



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Ing. Claudia Rodriguez Espino

Asignatura: Fundamentos de Programacion

Grupo: 3

No de Práctica(s): Practica 11


Integrante(s): Borja Portela Jose Fabio

Semestre: 2018-2

Fecha de entrega: 30 de abril del 2018

Observaciones:

CALIFICACIÓN: _____

	Manual de prácticas del Laboratorio de Fundamentos de programación	Código:	MADO-17
		Versión:	02
		Página	180/214
		Sección ISO	8.3
		Fecha de emisión	6 de abril de 2018
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

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.

INTRODUCCION:

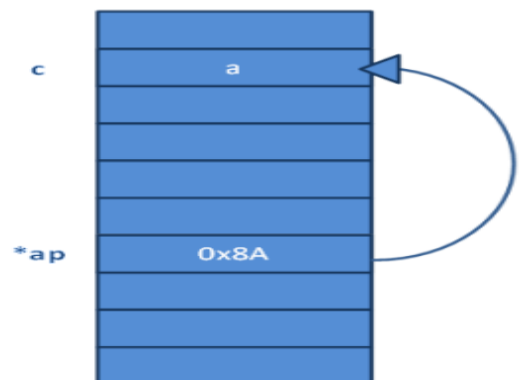
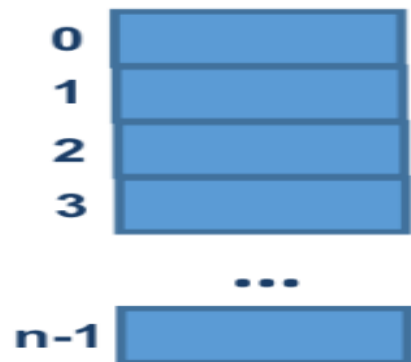
En el presente reporte se entregan capturas de pantalla pertinentes respectivas a los desarrollos de la práctica y también para los ejercicios de tarea, siempre y cuando sea el caso, dichas imágenes y procedimientos irán detallados por una lista de acciones y actividades seguidas en la clase de laboratorio.

DESARROLLO:

1.- se comenzó la clase dando la explicación de que es un arreglo, utilizando como apoyo principal un esquema, el cual será anexado a continuación, donde se estableció el tamaño de lo que es un arreglo, que va desde el índice cero hasta la componente $n-1$, ya que el primer espacio corresponde al índice cero de no ser inicializado con valores constantes para el arreglo.

Posteriormente se le dio importancia a lo que son los arreglos multidimensionales y haciendo su semejanza a las coordenadas de un vector o matriz, cada arreglo múltiple corresponde a la información que se llenará con la combinación de dichas coordenadas.

2.- por último se referenció a los apuntadores, los cuales son variables que funcionan para expresar el valor en memoria del buffer de otra variable normal, estos pueden ser bastante versátiles ya que no solamente muestran eso, también pueden tomar el valor asignado mediante la declaración o las operaciones de la función.



COMPLEMENTOS DE LA PRACTICA:

Para esta seccion se anexaran programas solicitados por la profesora, con el proposito de reforzar lo aprendido en clase, dichos codif¿gos de programa seran: el promedio de 5 numeros con arreglo unidimensional, los puntos de 3 jugadores de basketball con arreglos multidimensionales y por ultimo un simulador de gastos mensuales con apuntadores.

Primer Programa:

Promedio de 5 numeros con arreglos.cpp

```
1 #include<stdio.h>
2 #include<conio.h>
3 float arr[6],x;
4 int a;
5 main()
6 {
7     printf("Este programa calcula el promdeio de 5 numeros con arreglos unidimensionales\n");
8     for(a=1;a<6;a++)
9     {
10        printf("Dame el numero %d\n", a);
11        scanf("%f",&arr[a]);
12        x=x+arr[a];
13    }
14    x=x/5;
15    printf("el promedio es:\t%f\n",arr[a]);
16 }
17
```

Segundo Programa:

Promedio de 5 numeros con arreglos.cpp

Basket.cpp

meses con apuntadores.cpp

[*]ap me.cpp

```
1 #include<stdio.h>
2 #include<conio.h>
3 int i;
4 const char *mes[13]={"ENERO","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre"};
5
6 main()
7 {
8     printf("\t MESES\n");
9
10    for(i=0;i<12;i++)
11    {
12        printf("Mes [%d]: %c\n",i+1,*mes[i]);
13    }
14 }
```

Tercer programa:

Promedio de 5 numeros con arreglos.cpp

Basket.cpp

```
1 #include<stdio.h>
2 #include<conio.h>
3 int play[5][5], x, y, z;
4
5 main()
6 {
7     printf("Este programa realiza la suma de puntos de 3 jugadores de basket en 3 partidos\n");
8     printf("El jugador 1 es El Maicul,\nel jugador 2 es El Brayan y\nel jugador 3 es El yon\n");
9
10    for(x=1; x<4; x++)
11    {
12        for(y=1; y<4; y++)
13        {
14
15            printf("Dame los puntos(jugador | partido) [%d][%d]\n", x, y);
16            scanf("%d", &play[x][y]);
17        }
18    }
19
20    printf("Sumas totales\n");
21
22    for(x=1; x<4; x++)
23    {
24
25        for(y=1; y<4; y++)
26        {
27            z=play[x][y]+play[x][y]+play[x][y];
28
29
30        }
31        printf("Los puntos totales para %d son:\t%d\n", x, z);
32    }
33
34 }
```

CONCLUSIONES:

Se reconocio la importancia de utilizar arreglos para la programación estructurada de C++, ya que realizan funciones que ningún otro tipo de variable puede realizar.