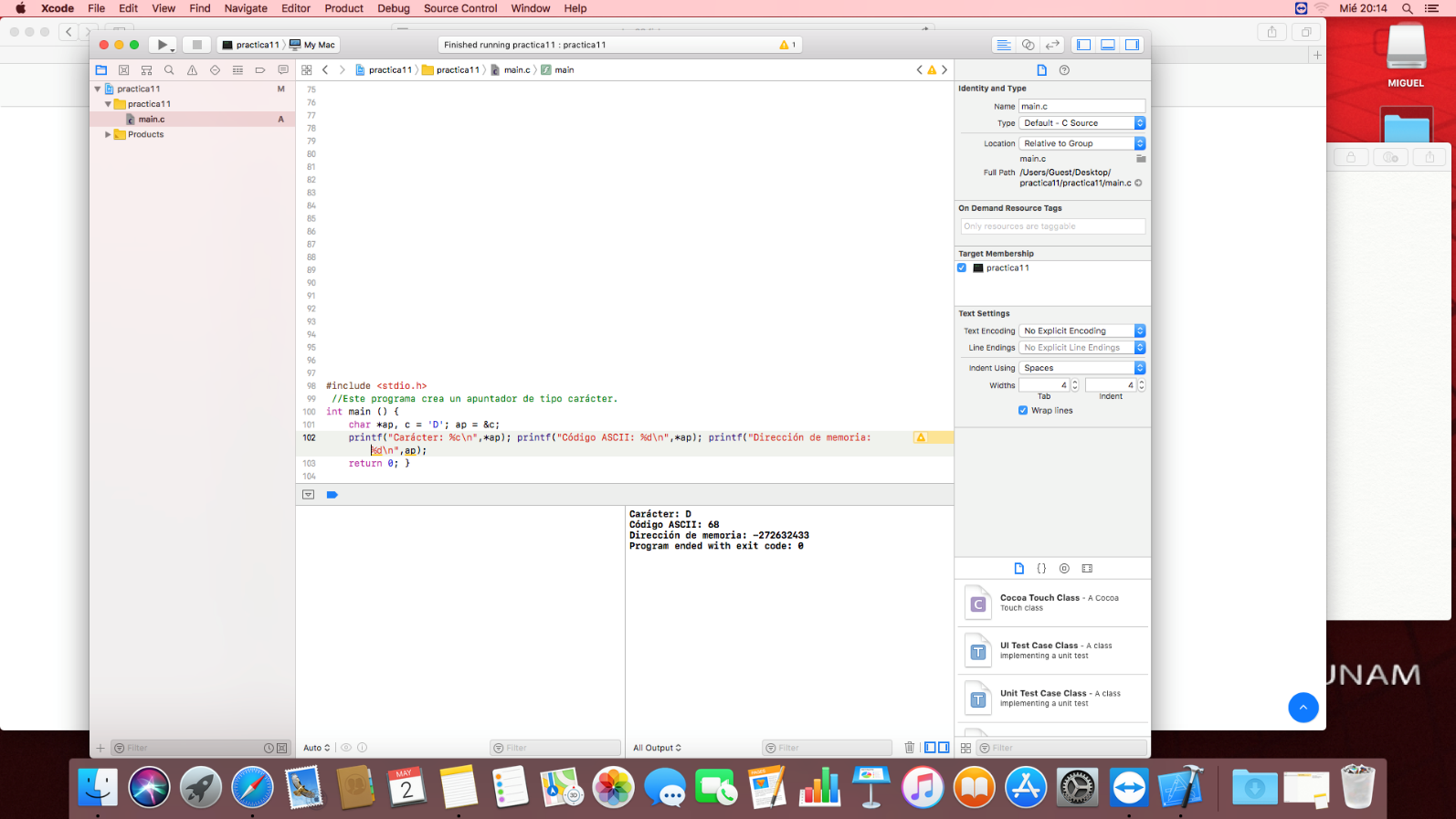
ap = &c;

printf("Carácter: %c\n",\*ap);

printf("Código ASCII: %d\n",\*ap);

printf("Dirección de memoria: %d\n",ap);

return 0;

}

En este caso al copiarlo nos dio un orden distinto pero, esto no afecto el proceso del programa; después lo que hicimos en cambiar la “C” por una “D” y así hacer que nos de el valor en código ASCII y la dirección de memoria en donde se guardo.

Después vimos un ejercicio combinando apuntadores con arreglos:

#include<stdio.h>

/\*

Este programa accede a las localidades de memoria de distintas variables a

través de un apuntador.

\*/

int main () {

int a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0};

int \*apEnt;

apEnt = &a;

printf("a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0}\n");

printf("apEnt = &a\n");

b = \*apEnt;

printf("b = \*apEnt \t-> b = %i\n", b);

b = \*apEnt +1;

printf("b = \*apEnt + 1 \t-> b = %i\n", b);

