



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Claudia Rodríguez Espino

*Asignatura:* Fundamentos de Programación

*Grupo:* 1102

*No de Práctica(s):* Práctica número 11

*Integrante(s):* Martínez Ramírez Pablo César

*Semestre:* 2018-1

*Fecha de entrega:* 03/10/2017

*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.

## ACTIVIDADES EN LABORATORIO:

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.

### MATRIZ POR ESCALAR:

## CÓDIGO FUENTE:

```
Programa22.cpp
1  #include<stdio.h>
2  #include<conio.h>
3
4  int cont, cont2, escalar, tamano;
5
6  main()
7  {
8      printf("\t\tMatriz por un escalar\n\n");
9      printf("Define el tamaño de tu matriz n x n: ");
10     scanf("%d",&tamano);
11     printf("\n");
12     int matriz[tamano][tamano];
13
14     for(cont=0;cont<tamano;cont++)
15     {
16         for(cont2=0;cont2<tamano;cont2++)
17         {
18             printf("Valor[%d][%d]de la matriz= ",cont,cont2);
19             scanf("%d",&matriz[cont][cont2]);
20             printf("\n");
21         }
22     }
23
24     printf("\nIntroduce el escalar a multiplicar con la matriz: ");
25     scanf("%d",&escalar);
26     printf("\nLa matriz resultante es:\n\n");
27
28     for(cont=0;cont<tamano;cont++)
29     {
30         for(cont2=0;cont2<tamano;cont2++)
31         {
32             printf("Valor [%d][%d]: %d\n",cont,cont2,escalar*matriz[cont][cont2]);
33         }
34     }
35     getch();
36 }
```

## CAPTURA DE CORRIDA:



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe

Matriz por un escalar

Define el tamaño de tu matriz n x n: 2

Valor[0][0] de la matriz= 1
Valor[0][1] de la matriz= 2
Valor[1][0] de la matriz= 3
Valor[1][1] de la matriz= 4

Introduce el escalar a multiplicar con la matriz: 2

La matriz resultante es:

Valor [0][0]: 2
Valor [0][1]: 4
Valor [1][0]: 6
Valor [1][1]: 8
_
```

## SUMA DE MATRICES

## CÓDIGO FUENTE:

```
[*] Programa 21.cpp
1  #include<stdio.h>
2
3  int arr1[3][3],arr2[3][3],arr3[3][3],a=0,b=0;
4
5  main()
6  {
7      printf("\t\tSumar dos matrices de 3x3\n\n");
8      printf("Llena los 9 valores de tu primera matriz:\n");
9
10     for(a=0;a<3;a++)
11     {
12         for(b=0;b<3;b++)
13         {
14             printf("\nIngresa el valor [%d][%d]: ",a,b);
15             scanf("%d",&arr1[a][b]);
16         }
17     }
18
19     printf("\n\nLlena los 9 valores de tu segunda matriz:\n");
20
21     for(a=0;a<3;a++)
22     {
23         for(b=0;b<3;b++)
24         {
25             printf("\nIngresa el valor [%d][%d]",a,b);
26             scanf("%d",&arr2[a][b]);
```

```

27     }
28 }
29
30 for(a=0;a<3;a++)
31 {
32     for(b=0;b<3;b++)
33     {
34         arr3[a][b]=arr1[a][b]+arr2[a][b];
35     }
36 }
37
38 printf("\n\nMatriz resultante:\n");
39
40
41 for(a=0;a<3;a++)
42 {
43     for(b=0;b<3;b++)
44     {
45         printf("\nLugar [%d][%d]= %d\n",a,b,arr3[a][b]);
46     }
47 }
48
49
50 }

```

## CAPTURA DE LA CORRIDA:

C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe  
Sumar dos matrices de 3x3

Llena los 9 valores de tu primera matriz:

Ingresa el valor [0][0]: 1

Ingresa el valor [0][1]: 2

Ingresa el valor [0][2]: 3

Ingresa el valor [1][0]: 4

Ingresa el valor [1][1]: 5

Ingresa el valor [1][2]: 6

Ingresa el valor [2][0]: 7

Ingresa el valor [2][1]: 8

Ingresa el valor [2][2]: 9

Llena los 9 valores de tu segunda matriz:

Ingresa el valor [0][0]: 1

Ingresa el valor [0][1]: 2

Ingresa el valor [0][2]: 3

Ingresa el valor [1][0]: 4

Ingresa el valor [1][1]: 5

Ingresa el valor [1][2]: 6

Ingresa el valor [2][0]: 7

Ingresa el valor [2][1]: 8

Ingresa el valor [2][2]: 9

Matriz resultante:

Lugar [0][0]= 2

Lugar [0][1]= 4

Lugar [0][2]= 6

Lugar [1][0]= 8

Lugar [1][1]= 10

Lugar [1][2]= 12

Lugar [2][0]= 14

Lugar [2][1]= 16

Lugar [2][2]= 18

-----  
Process exited with return value 0  
Press any key to continue . . .

## GASTOS MENSUALERS CON APUNTADORES:

### CÓDIGO FUENTE:

```
Programa 29.cpp
1 #include<stdio.h>
2 #include<conio.h>
3
4 const char *mes[15]={"Enero","Febrero","Marzo","Abril","Mayo","Junio",
5 "Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre"};
6 float gastos[13],b=0,promedio;
7
8 int cont;
9
10 main()
11 {
12     printf("\t\tGastos Mensuales\n\n");
13     for(cont=0;cont<12;cont++)
14     {
15         printf("Ingresa los gastos de %s: ",mes[cont]);
16         scanf("%f",&gastos[cont]);
17         printf("\n");
18         b=b+gastos[cont];
19     }
20
21     promedio=b/12;
22     printf("\n\nEl promeido de gastos al mes es: %f\n\n",promedio);
23
24     printf("Los gastos del %s son: %f pesos\n\n",mes[5], gastos[5]);
25     printf("Los gastos del %s son: %f pesos\n\n",mes[11], gastos[11]);
26     getch();
27 }
28
```

Registro de Compilación Depuración Resultados

### CAPTURA DE LA CORRIDA:

```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
Ingresa los gastos de Marzo: 258
Ingresa los gastos de Abril: 654
Ingresa los gastos de Mayo: 152
Ingresa los gastos de Junio: 365
Ingresa los gastos de Julio: 541
Ingresa los gastos de Agosto: 254
Ingresa los gastos de Septiembre: 265
Ingresa los gastos de Octubre: 325
Ingresa los gastos de Noviembre: 120
Ingresa los gastos de Diciembre: 145

El promedio de gastos al mes es: 312.000000
Los gastos del Junio son: 365.000000 pesos
Los gastos del Diciembre son: 145.000000 pesos

-----
Process exited with return value 0
Press any key to continue . . .
```

**CONCLUSIONES:**

El uso de los arreglos es mucho más sencillo de lo que creí que eran en un inicio, y también son demasiado útiles, creo yo que están en comparación a los ciclos o condicionales, ya que hay cosas que no se podrían realizar con estos. De algún modo creo que facilita todo al usuario y al programador. Basta con tener presente lo básico en teoría de los arreglos, para poder hacer programas de mayor dificultad, o mejor dicho, programas más elaborados o yendo más allá de lo básico, considero que teniendo presente el uso de algoritmos y de la lógica podemos sacar mucho provecho de todo esto.

**FUENTE:**

- [http://lcp02.fi-b.unam.mx/static/docs/PRACTICAS\\_FP/fp\\_p11.pdf](http://lcp02.fi-b.unam.mx/static/docs/PRACTICAS_FP/fp_p11.pdf)