Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática

FUNDAMENTOS DE PROGRAMACIÓN

Asignatura correspondiente al plan de estudios de la carrera de Ingeniería Informática

UNIDAD 4

Arreglos

2020

UNIDAD 4

Arreglos

Introducción

Hasta ahora hemos empleado variables simples. Su uso estaba limitado a una única posición de memoria en la cual podíamos alojar un dato individual. En ciertos casos es conveniente almacenar conjuntos de datos que guardan cierta relación entre sí; para ello requerimos el uso de estructuras de datos que permitan una mejor organización y tratamiento de esos datos.

En esta unidad trataremos la estructura de tipo arreglos que nos permite organizar un conjunto de datos homogéneos [de igual tipo] almacenados en forma contigua. Para introducirnos en el tema y justificar su empleo, analizaremos el ejemplo siguiente.

Ejemplo
Se poseen los resultados de una evaluación de la asignatura Análisis Matemático de un curso de 60 estudiantes. Se desea obtener e informar cuántos de ellos lograron superar la calificación media del curso.

```
Algoritmo Notas
      c ← 0; sum ← 0
      Repetir
            Leer Nota
            c ← c+1
            sum ← sum+Nota
      Hasta que c=10
      Media ← sum/c
      t ← 0; SupM ← 0
      Repetir
            t ← t+1
            Leer Nota
            Si Nota > Media entonces
                  SupM ← SupM+1
            FinSi
      Hasta que t = 60
      Escribir 'Nro de alumnos que superan la
                                    media:', SupM
FinAlgoritmo
```

En algorítmica computacional las operaciones de entrada/salida suelen ser las más lentas, pues implica el accionar de dispositivos electromecánicos (discos,

controladores de diskettes, de cintas, impresoras) que carecen de la velocidad del proceso puramente electrónico de cálculo. La situación empeora si la entrada es interactiva y depende de la velocidad de tipeo de un operador.

En el algoritmo resuelto del ejemplo introductorio ocurre el caso de tener que leer dos veces la lista de datos, pues la variable Nota sólo es capaz de almacenar un único valor y cada que vez que se le asigna una nueva lectura pierde el valor anterior. Imaginemos a un operador sentado frente a una computadora donde un programa le pide el ingreso de las 60 calificaciones del ejemplo, y al finalizar esta tarea el programa le solicita... ¡que las vuelva a tipear! Indudablemente, debemos pensar en una mejor solución en cuanto a la forma de organizar nuestra información en un algoritmo computacional.

Definición de Arreglo

Los problemas como los del ejemplo anterior plantean la necesidad de extender el concepto de dato para introducirnos en las estructuras de datos. Una estructura de datos permite organizar un conjunto de elementos de información bajo un mismo nombre o identificador.

Existen distintas estructuras de datos que se diferencian por la forma en que se relacionan los datos primitivos, por el tipo de los mismos y por la manera de referenciar cada dato. Al tratar la solución de un problema en particular, debemos analizar la organización de sus datos y las relaciones entre ellos, para definir y proponer estas estructuras.

En esta unidad analizaremos una estructura de datos llamada ARREGLOS y la definimos de la siguiente manera:

*Un **arreglo** es una estructura que permite representar un conjunto de datos del mismo tipo, cuyos elementos se referencian por su posición dentro de la estructura.*

Características de los arreglos

Al ser una estructura de datos, significa que se trata de un conjunto de datos o valores, donde cada elemento individual se almacena en una posición de memoria diferente. En el caso particular de un arreglo, sus componentes se almacenan en posiciones de memoria contiguas o consecutivas.

Resumiendo: Todo el arreglo tiene un nombre genérico único --que debe respetar las reglas sintácticas de los identificadores de variables-- y cada elemento del conjunto se identifica por este nombre más la posición que ocupa en la estructura. Sus componentes o elementos son homogéneos [del mismo tipo]: numérico, caracter, o lógico.

Los elementos se relacionan lógicamente entre sí. Representan distintos valores de un mismo ente o clase. Por ejemplo los datos de un arreglo pueden representar los legajos de los alumnos, los caudales diarios de una sección de un río, los nombres de los socios de un club, las notas de un conjunto de estudiantes, etc.
El arreglo tiene una dimensión declarada en el algoritmo, que establece la máxima cantidad de componentes que puede referenciar.