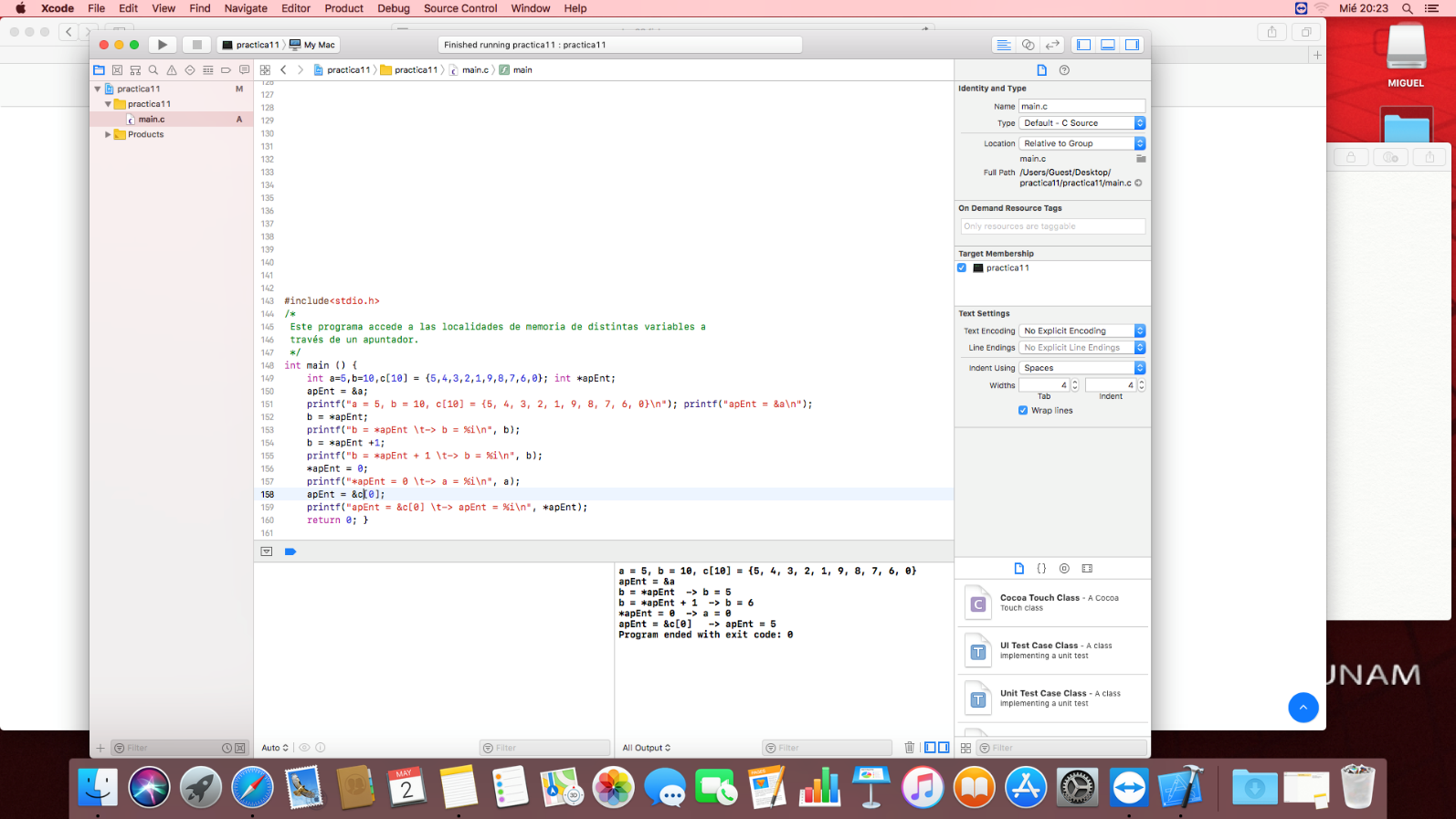
\*apEnt = 0;

printf("\*apEnt = 0 \t-> a = %i\n", a);

apEnt = &c[0];

printf("apEnt = &c[0] \t-> apEnt = %i\n", \*apEnt);

return 0;

}

En este código se hacen ciertas asignaciones de apuntadores con ayuda del arreglo que se da al inicio.

Después vimos el siguiente código:

#include <stdio.h>

/\*

Este programa trabaja con aritmética de apuntadores para acceder a los

valores de un arreglo.

\*/

int main () {

int arr[] = {5, 4, 3, 2, 1};

int \*apArr;

apArr = arr;

printf("int arr[] = {5, 4, 3, 2, 1};\n");

printf("apArr = &arr[0]\n");

int x = \*apArr;

printf("x = \*apArr \t -> x = %d\n", x);

x = \*(apArr+1);

