PRÁCTICA 7: PROGRAMAS EN ENSAMBLADOR PARA MANEJO DE MATRICES

OBJETIVOS:

Esta práctica tiene como objetivo la definición de una estructura de datos más compleja así como el manejo de la misma para la realización de operaciones matemáticas sencillas. La estructura de datos es la de *matriz o array*.

La mayor dificultad estriba en que la estructura matriz se almacena en memoria como un vector, esto es, de forma lineal, y el manejo de los índices resulta un tanto artificial, pues no hay una correspondencia entre la estructura física y la estructura lógica.

CONTENIDOS:

1. Definición de la estructura de una matriz en memoria
2. Operaciones de insertar, borrar y buscar elementos en una matriz por filas y columnas
3. Algoritmo de Booth
4. Triángulo de Tartaglia

1. Definición de la estructura de memoria

Partamos de la definición de una matriz de 3 x 3. Se tienen dos índices *i,j* para identificar las filas y las columnas respectivamente. Estos índices para el ejemplo de 3 x 3 tomarán los valores 0,1 ó 2 para indicar todos los elementos de la matriz. Para el manejo de los elementos almacenados en memoria, habrá que establecer la correspondencia entre los índices *i,j* y un índice lineal, por ejemplo n:

La matriz en memoria realmente es:

n	0	1	2	3	4	5	6	7	8	9

Pero la estructura matemática de una matriz es::

i/j	Col 0	Col 1	Col 2
Fila 0	0,0 (0)	0,1 (1)	0,2 (2)
Fila 1	1,0 (3)	1,1 (4)	1,2 (5)
Fila 2	2,0 (6)	2,1 (7)	2,2 (8)

La relación entre los índices es:

$$n=3*i + j$$

Una operación común de matrices es la transpuesta. Si a la matriz de 3 x 3 anterior le hallamos su transpuesta entonces quedaría:

i/j	Col 0	Col 1	Col 2
Fila 0	0,0 (0)	1,0 (1)	2,0 (2)
Fila 1	0,1 (3)	1,1 (4)	2,1 (5)
Fila 2	0,2 (6)	1,2 (7)	2,2 (8)

La relación entre los índices es ahora:

$$n=i +3* j$$

2. Operaciones de insertar, borrar y buscar elementos de una matriz

Las operaciones de insertar, borrar y buscar elementos se realizan teniendo en cuenta la relación entre los índices de la matriz y el elemento de memoria correspondiente.

3. Ejemplo 3: ALGORITMO DE BOOTH

El **algoritmo de Booth** es un método rápido y sencillo para obtener el producto de dos números binarios con signo en notación complemento a dos.

Debemos saber que un **número binario** está formado por bits de ceros y unos, y que se puede traducir a decimal fácilmente de la siguiente forma:

128	64	32	16	8	4	2	1
0	1	0	1	0	1	1	0

Sabiendo que la posición de cada bit es 2^n (*elevado a n*) y partimos de n=0 de derecha a izquierda, sólo queda realizar la suma total de multiplicar por dicho bit, en este caso: (*0·2^7+1·2^6+0·2^5+1·2^4+0·2^3+1·2^2+1·2^1+0·2^0 = 86*).

También debemos saber que el **complemento a uno** de un número binario es cambiar sus ceros por unos, y sus unos por ceros (*complementar*): (*010010* -> ca1: *101101*) y que el **complemento a dos** de un número binario es el resultado de sumar 1 al complemento a uno de dicho número binario:

número binario:
0101 0110

Ca1: 1010 1001
+1

Ca2: 1010 1010

Realizar una suma con dos números binarios es tarea fácil, pero la multiplicación resulta algo más complicada. Con el algoritmo de Booth, resulta mucho más sencillo de implementar. Partimos del ejemplo de la multiplicación 6·2=12:

$$\underbrace{\texttt{0000 0110}}_{\text{8 bits}}\ \texttt{(6)}$$

```
0000 0110  (6)
0000 0010  (2)
└────────┘
  8 bits
```

```
   multiplicando                  bit extra
  ┌─────────┐                         ┌┐
A=0000 0110  0000 0000  0
   multiplicando en ca2
  ┌─────────┐
S=1111 1010  0000 0000  0
              multiplicador
             ┌─────────┐
P=0000 0000  0000 0010  0
```

Como se puede ver en la imagen superior, partiendo de los números binarios de la multiplicación 6·2 (*multiplicando y multiplicador*) creamos tres nuevos números binarios del doble de tamaño (*16 en el ejemplo*): A, S y P.

Partiendo del número P (*producto*) comenzamos a comparar los últimos 2 bits de la derecha, siguiendo los casos base del recuadro:

```
CASOS BASE
0 0 -> No se realiza
       ninguna acción
0 1 -> P = P + A
1 0 -> P = P + S
1 1 -> No se realiza
       ninguna acción
```

Se realizará esta comparación 8 veces en este ejemplo (*número de bits de los operandos*) y al final de cada comparación, realizamos un desplazamiento de un bit hacia la derecha, manteniendo el último bit de la izquierda, y descartando el último bit del lado contrario. Si hacemos una traza paso a paso nos quedarían los siguientes resultados:

```
0000 0000 0000 001[0 0]       ->
0000 0000 0000 000[1 0]   P=P+S
1111 1010 0000 000[1 0]       ->
1111 1101 0000 000[0 1]   P=P+A
0000 0011 0000 000[0 1]       ->
0000 0001 1000 000[0 0]       ->
0000 0000 1100 000[0 0]       ->
0000 0000 0110 000[0 0]       ->
0000 0000 0011 000[0 0]       ->
0000 0000 0001 100[0 0]       ->


0000 0000 0000 1100 [0]   (12)
```

Finalmente obtenemos el número en binario resultante (*12 en este ejemplo*), descartando el bit extra que hemos añadido al principio del procedimiento y que se encuentra en el extremo a la derecha.