Los arreglos pueden clasificarse de acuerdo a la organización de sus elementos y a la forma de referenciar las posiciones de sus componentes. Así podemos distinguir arreglos unidimensionales o lineales, bidimensionales o tablas, y multidimensionales.

Arreglos Lineales

Un arreglo es lineal o unidimensional cuando la referencia a uno de sus componentes se realiza a través de un único valor llamado **índice**, que determina la posición del elemento dentro del arreglo. Estos arreglos son conocidos también como vectores.

Cada elemento del arreglo lineal se indica con el nombre del vector seguido del índice entre corchetes. Por ejemplo en un arreglo **a**:

- **a[5]** representa el quinto elemento del vector a
- **a[1]** representa el primer elemento del vector a
- **a[k]** representa el k-ésimo elemento del vector a

En general haremos referencia a un elemento de un arreglo lineal de la siguiente forma: **a[i]**. Siendo **a** el nombre del arreglo, e **i** el índice correspondiente al i-ésimo elemento del arreglo.

Destacamos que el nombre del arreglo es **a**, y las posiciones de sus elementos pueden ser accedidas a través e cualquier variable numérica, constante o expresión numérica; en todos los casos el valor de dicho índice debe ser un número natural (entero positivo).

Ejemplos de referencias a elementos de un arreglo A:

- `a[i+2]`
- `a[2*i]`
- `a[TRUNC((3+i)/2)]`

Consideremos el siguiente conjunto de datos numéricos

105 129 178 234 392 165

para organizarlos en un arreglo lineal v. En ese caso la representación de estructura que debemos crear sería la siguiente:

v[1]	v[2]	v[3]	v[4]	v[5]	v[6]

105	129	178	234	392	165

V[1] es el identificador o nombre de la posición de memoria donde se encuentra el valor 105, es decir el elemento ubicado en la posición 1 del vector. El tercer elemento el arreglo V[3] representa el dato 178.

Si en un algoritmo indicamos V[k] estaremos referenciando al elemento que ocupa la posición k dentro del arreglo lineal y para ello la variable k debe estar previamente definida, es decir debe tener asignado un valor. Si k tiene asignado el valor 3, entonces V[k] hará referencia a V[3].

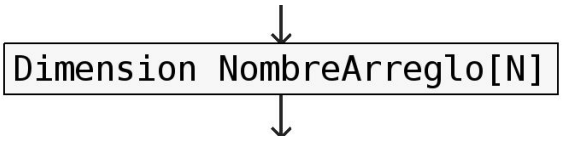
Se hace notar entonces que el índice puede obtenerse con cualquier identificador, constante o expresión.

Dimensión de un arreglo lineal

Por cada arreglo que se utilice en el algoritmo se debe indicar su dimensión, para
ello utilizaremos la siguiente sintaxis:

```
Dimension NombreArreglo[N]
```

Donde cantidad es un valor numérico entero que indica la máxima cantidad de elementos que tendrá el vector (cantidad que no podrá modificarse posteriormente). En un diagrama de flujo utilizaremos un bloque rectangular para indicar la acción de dimensionamiento.



Si se utilizan varios vectores en el algoritmo se debe indicar el nombre de cada uno separados por comas con su dimensión correspondiente:

```
Dimension arr1[200], arr2[450], arr3[50]
```

Con los conceptos hasta aquí vertidos vamos a resolver el mismo problema que nos introdujo en el tema al comienzo de esta unidad.

Ejemplo en pseudocódigo:
Se poseen los resultados de una evaluación de la asignatura Análisis Matemático de un curso de 60 estudiantes. Se desea obtener e informar cuántos de ellos lograron superar la calificación media del curso.

```
Algoritmo Notas
      Dimension Nota[60]
      c ← 0; sum ← 0
      Repetir
            c ← c+1
            Leer Nota[c]
            sum ← sum+Nota[c]
      Hasta que c=10
      Media ← sum/c

      t ← 0; Sup ← M0
      Repetir
            t ← t+1
            Si Nota[t] > Media Entonces
                  SupM ← SupM+1
            FinSi
      Hasta que c= 60
      Escribir 'Nro de alumnos que superan la
                                    media:', SupM
FinAlgoritmo
```

Obsérvense los cambios en el algoritmo propuestos como solución. Puede verse que en la segunda estructura de repetición hacemos referencia a cada elemento del arreglo evitando tener que leer nuevamente los datos; esto es porque la lista completa se conserva en memoria, ya que al ingresar el elemento Nota[2], el dato anterior Nota[1] no se altera.