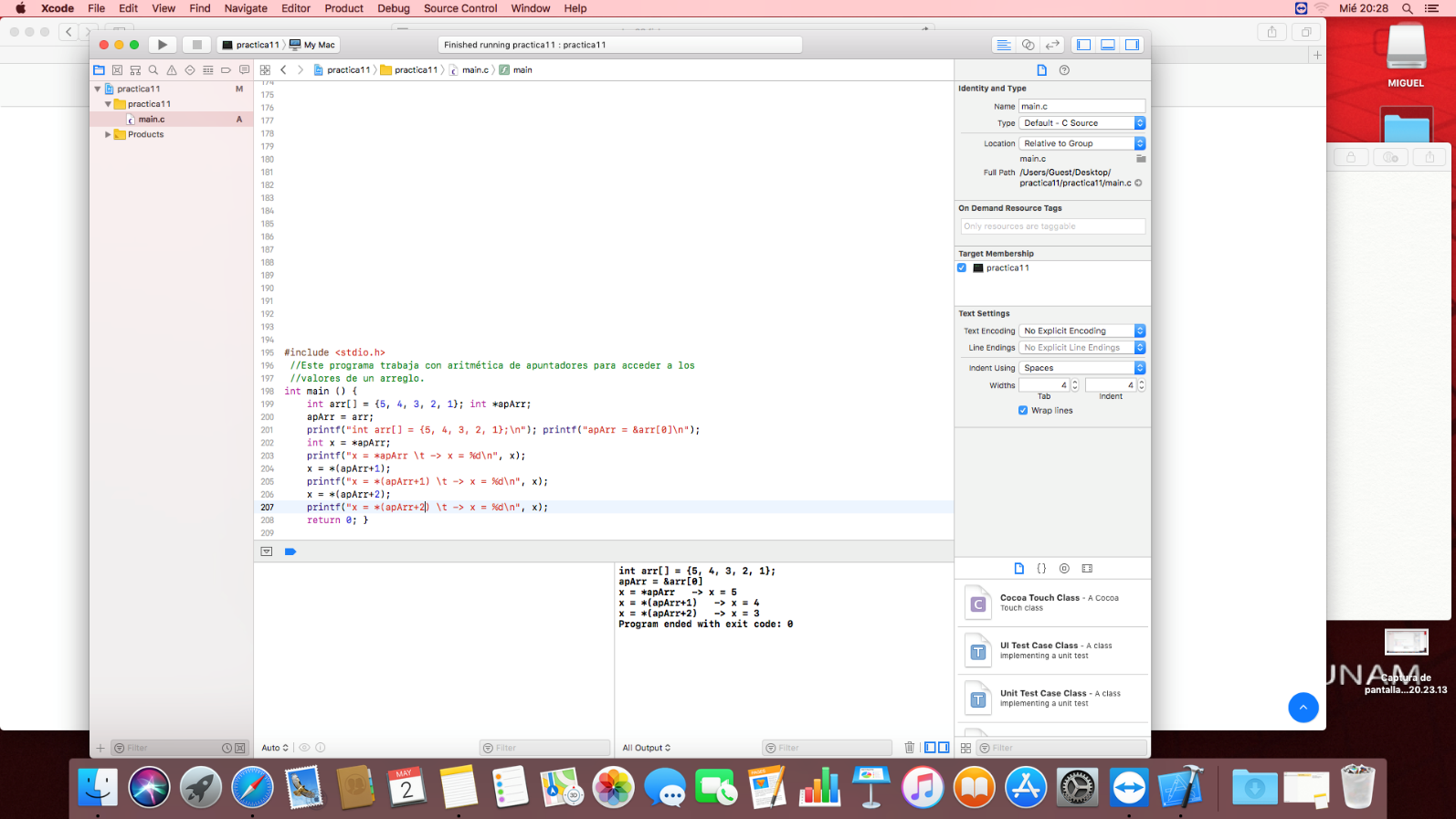
printf("x = \*(apArr+1) \t -> x = %d\n", x);

x = \*(apArr+2);

printf("x = \*(apArr+1) \t -> x = %d\n", x);

return 0;

}



En este código se hace casi lo mismo que en el anterior que a base de asignaciones da un valor distinto solo que en este no se dio la dimensión del arreglo y al principio se le asignó una posición del arreglo.

Ahora vimos el siguiente código de apuntadores con ciclo for:

#include <stdio.h>

/\*

Este programa genera un arreglo unidimensional de 5 elementos y

accede a cada elemento del arreglo a través de un apuntador

utilizando un ciclo for.

\*/

int main (){

#define TAMANO 5

int lista[TAMANO] = {10, 8, 5, 8, 7};

int \*ap = lista;

printf("\tLista\n");

for (int indice = 0 ; indice < 5 ; indice++){

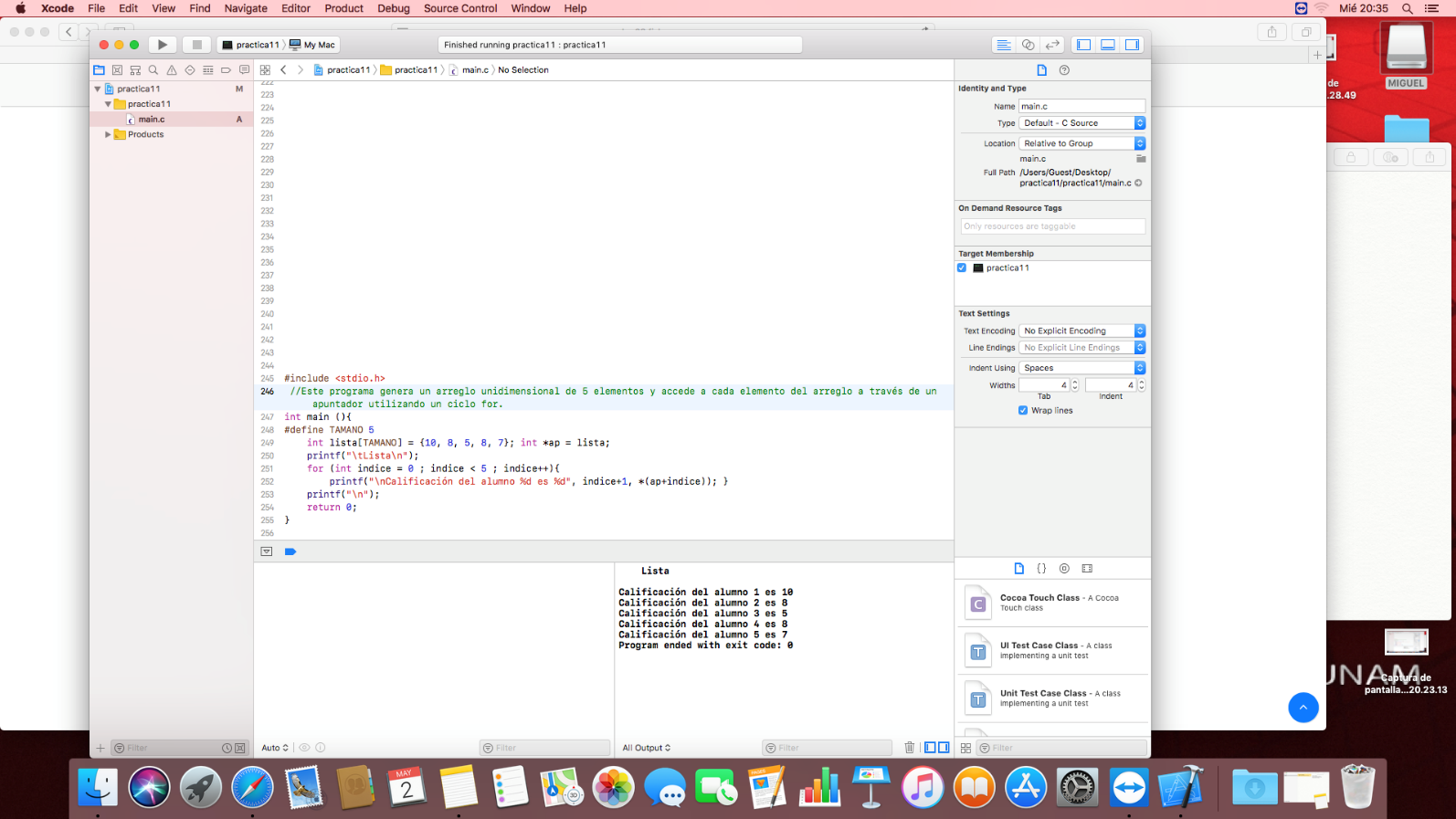
printf("\nCalificación del alumno %d es %d", indice+1, \*(ap+indice));

