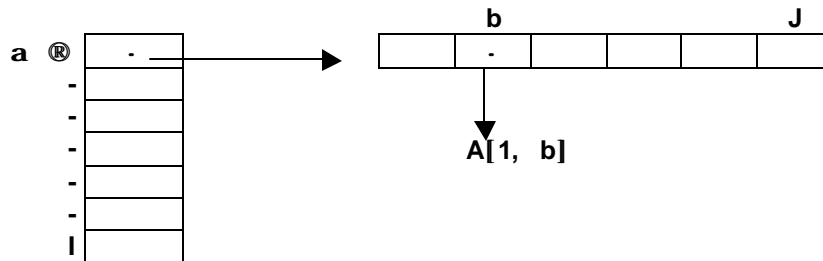


Trabajo Práctico Nº 5

Tema: DATOS - OPERACIONES

1. Muchos cálculos con matrices homogéneas implican el procesamiento secuencial de todos los elementos de una sola fila o columna. Es probable que en la iteración en la que se realiza el procesamiento intervengan referencias a $A[i, j]$ con el subíndice i o j aumentado en uno cada vez que se completa la iteración. En estos casos resulta ineficiente calcular $\text{valor}(A[i, j])$ de manera independiente en cada iteración; en vez de ello (si se supone que el subíndice i es el que está variando), $\text{valor}(A[i, j])$ se puede calcular de manera más sencilla a partir de $(A[i-1, j])$. Proporcione la fórmula para el cálculo de $\text{valor}(A[i, j])$ en términos de $\text{valor}(A[i-1, j])$

2. En LISP, SIMSCRIPT y otros lenguajes se representa una matriz heterogénea con una estructura enlazada. Dar un algoritmo para acceder.



3. Una SECCION o RABANADA de una matriz (por ejemplo $A[* , 3]$ o $A[2, *]$) denota una submatriz. Diseñar un descriptor adecuado y discutir la implementación de $B[i] = A[* , i]$ y / o $B[i] = A[i, *]$

Ejemplo

1	4	7
2	5	8
3	6	9

1	2	3	5
0	2	8	5
9	7	6	0
8	4	0	1

4. En un lenguaje de programación se utiliza la representación de dispersión para el tipo de datos conjunto. Diseñar un algoritmo $\text{UNION}(S_1, S_2, S_3)$ que compute $S_3 = S_1 \cup S_2$

5. La fecha es un dato muy utilizado en la mayoría de las aplicaciones, más allá del tipo que sean. Por tal motivo, se considera muy útil que un lenguaje de programación proporcione un tipo de dato FECHA, con el cual el usuario pueda declarar una variable de este tipo y representar la fecha como: dd-mm-aaaa..

- Proponga una representación de almacenamiento adecuada para datos FECHA.
- Proporcione métodos de acceso
- Enumere y declare la signatura de las operaciones del tipo de dato FECHA
- Defina las reglas semánticas para este tipo que sirvan para la verificación de tipos.

6. Implementar en Pascal un procedimiento $\text{Concatenar}(C1, C2, C3)$ que forma el String $C3$, agregando los caracteres de $C2$ a continuación de los caracteres de $C1$. No utilizar las funciones predefinidas del lenguaje.

7. Implementar en lenguaje C, las funciones que convierten cadenas (strings) a entero y viceversa.

8. Ídem anterior pero de cadena a flotante y viceversa.

9. Implementar en C una función $\text{expand}(s1, s2)$ que expanda notaciones abreviadas como a-z en la cadena $s1$, en una lista completa $abx...xyz$ en $s2$. Se deben permitir letras mayúsculas y minúsculas, dígitos y el manejo de cosas como $ab-c$, $az0-9$ y az . (Una convención útil es que los guiones del comienzo y del final se tomen literalmente) Capítulo 3, K&R.

Trabajo Práctico Nº 5

Tema: DATOS - OPERACIONES

10. En el lenguaje C, existe una estrecha relación entre punteros y arreglos. Cualquier operación que se pueda realizar mediante indexación de un arreglo se puede realizar también con punteros. La declaración:

```
int a [10]
```

define un arreglo de tamaño 10, es decir un bloque de 10 objetos consecutivos denominados $a[0]$, $a[1]$, $a[2]$, ..., $a[9]$. La notación $a[i]$ significa el elemento del arreglo que se encuentra a i posiciones del comienzo. Si pa es un puntero a un entero:

```
int *pa
```

entonces la asignación:

```
pa = &a[0]
```

hace que pa apunte al entero cero de a ; es decir pa contiene la dirección de $a[0]$. Ahora la asignación:

```
x = *pa
```

copiará el contenido de $a[0]$ en x .

Si pa apunta a un elemento particular de un arreglo a , entonces por *definición* $pa + 1$ apunta al siguiente elemento, y en general $pa - i$ apunta a i elementos antes de pa , y $pa + i$ apunta a i elementos después. Si pa apunta a $a[0]$, entonces:

```
*(pa+1)
```

se refiere al contenido de $a[1]$, $pa + i$ es la dirección de $a[i]$ y $*(pa+i)$ es el contenido de $a[i]$.

Estos comentarios son ciertos cualquiera que sea el tipo de variables del arreglo a . La definición de "sumar 1 a un apuntador", y por extensión, toda la aritmética de punteros establece que el incremento se adecua al tamaño en memoria del objeto apuntado. En $pa + i$, i se multiplica por el tamaño de los objetos a los que apunta pa antes de ser sumado a pa .

La correspondencia entre indexación y aritmética de punteros es evidentemente muy estrecha. En efecto, el compilador convierte toda referencia a un arreglo en un puntero al comienzo del arreglo. El efecto es que el nombre de un arreglo es una expresión de tipo puntero.

Escriba un programa en C que genere un arreglo de caracteres y luego imprimir los valores utilizando un puntero. (capítulo 5, K & R)

Nota

K & R : El lenguaje de programación C - Kernighan & Ritchie