UNIVERSIDAD CATÓLICA
de Colombia
Vigilada Mineducación

Procedural Programming
Basic Concepts

Diego Alberto Rincón Yáñez
darincon@ucatolica.edu.co

Que es un arreglo?

- Una forma de agrupar una serie de variables del mismo tipo.
- La serie tiene el mismo nombre y se diferencia por un índice

Uso

- Si queremos guardar 5 enteros al mismo tiempo ¿Cómo lo hacemos?
  - Declaramos 5 variables enteras
- Y si quisiéramos almacenar 100?
  - Declarar 100 variables es un proceso tedioso
- ES MEJOR DECLARAR UN ARREGLO!

Estructura de un arreglo

- <tipoDato> <nombre> [<posiciones>];
- Ejemplo:
  - int arregloEnteros [100];
- Con esta variable puedo almacenar 100 números de tipo entero con el mismo nombre
- ¿Cómo?
  - Usando un índice

Estructura de un arreglo

- En la DECLARACIÓN de un arreglo, se coloca el MAXIMO de posiciones del arreglo
- EJ:
  - int miArreglo[20]; -> Declaración
  - Es un arreglo de tipo entero con 20 posiciones
- Para guardar una posición le digo en que posición de las 20 lo quiero
- Ej:
  - miArreglo[5]=10; -> Después de la declaración

Ejemplo del mundo real

- Cajones!

Ejemplo del mundo real

- Cuantos cajones hay?

Ejemplo del mundo real

- 6 Cajones!

Equivalente a: int cajones [6];

Ejemplo del mundo real

- Supongamos que cada cajón tiene un índice.

Ejemplo del mundo real

- La instrucción: cajones [4] = 10;
  equivale a:

Ejemplo del mundo real

- La instrucción: cajones [4] = 10;
  equivale a:

Ejemplo del mundo real

- Noten que la cantidad de cajones (6) es uno mas del índice más grande (5)

Ejemplo del mundo real

- Y que el primer índice no es 1, sino 0

Para recorrer un arreglo

- Se utiliza un ciclo de la siguiente forma
- for(int pos = 0; pos<MAX;pos++){
  - cout<<“el valor en ”<<pos<<“es: ”<< arreglo[pos]<<endl;
- Donde MAX es el numero máximo de “cajones”

Para recorrer un arreglo

- ¿Por qué se inicia en 0?
- ¿Por qué la condición es <MAX y no <=MAX?

Formas de iniciar los valores de un arreglo

- Forma1:
  - int arreglo [5] = {1,2,4,7,9};
  - float arreglof [6] = {1.3, 2.0, 3.0, 9.5, 4.2, 7.8}
  - etc…
- Forma 2:
  - int arreglo [ ] = {1,2,4,7,9};
  - float arreglof [ ] = {1.3, 2.0, 3.0, 9.5, 4.2, 7.8}
  - etc…
- Estas 2 formas son equivalentes…

Formas de iniciar TODOS los valores de un arreglo

- Forma 1: Con un ciclo
  - for (int pos=0; pos<MAX ;pos++){
    - arreglo[pos] = 0;
  - }
- Forma 2: Sin ciclo
  - int arreglo [6]={3};
  - float arreglo [6]={5.0};
  - etc…

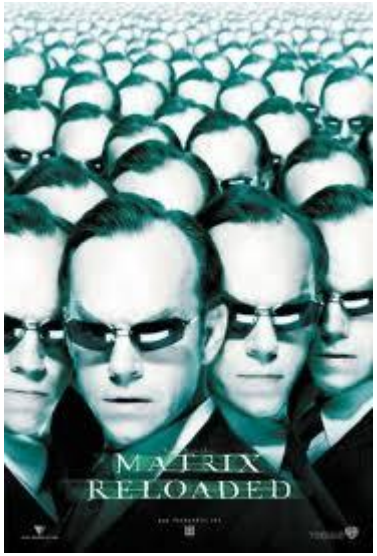
Algunas veces queremos enviar arreglos a funciones

- En la definición (o parámetro) de la función debe especificarse que la variable es un arreglo
- <tipo retorno> (<tipo> variable1, <tipo> arreglo1[] , <tipo> arreglo2[],…);

Igualmente el cuerpo de la función

- <tipo retorno> (<tipo> variable1, <tipo> arreglo1[] , <tipo> arreglo2[],…){
- //…….
- }

Las variables se “Clonan”, los arreglos NO



Variables en funciones

Arreglos en funciones

Matrices

- Son arreglos en 2 dimensiones.
- Cada uno de los datos se guarda en una posición dada su fila y su columna.