}

printf("\n");

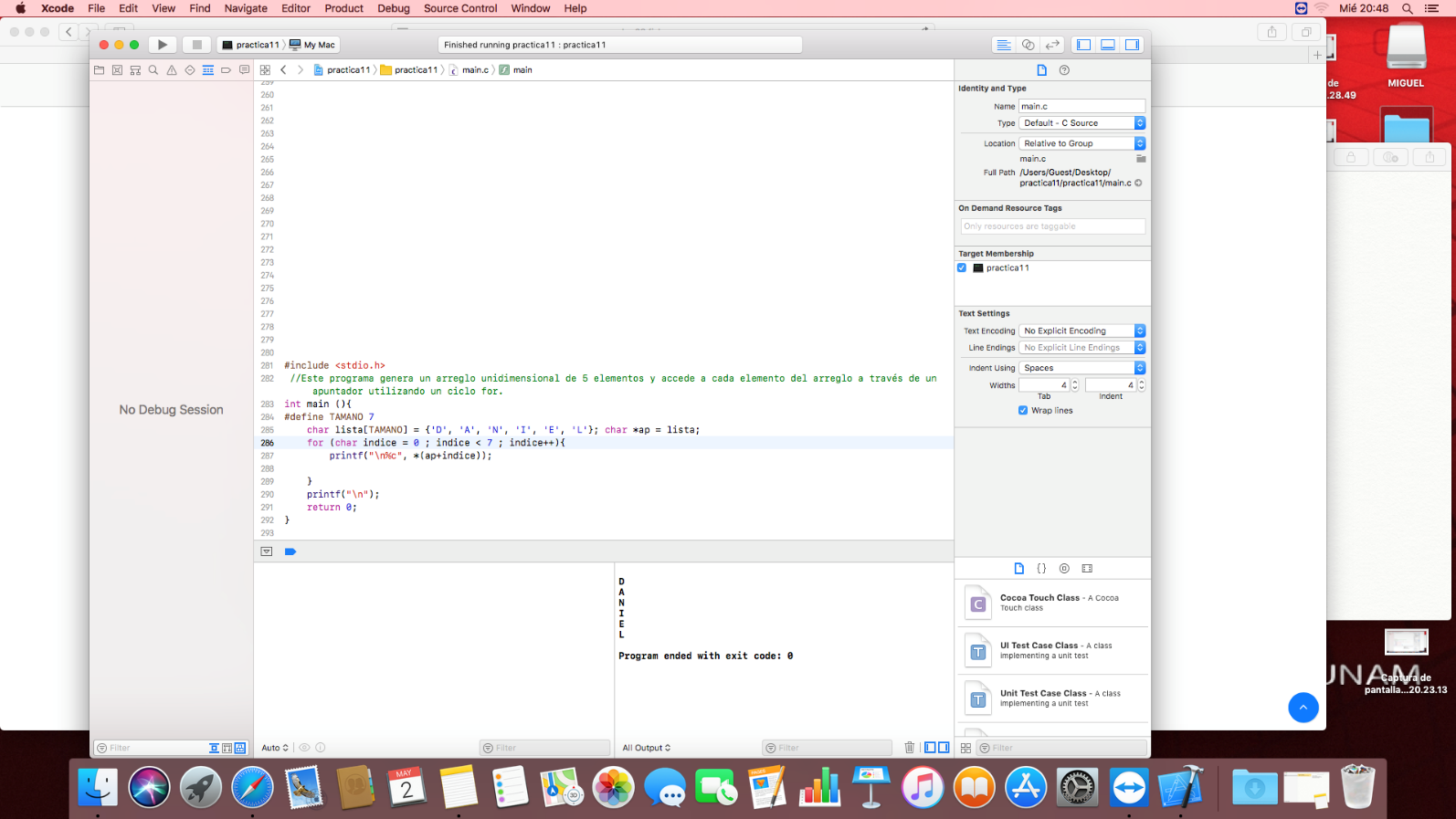
return 0;

}



Ahora e este código lo que se hace es que con ayudas de los apuntadores y un ciclo for, nos da cada uno de los valores del arreglo y con el ciclo for los coloca en diferentes alumnos y por el for tambien hace el conteo de cada alumno para poder colocarlos en orden correcta.