También destaquemos que al emplear el arreglo Nota se emplearon variables diferentes para manejar el subíndice, sin embargo se trata del mismo arreglo.

Una desventaja del empleo de este tipo de estructuras es el hecho de requerir más posiciones de memoria para almacenar la lista completa. Pero las ventajas de su empleo son notorias.

Acciones que involucran a arreglos lineales

Al tratarse de un conjunto de variables cualquier acción algorítmica que admita la presencia de variables es posible utilizarla para el manejo de arreglos lineales. Es el caso de las sentencias de asignación, lectura, escritura o expresiones donde intervengan variables.

Por ejemplo, para leer un arreglo lineal de 100 elementos:

```
Dimensión A[100]
i ← 0
Mientras i<100 hacer
      i ← i+1
      Leer A[i]
FinMientras
```

Es posible alterar elementos particulares del arreglo. Por ejemplo, supongamos que en el arreglo A del ejemplo anterior deseamos duplicar el valor de los elementos inferiores a 700:

```
i ← 0
Mientras i<100 hacer
      i ← i+1
      Si A[i] < 700 Entonces
            A[i] ← A[i]*2
      FinSi
FinMientras
```

Si queremos escribir los valores que componen los elementos de un vector, deberemos hacerlo uno a uno, indicando el nombre del arreglo y lo posición del elemento que queremos obtener en la salida. Para mostrar los valores de cada uno de los elementos del vector A del ejemplo anterior debemos hacer:

```
i ← 1
Mientras i<=100 hacer
      Escribir A[i]
      i ← i+1
FinMientras
```

Estructura repetitiva Para-Hasta

Para facilitar el manejo de los arreglos incorporaremos nueva estructura de repetición llamada Para-Hasta. Esta estructura de control difiere de las ya conocidas Repetir y Mientras por el hecho de que el número de repeticiones es predeterminado y no depende de una proposición lógica, como es el caso de las dos mencionadas.

Sintaxis

```
Para i ← Vi Hasta Vf Con paso P Hacer
      acción 1
      acción 2
      .
      .
      acción n
FinPara
```

Donde

- **i**: variable numérica llamada variable de control de la estructura.
- **Vi**: variable o constante numérica; es el valor inicial que toma i.
- **Vf**: variable o constante numérica; es el valor final que toma i.
- **P**: variable o constante numérica; es el paso o incremento de i, el valor de P puede se positivo o negativo pero no puede ser cero.

A diferencia de las estructuras *Repetir* y *Mientras*, en esta estructura iterativa el procesador tiene a su cargo la variación de la variable de control (incrementa su

valor en P automáticamente en cada ciclo). Cuando i llega al valor final Vf se cumple el último ciclo y el control de ejecución abandona la estructura.

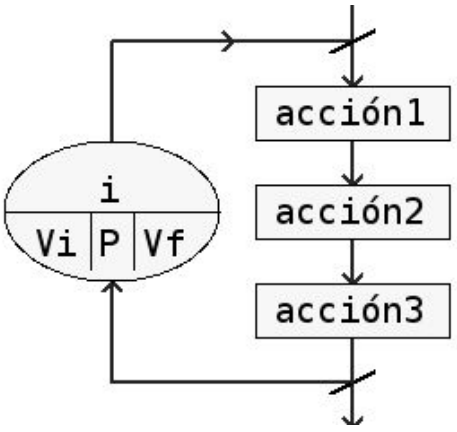
Ante la presencia de esta estructura de control el procesador ejecutará los siguientes pasos en el orden indicado:

1. A la variable i le asigna el valor de Vi.
2. Evalúa si el valor de la variable de control i es menor o igual a Vf.
   En caso afirmativo:
   a. Ejecuta las acciones previstas (acción 1, acción2, ...,acción n).
   b. Incrementa la variable i sumándole P..
   c. vuelve al paso 2

Cuando el paso o incremento es 1 la expresión *Con Paso 1* puede omitirse y la sintaxis se reduce:

```
Para i ← Vi Hasta Vf Hacer
     acción 1
     acción 2
     .
     .
     acción n
FinPara
```

Representación en diagrama de flujo:



Observaciones:

- Cuando P>0 el valor de Vf debe ser mayor o igual a Vi.
- Cuando P<0 el valor de Vf debe ser menor o igual a Vi.

Ejemplo de cómo informar los elementos de un arreglo:
Supongamos que en cierto algoritmo requerimos informar los elementos de un arreglo lineal B de 200 componentes:

```
.
Dimensión B[200].
.
.
Para i ← 1 Hasta 200 Hacer
     Escribir B[i]
FinPara
.
```

Anidamiento de estructuras Para-Hasta

El anidamiento de estas estructuras es una forma más de ciclos anidados, por lo tanto se siguen las mismas reglas, es decir la estructura interna anidada debe estar completamente incluida dentro de la estructura externa o anidante.
Además es posible anidar distintas estructuras iterativas: *Repetir*, *Mientras* y *Para*, siempre y cuando se respeten las reglas antes indicadas.

Arreglos Bidimensionales

Si la posición de un elemento en un arreglo debe ser precisada a través de 2 índices decimos que el arreglo es bidimensional. También se los conoce como tablas o matrices.

En una tabla, el primer índice nos da la posición de la fila y el segundo el de la columna. Por ejemplo, en una arreglo A de 3 filas y 4 columnas:

A[1,1]	A[1,2]	A[1,3]	A[1,4]
A[2,1]	A[2,2]	A[2,3]	A[2,4]
A[3,1]	A[3,2]	A[3,3]	A[3,4]

En este caso el dimensionamiento y lectura de los elementos del arreglo A debemos hacerlo de la siguiente forma:

```
.
Dimensión A[3,4]
Para f ← 1 Hasta 3 Hacer
     Para c ← 1 Hasta 4 Hacer
          Leer B[f,c]
     FinPara
FinPara
.
```

En el tramo de algoritmo propuesto, estamos leyendo los elementos de la tabla por filas. Esto es porque para un valor de fila f recorremos con la variable c todas las columnas. Si intercambiáramos el anidamiento de las estructuras Para-Hasta podríamos leer por columnas la matriz.

Arreglos Multidimensionales

Es posible proponer estructuras de tipo arreglo, donde las posiciones de cada elemento deban ser referenciadas por 3 o más índices. Por ejemplo: si deseamos manejar en un algoritmo la lista de alumnos de una escuela de enseñanza media que posee 5 años, 6 divisiones por año y 30 alumnos por curso o división, deberemos dimensionar un arreglo tridimensional:

```
Dimensión A[5,6,30]
```

Y para referenciar al alumno Nro. 17, que cursa en 2do año, división 4: A[2,4,17]. Todos los conceptos vertidos para arreglos lineales y bidimensionales son válidos aquí también.

Síntesis

1. Los arreglos nos permiten organizar un conjunto de datos homogéneos.

2. Un arreglo es una estructura que permite representar un conjunto de datos del mismo tipo y cuyos elementos se referencian por su posición dentro de la estructura.

3. Todo el arreglo tiene un nombre genérico único.

4. Sus componentes o elementos son homogéneos.

5. Los elementos se relacionan lógicamente entre sí.

6. El arreglo tiene una dimensión declarada en el algoritmo.

7. Dimensión de un arreglo lineal sintaxis:

```
Dimensión nombrearreglo[cantidad]
```

8. La estructura de control Para Hasta difiere de Repetir y Mientras por el hecho de que el número de repeticiones es predeterminado y no depende de una proposición lógica.

9. Es posible anidar distintas estructuras iterativas: Repetir, Mientras y Para-Hasta.

10. Es posible proponer estructuras de tipo arreglo, donde las posiciones de cada elemento deban ser referenciadas por 3 o más índices.