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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Rodríguez Espino Claudia |
| *Asignatura:* | Fundamentos de Programación |
| *Grupo:* | 1102 |
| *No de Práctica(s):* | 11 |
| *Integrante(s):* | Morales Luna Emilio |
|  |  |
|  |  |
| *Semestre:* | 2018-1 |
| *Fecha de entrega:* | 12 de Noviembre de 2017 |
| *Observaciones:* |  |
|  |  |

Guía práctica de estudio 11: Arreglos unidimensionales y multidimensionales

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.

Actividades:

▪ Elaborar un programa en lenguaje C que emplee arreglos de una dimensión.

▪ Resolver un problema que requiera el uso de un arreglo de dos dimensiones, a través de un programa en lenguaje C.

▪ Manipular arreglos a través de índices y apuntadores.

Introducción

Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse.

A cada elemento (dato) del arreglo se le asocia una posición particular, el cual se requiere indicar para acceder a un elemento en específico. Esto se logra a través del uso de índices.

Los arreglos pueden ser unidimensionales o multidimensionales. Los arreglos se utilizan para hacer más eficiente el código de un programa.

Suma de Matrices

#include <stdio.h>

#include <conio.h>

int matriz1[4][4],matriz2[4][4],matriz3[4][4],a,b;

int main()

{

for (b=1;b<4;b++)

{

for (a=1;a<4;a++)

{

printf ("Ingrese valor de matriz1(%d,%d):",b,a);

scanf ("%d",&matriz1 [b][a]);

}

}

printf ("Matriz1:\n");

for (b=1;b<4;b++)

{

printf ("%d\t%d\t%d\n",matriz1 [b][1],matriz1 [b][2],matriz1 [b][3]);

}

for (b=1;b<4;b++)

{

for (a=1;a<4;a++)

{

printf ("Ingrese valor de Matriz2(%d,%d):",b,a);

scanf ("%d",&matriz2[b][a]);

}

}

printf ("Matriz2:\n");

for (b=1;b<4;b ++)

{

printf ("%d\t%d\t%d\n",matriz2 [b][1],matriz2 [b][2],matriz2 [b][3]);

}

printf ("Matriz1+matriz2:\n:");

for (b=1;b<4;b++)