Después utilizamos un código similar ya que se nos pidió hacer lo mismo pero ahora dando nuestro nombre para que apareciera:



Ahora con ayuda de arreglos multimencionales:

#include<stdio.h>

/\* Este programa genera un arreglo de dos dimensiones (arreglo

multidimensional) y accede a sus elementos a través de dos ciclos

for, uno anidado dentro de otro.

\*/

int main(){

int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};

int i, j;

printf("Imprimir Matriz\n");

for (i=0 ; i<3 ; i++){

for (j=0 ; j<3 ; j++){

printf("%d, ",matriz[i][j]);

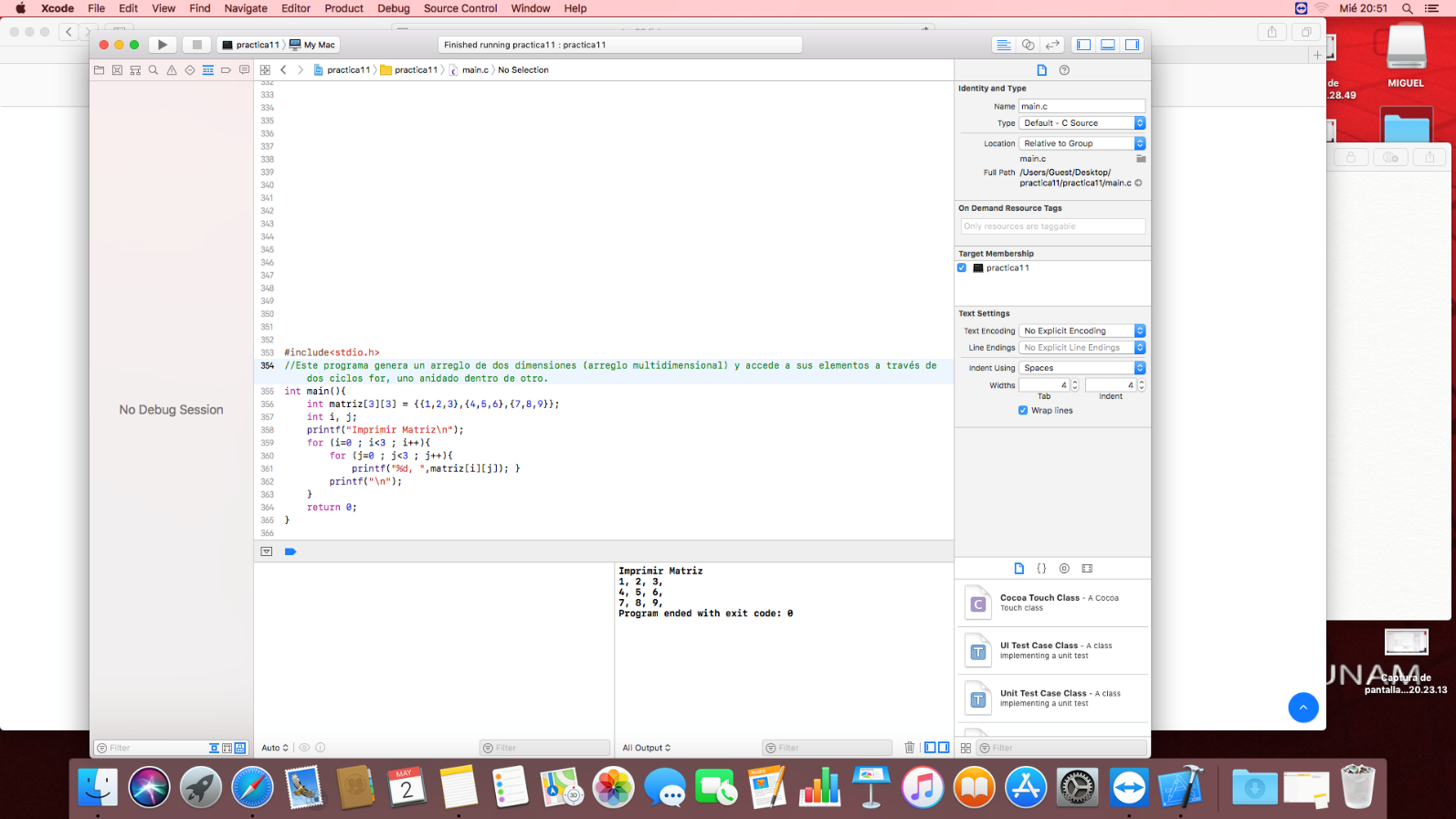
}

printf("\n");

}

return 0;

}



Y lo que se nos da es el valor del arreglo multidimensional en orden exacta.

Ahora vimos un ejemplo de apuntadores con arreglos multidimensionales:

#include<stdio.h>

/\* Este programa genera un arreglo de dos dimensiones (arreglo

multidimensional) y accede a sus elementos a través de un apuntador utilizando

un ciclo for.