En un vector…

- Para acceder a una posición bastaba con el índice…

En un vector…

- Quiero guardar un elemento en el cajón de la fila 1

En una matriz…

En una matriz…

- Quiero guardar un elemento en el cajón de la fila 1

En una matriz…

- Basta sólo con 1 índice?

En una matriz…

- Son necesarios 2 índices.

En una matriz…

- Quiero guardar un elemento en el cajón de la fila 1 y columna 3

Estructura de una matriz

- <tipoDato> <nombre> [<filas>][<columnas>];
- Ejemplo:
  - int matrizEnteros [10][5];
- Con esta variable puedo almacenar 50 elementos (10 x 5) de tipo entero.

Estructura de una matriz

- En la DECLARACIÓN de una matriz, se coloca el MAXIMO de filas y columnas
- EJ:
  - int miMatriz[20][30]; -> Declaración
  - Es una matriz de tipo entero con 20 filas y 30 columnas
- Para guardar una posición le digo en qué posición de las 20 filas y las 30 columnas lo quiero
- Ej:
  - miMatriz[3][5]=10; -> Después de la declaración

Para recorrer una matriz

- Se utilizan 2 ciclos de la siguiente forma

```
for(int fila=0;fila<MAXF;fila++){
        for(int columna=0; columna<MAXC;columna++){
                cout<<"El arreglo en la fila "<<fila<<" y columna "
                <<columna<<" es:"<<miMatriz[fila][columna]<<endl;
        }
}
```

- Donde MAXF es el numero máximo de “filas” y MAXC el de “columnas”

Formas de iniciar los valores de una matriz

- Forma1:
  - int matriz [2][3] = {1,2,4,7,9,10};
  - int matriz [2][3] = {{1,2,4},{7,9,10}};
  - etc…
- Forma 2:
  - int matriz [][3] = {1,2,4,7,9,6};
  - float matrizf [ ][3] = {{1.3, 2.0, 3.0},{9.5, 4.2, 7.8}};
  - etc…
- Nótese que SIEMPRE los valores después del primero deben tener valor

Formas de iniciar los valores de una matriz

- Forma1:
  - int matriz [2][3] = {1,2,4,7,9,10};
  - int matriz [2][3] = {{1,2,4},{7,9,10}};
  - etc…
- Forma 2:
  - int matriz [][3] = {1,2,4,7,9,6};
  - float matrizf [ ][3] = {{1.3, 2.0, 3.0},{9.5, 4.2, 7.8}};
  - etc…
- Nótese que SIEMPRE los valores después del primero deben tener valor

Matrices y Funciones

- Para enviar una matriz a una función, la firma de la función debe ser de la siguiente forma

<tipoRetorno> <nombre> (<tipoDato> <nombreMatriz> [][X] , …);

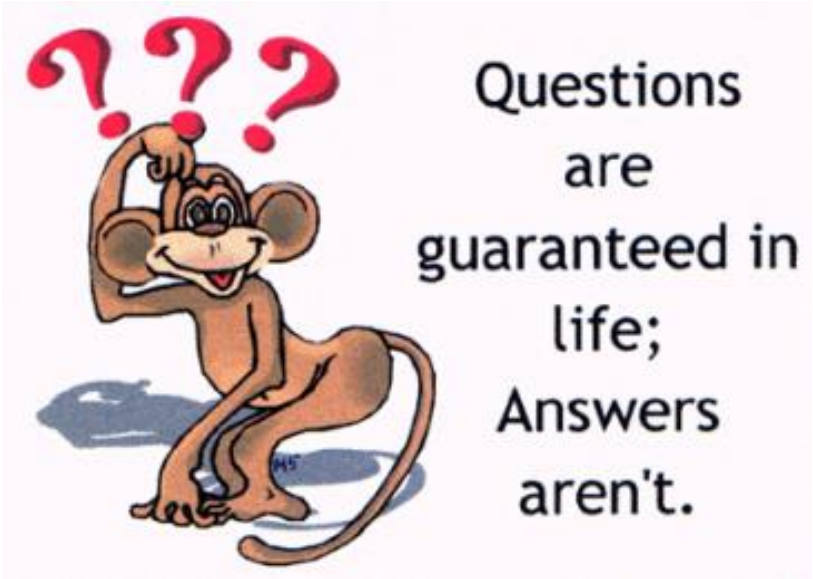
- Donde X es el número de columnas, este número es OBLIGATORIO

Matrices y Funciones

- La definición de la función es de la misma manera:

```
<tipoRetorno> <nombre> (<tipoDato> <nombreMatriz> [][X] , …){
    //código….
}
```

- Donde X es el número de columnas, este número es OBLIGATORIO



¿Questions?

Diego Alberto Rincón Yáñez MCSc.
Twitter: @d1egoprog.