{

printf ("%d\t%d\t%d\n",

(matriz1[b][1]+matriz2[b][1]),

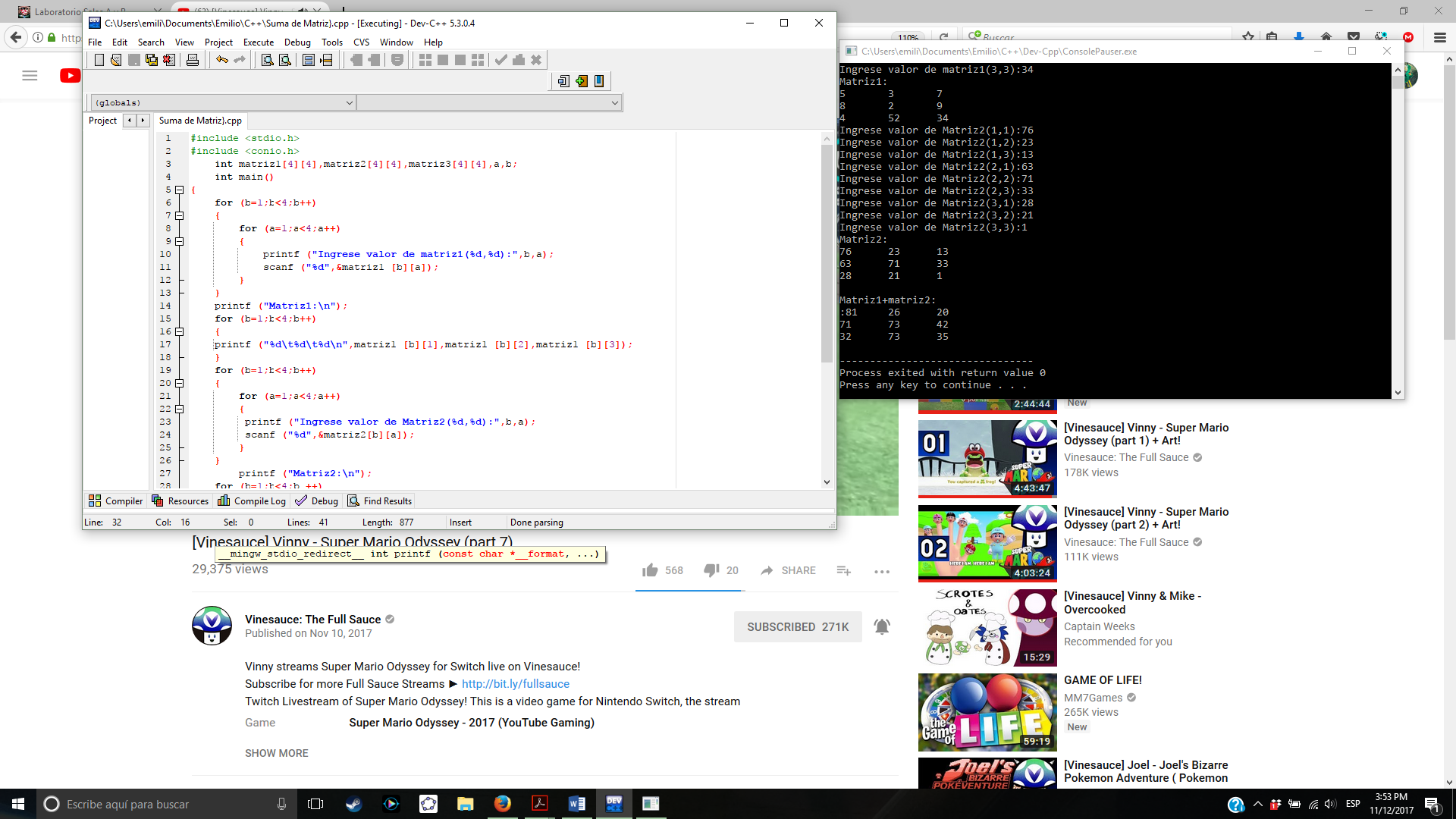
(matriz1[b][2]+matriz2[b][2]),

(matriz1[b][3]+matriz2[b][3]));

}

return 0;

}



Multiplicación de matriz con unidad escalar

#include <stdio.h>

#include <conio.h>

int matriz[4][4],a,b,c;

int main()

{

for (b=1;b<4;b++)

{

for (a=1;a<4;a++)

{

printf ("Ingrese valor de matriz(%d,%d):",b,a);

scanf ("%d",&matriz [b][a]);

}

}

printf ("Matriz:\n");

for (b=1;b<4;b++)

{

printf ("%d\t%d\t%d\n",matriz [b][1],matriz [b][2],matriz [b][3]);

}

printf ("Ingresar valor de c:\n");

scanf ("%d",&c);

printf ("\nMatriz multiplicara a %d:\n",c);

for (b=1;b<4;b++)

