

Algorítmica y Programación II

Tipos de datos
estructurados

Arreglos

- › El tipo de dato arreglo puede tener una, dos o más dimensiones.
- › Es un conjunto de elementos homogéneos.
- › Se necesitan índices, como cantidad de dimensiones, para referenciar a un único elemento.
- › Ej $M[\text{fila}, \text{columna}]$

$M[1,1] := 2$	2	5	0	0
$M[2,3] := 7$	3	-4	7	2
$M[3,4] := 1$	0	2	4	1

Matrices – Declaración Pascal

Const

N = 3;
M = 4;

El tipo de los índices debe ser ordinal
Entero, Caracter: 'a'..'z' o Enumerado

Type

TMatriz = array [1..N, 1..M] of Integer

...

Var

matriz: TMatriz;

3 filas

4 columnas

2	5	0	0
3	-4	7	2
0	2	4	1

Matrices – ejemplo

```
procedure inicializar(var M:TMatriz; const  
N,M:integer)
```

```
var I,j:integer;
```

```
Begin
```

```
    for i := 1 to N do
```

```
        for j := 1 to M do
```

```
            M[I,j] := i*100 + j*10;
```

```
end;
```

110	120	130	140
210	220	230	240
310	320	330	340

Implementación

› Arreglo unidimensional

- Formula para determinar el numero de celdas que se necesitan:

$$(\text{ord}(\text{cotaSuperior}) - \text{ord}(\text{cotaInferior}) + 1) * \text{tamaño}$$

- Formula para determinar la posición:

$$\text{Posición} = \text{base} + \text{ord}(\text{indice}) - \text{ord}(\text{cotaInferior})$$

Si el tamaño es más de una palabra:

$$\text{Posición} = \text{base} + (\text{ord}(\text{indice}) - \text{ord}(\text{cotaInferior})) * \text{tamaño}$$

Implementación

› Arreglo bidimensional

- Cantidad de elementos de la matriz:

$$(\text{ord}(\text{lsc}) - \text{ord}(\text{lic}) + 1) * (\text{ord}(\text{lsf}) - \text{ord}(\text{lif}) + 1)$$

- En Pascal los elementos se almacenan por filas:

a11- a12..... a1n; a21- a22a2n;

- Cantidad de elementos de la fila:

$$\text{Cantef} = \text{ord}(\text{lsc}) - \text{ord}(\text{lic}) + 1$$

Implementación

› Función de acceso en un arreglo bidimensional:

```
M[i, j] = dbase +  
  ( (ord(lsc) - ord(lic) + 1)  
    * (ord(i) - ord(lif))  
    * tamaño  
  + (ord(j) - ord(lic))  
    * tamaño  
  )
```

Registros

- › El tipo de datos registro puede almacenar diferentes tipos de datos relacionados.
- › Es un conjunto de elementos heterogéneo.
- › Se necesita el nombre del campo para acceder a un elemento y luego el valor dependerá de su estructura.

Type T = **record**

campo1 : tipo1;

campo2 : tipo2;

... campoN : tipoC

end;

Implementación

› Registro

- Los datos de una variable tipo registro se almacenan en posiciones consecutivas de memoria.
- Se reservan tantas posiciones como la suma de las posiciones de cada tipo de campo requieran.
- La variable registro tiene asociada una única dirección, llamada dirección Base, que es donde se almacena el primer campo del registro.

Implementación

› Registro variante

- Permite que una variable tipo registro, pueda ser **interpretada de varias formas.**
- Tiene:
 - Una parte fija (puede no existir)
 - Una parte variante, precedida por la instrucción CASE y un campo distintivo

Ejemplo de uso

Type

```
TPersonaUni = (docente,estudiante,egresado);
```

```
TUniveritario = record
```

fija

```
{ Legajo : Tlegajo;
```

```
credencial : TCredencial;
```

```
case orden:TPersonUni of
```

variable

```
{ docente      :( grado: 1..5; carga: 0..40 );
```

```
estudiante:(semestre:1..15; materias: integer);
```

```
egresado      :( egreso:1900..3000; titulo: string);
```

```
End;
```

Ejercicio

- a) En base al ejemplo anterior realice un subprograma que le permita al usuario ingresar todos los datos.
- b) En base al tipo de persona que sea, realice un subprograma que visualice por pantalla los datos de la persona.