4. Ejemplo 4: Triangulo de Tartaglia

El Triángulo de Tartaglia no es un triángulo en el sentido geométrico de la palabra, sino una colección de números dispuestos en forma triangular que se obtienen de una manera muy sencilla.

```
                    1
                  1   1
                1   2   1
              1   3   3   1
            1   4   6   4   1
          1   5  10  10   5   1
        1   6  15  20  15   6   1
      1   7  21  35  35  21   7   1
    1   8  28  56  70  56  28   8   1
  1   9  36  84 126 126  84  36   9   1
1  10  45 120 210 252 210 120  45  10   1
```

Como se puede observar, en la cúspide del triángulo hay un 1, en la segunda fila hay dos 1, y las demás filas empiezan con 1 y terminan con 1, y cada número intermedio se obtiene **sumando los dos que se encuentran justo encima**.

El Triángulo de Tartaglia, llamado también **de Pascal**, es infinito. Podemos construir todas las filas que queramos. En el ejemplo de arriba hemos desarrollado once filas. Por convenio, a la primera fila, que solo contiene el 1, le llamaremos fila 0, a la segunda fila, fila 1, a la tercera, fila 2, para que así coincida el nombre de la fila con el número que viene detrás del primer 1 y antes del último 1, etc.

```
          1              fila 0
        1   1            fila 1
      1   2   1          fila 2
    1   3   3   1        fila 3
  1   4   6   4   1      fila 4
1   5  10  10   5   1    fila 5…
```

El Triángulo de Tartaglia está relacionado con el desarrollo de las potencias de un binomio y con los números combinatorios.

Si queremos desarrollar las potencias de una suma, tenemos:

$(a + b)^2 = a^2 + 2ab + b^2$

$(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$

$(a + b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$

etc...

Como se puede comprobar si nos fijamos en los coeficientes que acompañan a las potencias de a y de b, son los mismos números que los de la fila correspondiente del Triángulo. Así por ejemplo:

$$(a + b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4 = \mathbf{1}a^4 + \mathbf{4}a^3b + \mathbf{6}a^2b^2 + \mathbf{4}ab^3 + \mathbf{1}b^4$$

Es fácil también darse cuenta de que a aparece elevado a la potencia máxima y luego en cada sumando va disminuyendo la potencia (en el último ejemplo, a^4, a^3, a^2, etc.), mientras que b no aparece en el primer término, sí en el segundo, y luego va aumentando su potencia hasta acabar solo en el último término.

Cada número que aparece en el Triángulo se puede calcular independientemente del resto. Si queremos averiguar un número de la fila 20, por ejemplo, no es necesario calcular todas las filas anteriores. Cada número en realidad es un número combinatorio dado por la fórmula:

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

Veamos un ejemplo de cálculo para entender la fórmula:

$$\binom{8}{5} = \frac{8!}{5! \cdot (8-5)!} = \frac{8!}{5! \cdot 3!} = \frac{8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}{(5 \cdot 4 \cdot 3 \cdot 2 \cdot 1) \cdot (3 \cdot 2 \cdot 1)} = \frac{40320}{120 \cdot 6} = \frac{40320}{720} = 56$$

Hemos calculado el número combinatorio 8 sobre 5 y nos ha dado 56. Si nos fijamos en el Triángulo de Tartaglia de arriba del todo veremos que en la fila 8, en el quinto lugar si no contamos el primer 1, tenemos 56.

Cada número del Triángulo es el resultado de calcular el número combinatorio que corresponde a su fila y al lugar que ocupa dentro de ella. El Triángulo se puede expresar también así:

$$\binom{0}{0}$$

$$\binom{1}{0} \quad \binom{1}{1}$$

$$\binom{2}{0} \quad \binom{2}{1} \quad \binom{2}{2}$$

$$\binom{3}{0} \quad \binom{3}{1} \quad \binom{3}{2} \quad \binom{3}{3}$$

$$\binom{4}{0} \quad \binom{4}{1} \quad \binom{4}{2} \quad \binom{4}{3} \quad \binom{4}{4}$$

...

Si por ejemplo queremos calcular el término de la cúspide, cero sobre cero, podemos aplicar la fórmula, teniendo en cuenta que el factorial de cero es por definición igual a uno, 0! = 1.

$$\binom{0}{0} = \frac{0!}{0! \cdot (0-0)!} = \frac{0!}{0! \cdot 0!} = \frac{1}{1 \cdot 1} = \frac{1}{1} = 1$$

De forma análoga se pueden ir calculando todos los restantes números combinatorios y se puede comprobar que van coincidiendo con los términos del Triángulo.

El nombre del triángulo procede de su autor, el matemático italiano, **Niccolò Fontana**, apodado *Tartaglia* porque era tartamudo. Vivió entre los años 1500 y 1557, nació en Brescia, Italia y enseñó en varias universidades hasta establecerse en Florencia en 1542. Resolvió la ecuación de tercer grado y escribió tratados sobre **artillería**.