{

printf ("%d\t%d\t%d\n",

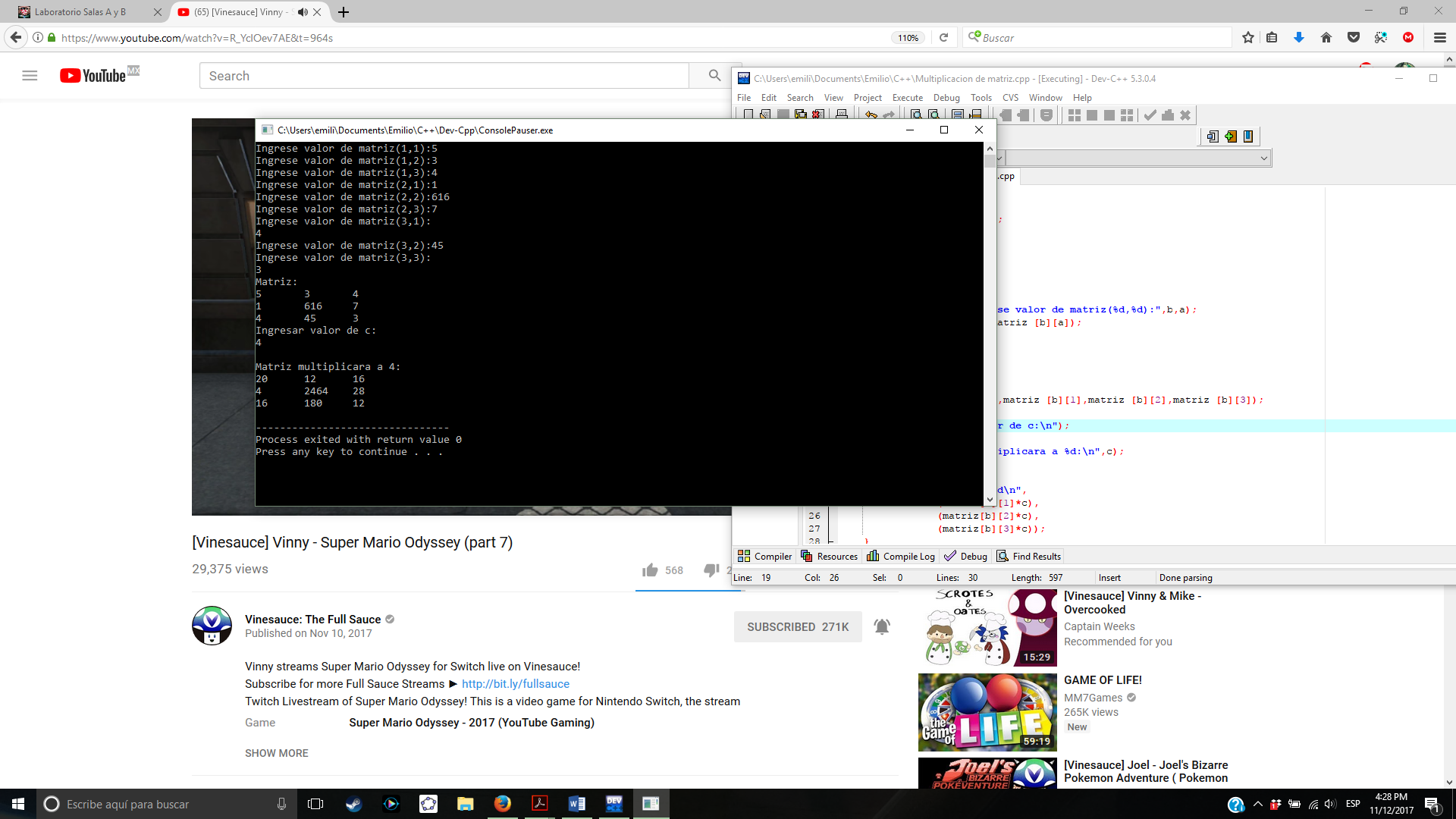
(matriz[b][1]\*c),

(matriz[b][2]\*c),

(matriz[b][3]\*c));

}

return 0;

