



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Rodríguez Espino Claudia Ing.

Asignatura: Fundamentos de Programación

Grupo: 1104

No de Práctica(s): 11

Integrante(s): Santa Rosa Ortiz Thelma Jazmín.

*No. de Equipo de
cómputo empleado* 50

Semestre: 1°

Fecha de entrega: 28 de octubre del 2018

Observaciones:

CALIFICACIÓN: _____

Práctica 11. Arreglos multidimensionales

Objetivo:

Elaborar programas en lenguaje FORTRAN para resolver problemas que requieran agrupar conjuntos de datos del mismo tipo en arreglos multidimensionales.

Desarrollo:

Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo, definido al momento de crearse. Los arreglos pueden ser unidimensionales (como se vio en la práctica anterior) o multidimensionales y se utilizan para hacer más eficiente el código de un programa.

Arreglos multidimensionales

Lenguaje FORTRAN permite crear arreglos de varias dimensiones con la siguiente sintaxis:

tipoDato nombre (tamaño1, tamaño2,...,tamaño7)

o con la siguiente sintaxis:

tipoDato nombre
dimension nombre (tamaño1, tamaño2,...,tamaño7)
tipoDato DIMENSION (d1[, d2...]) :: nombre

Donde nombre se refiere al identificador del arreglo, tamaño es un número entero y define el número máximo de elementos que puede contener el arreglo por dimensión. La palabra reservada DIMENSION permite definir el tamaño máximo de las diferentes dimensiones.

Los tipos de dato que puede tolerar un arreglo multidimensional son: entero, real, carácter o complejo.

De manera práctica se puede considerar que la primera dimensión corresponde a los renglones, la segunda a las columnas, la tercera al plano, y así sucesivamente.

Por defecto, los arreglos inician en la posición 1 y, por tanto, las dimensiones se recorren de la posición 1 a la posición d1 para la primera dimensión, de la posición 1 a la posición d2 en la segunda dimensión, y así sucesivamente. Sin embargo, es posible indicarle al compilador de FORTRAN donde inician las dimensiones de la siguiente manera:

tipoDato DIMENSION (inicio_d1: d1, inicio_d2:d2, ...)

En este caso, las dimensiones se recorren desde la posición inicio_d1 hasta la posición d1 para la primera dimensión, de la posición inicio_d2 hasta la posición d2 para la segunda dimensión y así sucesivamente.

Arreglo unidimensional

Un arreglo unidimensional es una lista de valores guardados bajo el mismo nombre y del mismo tipo. Cada valor dentro del arreglo se le conoce como elemento del arreglo.

Para declarar un arreglo unidimensional en C++ se utiliza la siguiente línea de código:

tipo_dato identificador[tamaño];

En "tipo de dato" introduciremos que tipo de dato queremos (int, char, float.. etc) , luego en "identificador" introducimos como llamaremos a nuestro arreglo para ubicarlo e identificarlo de manera rápida. Y entre los corchetes colocaremos el tamaño que le demos a nuestro array.

PROMEDIO 5 NUMEROS CON ARREGLO UNIDIMENCIONAL.

Promedio de 5 números arreglo unidimensional.cpp Suma de matriz 4x4

```

1  #include<stdio.h>
2  #include<conio.h>
3  int a;
4  float r=0, arr[8], suma=0;
5  main()
6  {
7      for(a=1;a<6;a++)
8      {
9          printf("Dame el elemento %d: ",a);
10         scanf("%f", &arr[a]);
11         suma=(suma+arr[a]);
12     }
13     a--;
14     r=(suma/(a));
15     printf("El promedio es: %.2f",r);
16     getch();
17 }
```

E:\Programacion\Promedio de 5 n-meros arreglo unidimensional.exe

```

Dame el elemento 1: 96
Dame el elemento 2: 32
Dame el elemento 3: 1
Dame el elemento 4: 69
Dame el elemento 5: 3
El promedio es: 40.20
```

SUMA MATRICES 4X4

E:\Programacion\Suma de matriz 4x4.exe

Para la matriz 1

```

Dame el elemento[1][1]: 4
Dame el elemento[1][2]: 5
Dame el elemento[1][3]: 6
Dame el elemento[1][4]: 3
Dame el elemento[2][1]: 4
Dame el elemento[2][2]: 2
Dame el elemento[2][3]: 4
Dame el elemento[2][4]: 56
Dame el elemento[3][1]: 7
Dame el elemento[3][2]: 2
Dame el elemento[3][3]: 4
Dame el elemento[3][4]: 5
Dame el elemento[4][1]: 3
Dame el elemento[4][2]: 7
Dame el elemento[4][3]: 8
Dame el elemento[4][4]: 2
```

