



aprenderaprogramar.com

Más sobre arrays estáticos y dinámicos en pseudocódigo. Límites LBound y UBound. (CU00213A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº12 del Curso Bases de la programación Nivel II

24

En definitiva, hemos visto que los valores máximos en el array se convierten en importantes portadores de información y definidores del array. Vamos a considerar 2 variantes para el seguimiento de los valores máximos de los localizadores.

- a) Seguimiento con variables tipo señalero.
- b) Seguimiento mediante invocación a límite superior e inferior.

Seguimiento con variables tipo señalero.

Esta estrategia supondrá crear tantas variables como número de localizadores tenga el array, con el fin de que cada una de ellas almacene el valor máximo de cada localizador. Consideremos un caso hipotético de una matriz cuyas dimensiones $m \times n$ son establecidas por el usuario, así como el valor de cada elemento. Supongamos que tras la introducción de datos la matriz se amplía con una fila más y una columna más. El pseudocódigo sería algo así:

```

Pedir m
Maxdatom = m
Pedir n
Maxdaton = n
Redimensionar Dato(m,n)
Desde i = 1 hasta m Hacer
    Desde j = 1 hasta n Hacer
        Pedir Dato(i,j)
    Siguiete j
Siguiete i
Mostrar "El número de elementos es", Maxdatom * Maxdaton
Redimensionar Dato(m+1,n+1)
Maxdatom = m + 1
Maxdaton = n + 1
Mostrar "El número de elementos es", Maxdatom * Maxdaton

```

Los señaleros Maxdatom y Maxdaton van indicando en cada momento las dimensiones de la matriz por lo que tienen que ser actualizados en cada redimensionamiento. Usar contadores o usar el valor inicial del localizador máximo suele suponer, a la larga, equivocaciones a la hora de saber el estado real del array. Lo más claro y conveniente, en general, es disponer de señaleros específicos.

Seguimiento mediante invocación a límite superior e inferior.

Usaremos la sintaxis:

```

Limesuperior([array], [número de localizador])
Limiteinferior([array], [número de localizador])

```

La invocación a límite superior nos devolverá el valor máximo de uno de los localizadores del array, indicado con un número que se corresponde con su situación de izquierda a derecha dentro de la declaración del array. Si sólo existe un localizador, no será necesario indicar número de localizador siendo indiferente poner 1.

Ejemplos: Se han definido

Ciudad(5) Dato(4,2) Encuesta(5,3,7,7)

Limitesuperior(Ciudad) devuelve 5

Limitesuperior(Ciudad,1) devuelve 5

Limitesuperior(Ciudad,2) devuelve error por no existir un segundo localizador

Limitesuperior(Dato,1) devuelve 4

Limitesuperior(Dato,2) devuelve 2

Limitesuperior(Encuesta,3) devuelve 7

Para un programa como el anterior, de una matriz cuyas dimensiones $m \times n$ y valores son establecidos por el usuario, que después se amplía en una fila y columna más, el ejemplo de uso de *Limitesuperior* sería el siguiente:

Pedir m Pedir n Redimensionar Dato(m,n) Desde i = 1 hasta m Hacer Desde j = 1 hasta n Hacer Pedir Dato(i,j) Siguiente j Siguiente i Mostrar "El número de elementos es", <i>Limitesuperior(Dato,1)</i> * <i>Limitesuperior(Dato,2)</i> Redimensionar Dato(m+1,n+1) Mostrar "El número de elementos es", <i>Limitesuperior(Dato,1)</i> * <i>Limitesuperior(Dato,2)</i>

Entre las ventajas que observamos respecto a lo visto anteriormente tenemos:

- No es necesario el uso de señaleros auxiliares.
- El ajuste del valor del límite superior es automático tras un redimensionamiento. No es necesario por tanto realizar una actualización en paralelo, con lo que los riesgos de fallos por olvido o mala asignación quedan prácticamente eliminados.

Para programas largos con decenas de matrices y localizadores esto supone un alivio considerable.

La invocación a límite inferior funciona de manera análoga a lo visto hasta ahora, pero devolviéndonos el valor mínimo del localizador. Si se definió Ciudad(5), Limiteinferior(Ciudad) nos devuelve 1 (estamos considerando valor base 1 y no cero). Su utilidad la veremos más adelante.

La disponibilidad o no de una función que nos devuelva los valores máximo y mínimo de los localizadores de un array dependerá del lenguaje que utilicemos. Cuando no dispongamos de esta herramienta o sus características no sean las adecuadas a nuestros intereses, recurriremos al uso de señaleros.

Próxima entrega: CU00214A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60