\*/

int main(){

int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};

int i, cont=0, \*ap;

ap = matriz;

printf("Imprimir Matriz\n");

for (i=0 ; i<9 ; i++){

if (cont == 3){

printf("\n");

cont = 0;

}

printf("%d\t",\*(ap+i));

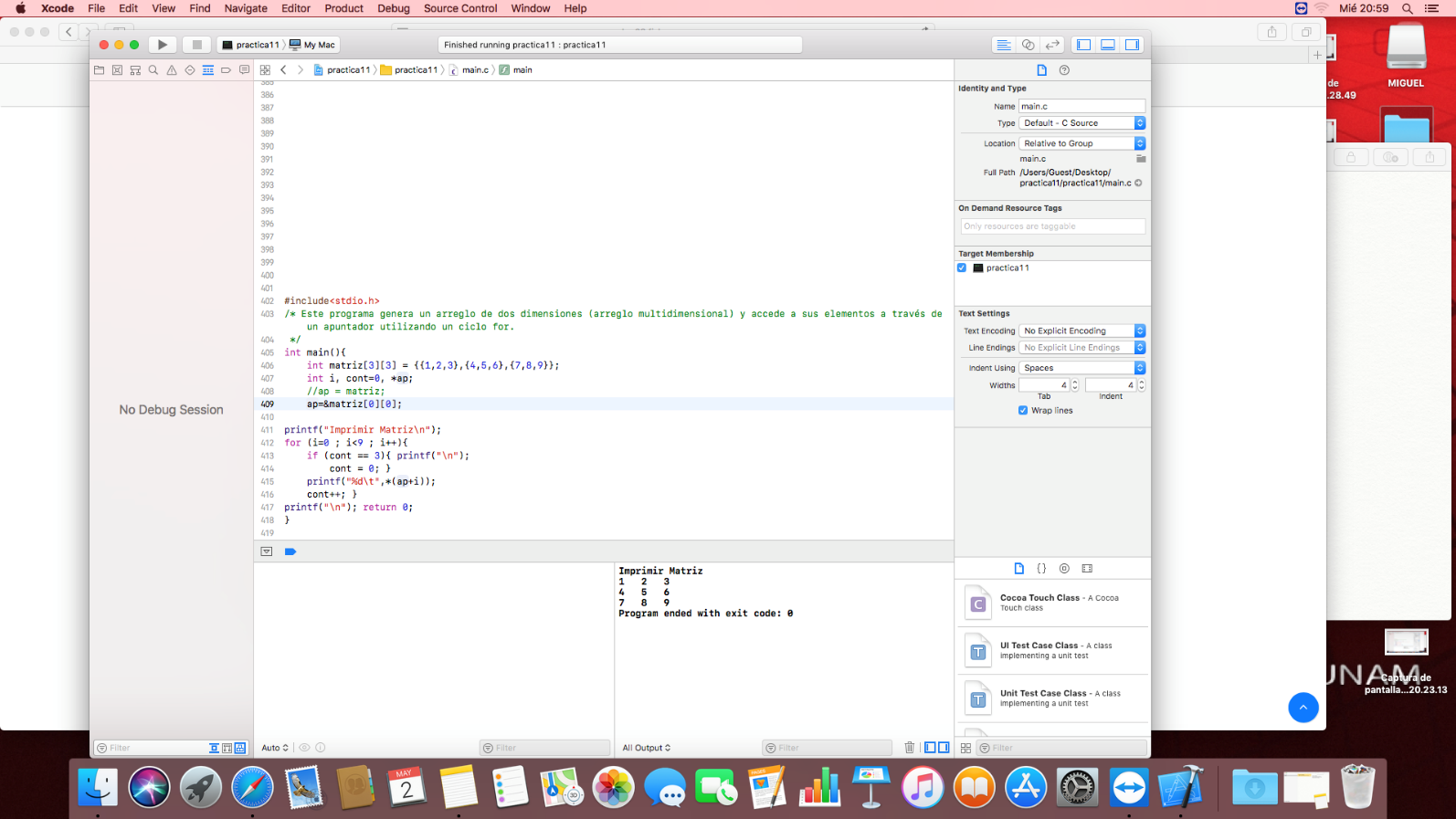
cont++;

}

printf("\n");

return 0;

}



Ahora gracias a la asignación de la matriz con al apuntador y el siclo for se pudo dar la matriz de manera correcta

# Ejercicios propuestos para realizar de tarea

## Ejercicio arreglo unidimensional:

Indica que realiza el siguiente programa:

#include <stdio.h>

int arreglo[] = {20,19,12,17,16,36,14,13,23,31};

int i, j, n, aux;

main() {

n = 10;

for(i=1; i<n; i++) {

j = i;

aux = arreglo[i];

while(j>0 && aux<arreglo[j-1]) {

arreglo[j] = arreglo[j-1];

j=j-1;

}

arreglo[j] = aux;

}

printf("\n\nLos elementos obtenidos del arreglo son: \n");