Para la matriz 2

```

Dame el elemento[1][1]: 1
Dame el elemento[1][2]: 5
Dame el elemento[1][3]: 4
Dame el elemento[1][4]: 6
Dame el elemento[2][1]: 3
Dame el elemento[2][2]: 2
Dame el elemento[2][3]: 8
Dame el elemento[2][4]: 9
Dame el elemento[3][1]: 7
Dame el elemento[3][2]: 4
Dame el elemento[3][3]: 1
Dame el elemento[3][4]: 2
Dame el elemento[4][1]: 5
Dame el elemento[4][2]: 3
Dame el elemento[4][3]: 5
Dame el elemento[4][4]: 4
```

La matriz resultante es:

```

El elemento [1] [1] es igual a: 5
El elemento [1] [2] es igual a: 9
El elemento [1] [3] es igual a: 11
El elemento [1] [4] es igual a: 9
El elemento [2] [1] es igual a: 7
El elemento [2] [2] es igual a: 4
El elemento [2] [3] es igual a: 12
El elemento [2] [4] es igual a: 65
El elemento [3] [1] es igual a: 14
El elemento [3] [2] es igual a: 6
El elemento [3] [3] es igual a: 5
El elemento [3] [4] es igual a: 7
El elemento [4] [1] es igual a: 8
El elemento [4] [2] es igual a: 10
El elemento [4] [3] es igual a: 13
El elemento [4] [4] es igual a: 6
```

Suma de matriz 4x4.cpp Gastos semanales apuntadores.cpp

```

1  #include<stdio.h>
2  #include<conio.h>
3  int x, y, arr1[5][5], arr3[5][5], arr2[5][5];
4
5  main()
6  {
7      printf("\nPara la matriz 1\n");
8
9      for(x=1;x<5;x++)
10     {
11         for(y=1;y<5;y++)
12         {
13             printf("\tDame el elemento[%d][%d]: ", x,y);
14             scanf("%d", &arr1[x][y]);
15         }
16     }
17     printf("\nPara la matriz 2\n");
18
19     for(x=1;x<5;x++)
20     {
21         for(y=1;y<5;y++)
22         {
23             printf("\tDame el elemento[%d][%d]: ", x,y);
24             scanf("%d", &arr2[x][y]);
25         }
26     }
27     printf("\nLa matriz resultante es: \n");
28     for(x=1;x<5;x++)
29     {
30         for(y=1;y<5;y++)
31         {
32             arr3[x][y]=arr1[x][y]+arr2[x][y];
33             printf("\tEl elemento [%d] [%d] es igual a: %d\n", x, y, arr3[x][y]);
34         }
35     }
36 }
```

GASTOS SEMANALES (con apuntadores)

Gastos semanales apuntadores.cpp

```
1
2 #include<stdio.h>
3 #include<conio.h>
4
5 int a,s=0;
6 float p=0,prom=0;
7 const char *dias[10]={"Lunes","Martes","Miercoles","Jueves","Viernes","Sabado","Domingo"};
8 float gasto[8];
9 main()
10 {
11     for(a=0;a<7;a++)
12     {
13         puts("");
14         printf("\tDame los gastos del dia %s: ",dias[a]);
15         scanf("%f", &gasto[a]);
16         s=(s+gasto[a]);
17     }
18     a--;
19     prom=(s/a);
20     printf("\n\t-El promedio Semanal es: $%.2f", prom);
21     getch();
22
23     getch();
24 }
```

E:\Programacion\Gastos semanales apuntadores.exe

```
Dame los gastos del dia Lunes: 634
Dame los gastos del dia Martes: 512
Dame los gastos del dia Miercoles: 436
Dame los gastos del dia Jueves: 200
Dame los gastos del dia Viernes: 300
Dame los gastos del dia Sabado: 164
Dame los gastos del dia Domingo: 458
-El promedio Semanal es: $450.00
```

Conclusión:

Gracias a esta práctica logramos elaborar programas en el lenguaje FORTRAN para resolver los problemas donde requerimos agrupar conjuntos de datos del mismo tipo en arreglos multidimensionales.