 }

Gastos de los meses

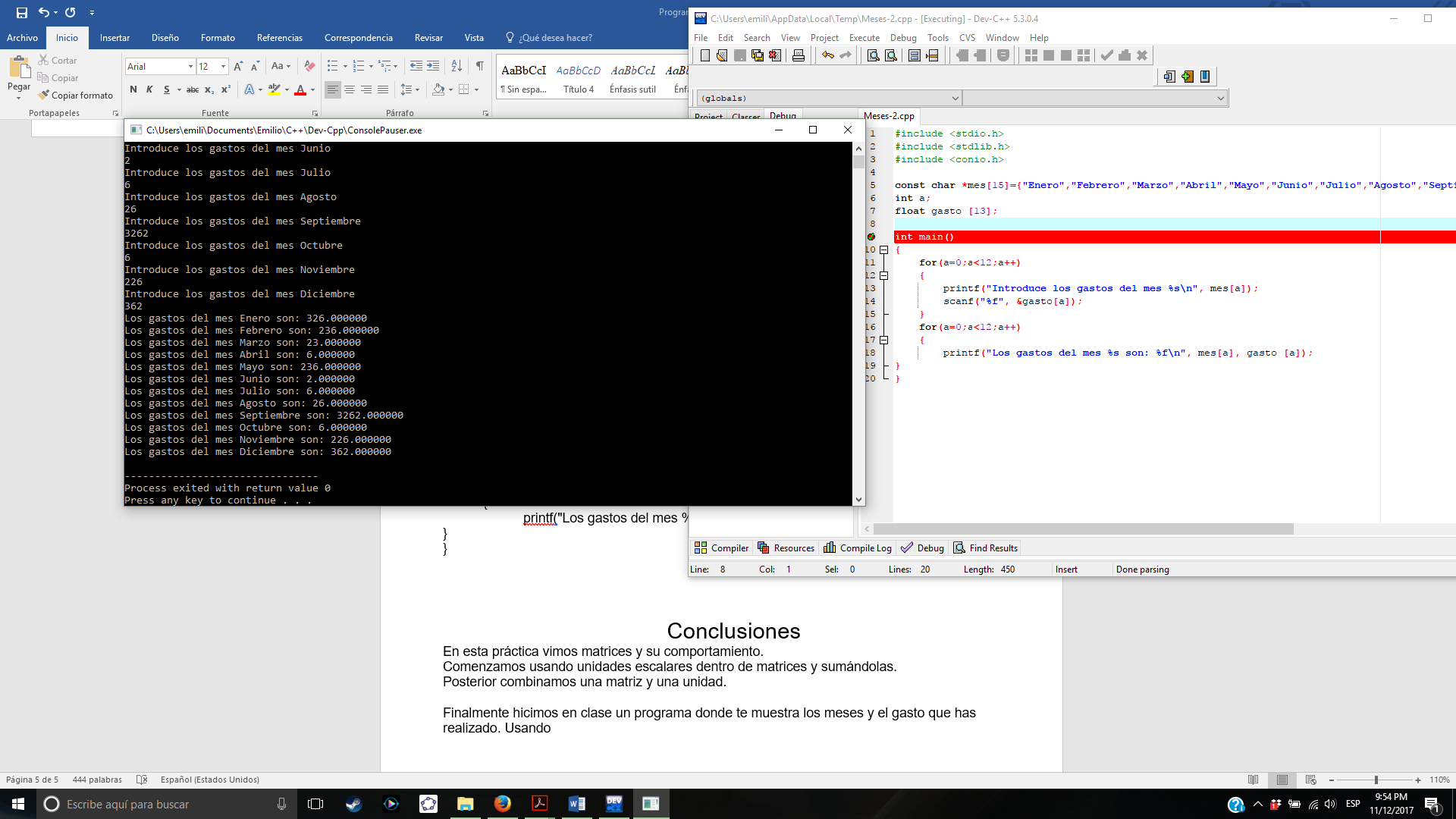
#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

const char \*mes[15]={"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre"};

int a;

float gasto [13];

int main()

{

for(a=0;a<12;a++)

{

printf("Introduce los gastos del mes %s\n", mes[a]);

scanf("%f", &gasto[a]);

}

for(a=0;a<12;a++)

{

printf("Los gastos del mes %s son: %f\n", mes[a], gasto [a]);

}

}

Conclusiones

En esta práctica vimos matrices y su comportamiento.

Comenzamos usando unidades escalares dentro de matrices y sumándolas.

Posterior combinamos una matriz y una unidad.

Finalmente hicimos en clase un programa donde te muestra los meses y el gasto que has realizado. Usando