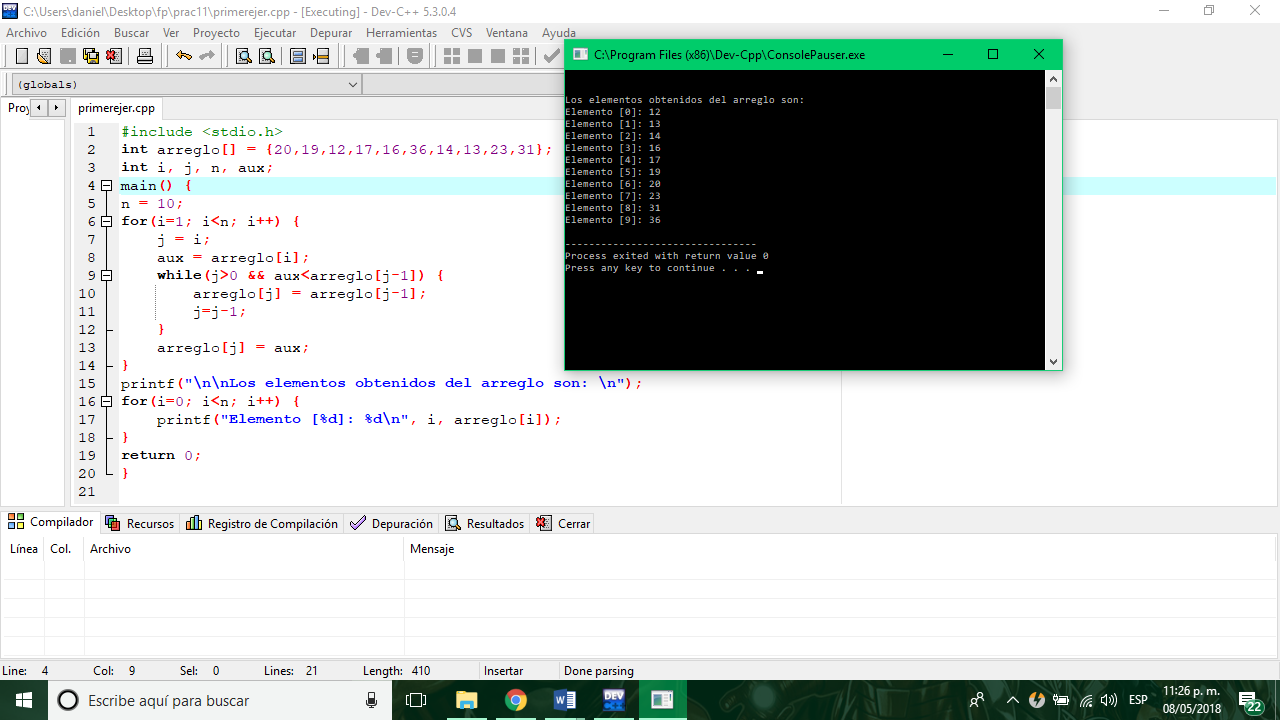
for(i=0; i<n; i++) {

printf("Elemento [%d]: %d\n", i, arreglo[i]);

}

return 0;

}



En este programa simplemente se emplean con ciclos for y while, que hacen que los números en contenidos en el arreglo se vallan ordenando de menor a mayor y así se vallan mostrando en este orden.

## Ejercicio arreglo multidimensional:

2.- Indica que realiza el siguiente programa y completa los letreros del programa:

#include <stdio.h>

#define p printf

#define s scanf

int main()

{

char nombres[6][10] = {" jose ","teresa","mariana","mario","pedro","juan"};

char alum[10];

int cont=0;

p("Ingrese el nombre del alumno: \n");

s("%s", alum);

int i;

for(i=0; i<6; i++){

if (strcmp(nombres[i],alum)==0){ // compara el número mayor de nombres[i] con alum

p("El alumno %s aquí indicar que muestra el programa\n", nombres[i]);

}else{

cont++;

