

LENGUAJE FORTRAN. VECTORES Y MATRICES

Programación en Fortran

Valentín Moreno

ÍNDICE

1. Concepto de array
2. Definición de arrays
3. Uso de vectores
4. Uso de matrices



1. CONCEPTO DE ARRAY

3

1. CONCEPTO DE ARRAY

- Array: colección de objetos que comparten identificador (nombre) y tipo de datos
- Los elementos individuales del array pueden identificarse utilizando un índice
 - por ejemplo, se usa el índice para
 - encontrar el valor de una posición
 - recorrer el array
- Cada posición tiene un valor independiente de los demás
- Si es unidimensional lo llamamos vector
- Si tiene dos o más dimensiones lo llamamos matriz
- Se pueden utilizar arrays de más dimensiones pero no los trataremos en este curso.

1. CONCEPTO DE ARRAY

- Una dimensión = vector

- Ciudades de España

Ciudades (5)

Vector de dimensión 5

| | | | | |
|----------------|--------------|----------------|--------------------|--------------|
| Córdoba | Lugo | Badajoz | Guadalajara | León |
| Ciudades (1) | Ciudades (2) | Ciudades (3) | Ciudades (4) | Ciudades (5) |

(Córdoba Lugo Badajoz Guadalajara León)

1. CONCEPTO DE ARRAY

- Dos dimensiones = matriz

- Ciudades de Europa

Ciudades (3,4)

Matriz de dimensión 3 x 4

Primera dimensión (filas) →

| | | | | |
|---------|----------|---------|-------------|----------------|
| Córdoba | Lugo | Badajoz | Guadalajara | Ciudades (1,_) |
| Roma | Milán | Venecia | Florenca | Ciudades (2,_) |
| Paris | Toulouse | Nantes | Lyon | Ciudades (3,_) |

Segunda dimensión (columnas) ↓

Ciudades (_,1) Ciudades (_,2) Ciudades (_,3) Ciudades (_,4)

$$\begin{pmatrix} \textit{Córdoba} & \textit{Lugo} & \textit{Badajoz} & \textit{Guadalajara} \\ \textit{Roma} & \textit{Milán} & \textit{Venecia} & \textit{Florenca} \\ \textit{Paris} & \textit{Toulouse} & \textit{Nantes} & \textit{Lyon} \end{pmatrix}$$



2. DEFINICIÓN DE ARRAYS

7

2. DEFINICIÓN DE ARRAYS

○ Para declarar arrays

- hay que indicar
 - tipo de datos
 - rango (número de dimensiones)
- de esta forma el compilador conoce el espacio de memoria que es necesario reservar
 - n variables del mismo tipo
 - direcciones consecutivas
- pueden utilizarse constantes para definir las dimensiones del array

2. DEFINICIÓN DE ARRAYS

○ Sintaxis

● Vectores

tipo identificador(dimensión)

○ Ejemplo:

real ciudades (50) o

real ciudades (numCiudades)

(*numCiudades*: constante que debemos definir antes)

● Matrices

tipo identificador (dimensión1, dimensión2)

○ Ejemplo:

integer ciudades (5,4) o

integer ciudades (numPaises,numCiudades)

(*numPaises* y *numCiudades*: constantes que debemos definir antes)



3. USO DE VECTORES

10

3. USO DE VECTORES

- Acceso al valor de una componente de un vector

identificador(indice_componente)

- Ejemplo:

character (20) ciudades (5) *definición*

Córdoba

Lugo

Badajoz

Guadalajara

León

Para acceder a la componente **“Badajoz”** del vector ciudades deberemos utilizar:

ciudades (3)

acceso a la componente 3

3. USO DE VECTORES

○ Asignación/lectura de un vector:

- Los vectores se recorren componente a componente
 - si el vector no es muy grande se puede asignar/leer directamente cada componente

```
read*, fecha(1), fecha(2), fecha(3)    o    fecha(1) = 21  
                                          fecha(2) = 9  
                                          fecha(3) = 2010
```

- si no será necesario la utilización de bucles

```
do i=1,20  
    read*, puntuacion(i)  
end do
```

- si se utilizan constantes para la dimensión del vector, es aconsejable utilizar bucles para recorrerlo, ya que si la constante cambiase el programa no funcionaría

```
do i=1,numEquipos  
    puntuacion(i) = 0  
end do
```

3. USO DE VECTORES

○ Como imprimir un vector:

- Los vectores se pueden imprimir componente a componente, mediante un bucle

```
do i=1,20  
    print*, puntuacion(i)  
end do
```

- O se pueden imprimir en una sola línea (si el tamaño lo permite)

```
print*, puntuacion
```



4. USO DE MATRICES

14

4. USO DE MATRICES

- Acceso al valor de una componente de una matriz

identificador(índice_fila, índice_columna)

- Ejemplo:

character (20) ciudades (3,4)

definición

| | | | |
|---------|----------|---------|-------------|
| Córdoba | Lugo | Badajoz | Guadalajara |
| Roma | Milán | Venecia | Florenca |
| Paris | Toulouse | Nantes | Lyon |

Para acceder a la componente “*Venecia*” (fila 2 columna 3) de la matriz ciudades deberemos utilizar:

ciudades (2,3)

acceso a 1a componente (2,3)

4. USO DE MATRICES

○ Asignación/lectura de una matriz:

- Las matrices se recorren componente a componente
 - si la matriz no es muy grande se puede asignar/leer directamente cada componente
 - si no será necesario la utilización de bucles:
 - ✓ Como las matrices tienen dos dimensiones se utilizan dos bucles anidados
 - ✓ El primer bucle recorre una dimensión y el segundo la otra

recorrido por filas

```
do i=1,20
  do j=1,15
    read*, puntos (i,j)
  end do
end do
```

recorrido por columnas

```
do j=1,15
  do i=1,20
    puntos (i,j) = 0
  end do
end do
```

- si se utilizan constantes para las dimensiones de la matriz, es aconsejable utilizar bucles para recorrerla, ya que si alguna de las constantes cambiase el programa no funcionaría

4. USO DE MATRICES

○ Como imprimir matrices:

- Las matrices pueden imprimir componente a componente (del mismo modo que asignamos o leemos)

recorrido por filas

```
do i=1,20
  do j=1,15
    print*, puntos (i,j)
  end do
end do
```

recorrido por columnas

```
do j=1,15
  do i=1,20
    print*, puntos (i,j)
  end do
end do
```

- O podemos imprimirlas por filas o columnas

impresión por filas

```
do i=1,20
  print*, puntos (i,:)
end do
```

impresión por columnas

```
do j=1,15
  print*, puntos (:,j)
end do
```