}

}

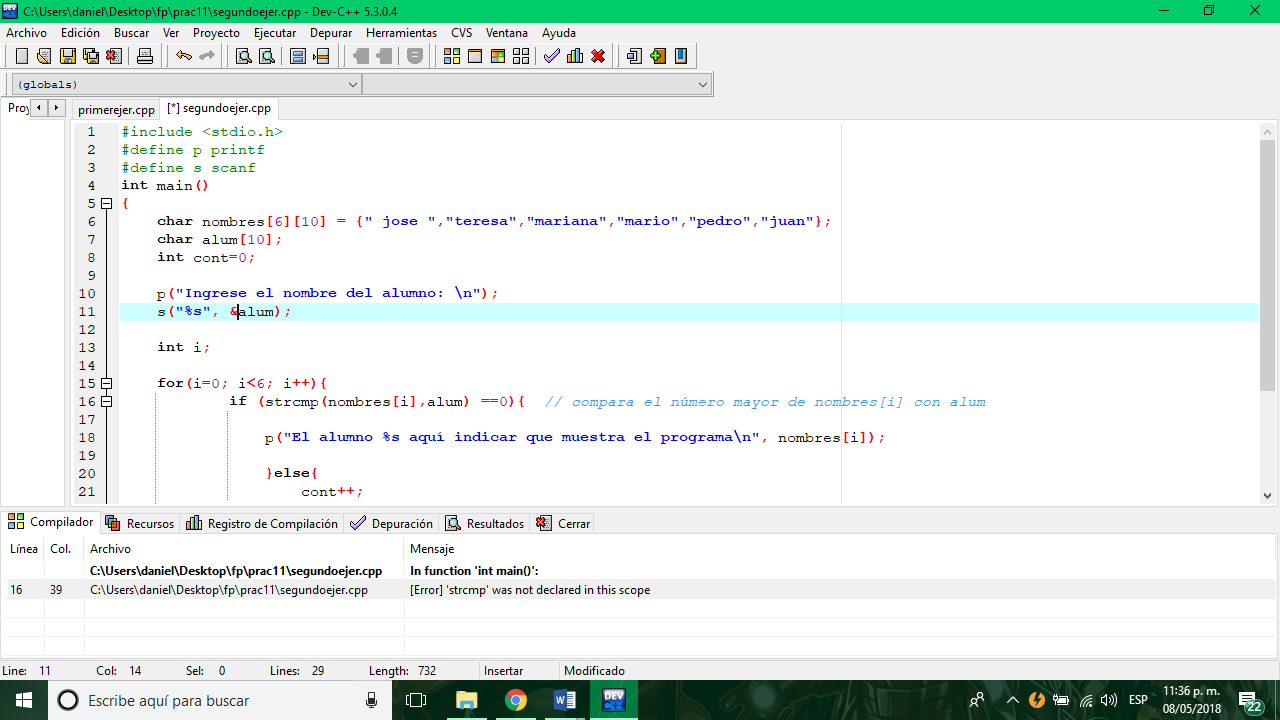
if (cont==6){

p("%s aquí indicar que muestra el programa ",alum);

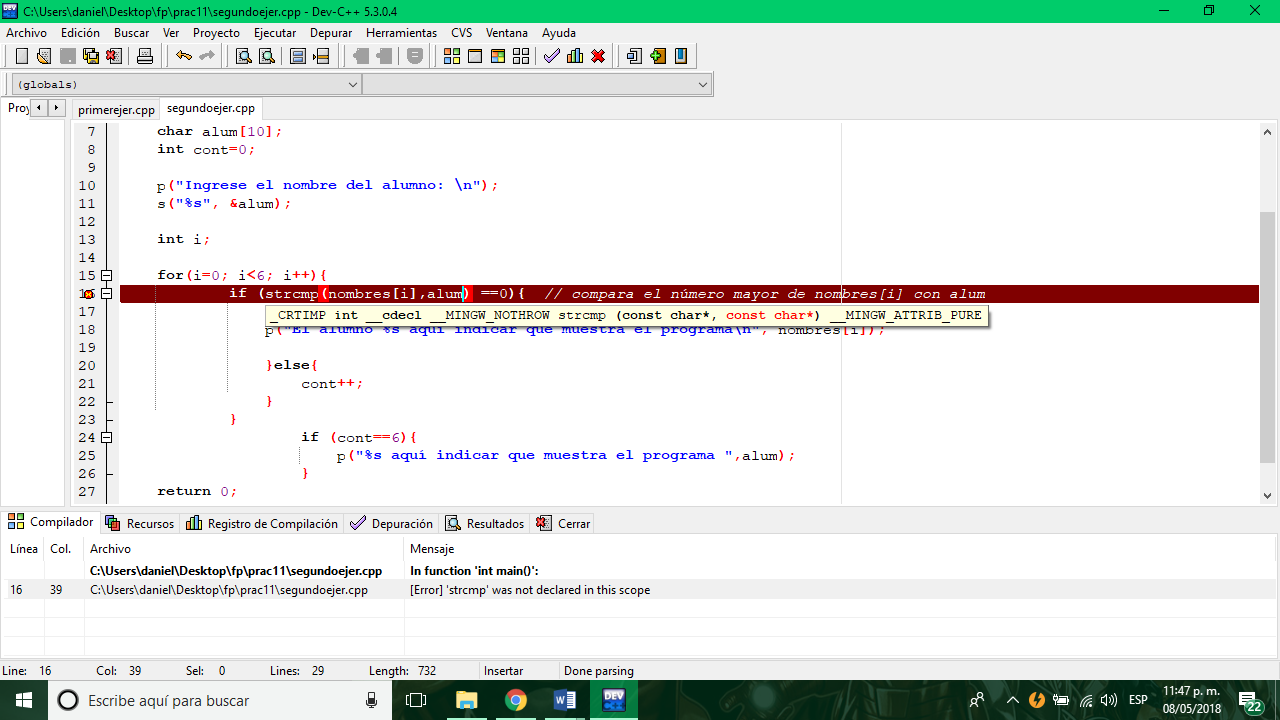
}

return 0;

}



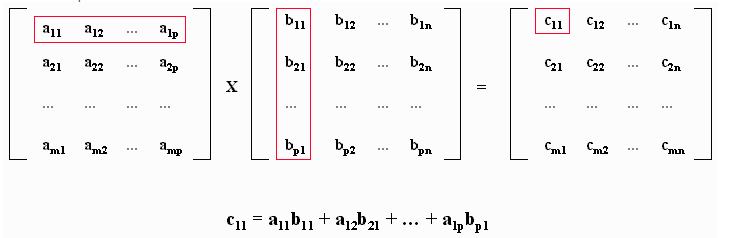
Primero lo que hice es que en el primer scanf le faltaba un “&”



Pero ahora e marca un erro ahí y no sé porque.

## Ejercicio arreglo y apuntadores:

Programa que solicite al usuario los valores de dos matrices de 3 x 3 y haga su multiplicación haciendo uso de arreglos. Conforme se muestrea a continuación:



**Conclusión:** Podemos concluir sabiendo la cantidad de cosas que se pueden hacer con los arreglos y los apuntadores y que gracias a ellos podemos acortar el código y ahora se nos abren más las posibilidades de soluciones de problemas, lo que si es que sigue siendo algo complejo y un poco complicado.

Bibliografía:

* El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.
* http://lcp02.fi-b.unam.mx/