UNIVERSIDAD DISTRITAL
FRANCISCO JOSE DE CALDAS

0,0	0,1	0,2	...	0,n-2	0,n -1	0,n
1,0	1,1	1,2	...	1,n-2	1,n -1	1,n
2,0	2,1	2,2	...	2,n-2	2,n -1	2,n
...	...	...	...	...	...	...
m-2,0	m-2,1	m-2,2	...	m-2,n-2	m-2,n -1	m-2,n
m-1,0	m-1,1	m-1,2	...	m-1,n-2	m-1,n -1	m-1,n
m,0	m,1	m,2	...	m,n-2	m,n -1	m,n

Matriz dinámica de M filas y N columnas.

Matrices

2013
Transversal de Programación Básica
Proyecto Curricular de Ingeniería de Sistemas

Objetivos

- Representar conjuntos de datos mediante matrices.
- Solucionar problemas utilizando matrices.

Introducción

La utilización de matrices constituye actualmente una parte esencial en los lenguajes de programación, ya que la mayoría de los datos se introducen en los ordenadores como tablas organizadas en filas y columnas: hojas de cálculo, bases de datos, etc. También son utilizadas para resolver problemas matemáticos, por ejemplo en la resolución de sistemas de ecuaciones lineales, de las ecuaciones diferenciales y de las derivadas parciales.

1. Definición

Una matriz es una estructura de datos, o más técnicamente, un espacio de memoria que permite almacenar una colección de elementos, todos del mismo tipo. La diferencia con los arreglos está en que, en las matrices, los elementos no están organizados linealmente sino que su organización es bidimensional, es decir, en filas y columnas. Conviene imaginar una matriz como una organización de celdas de memoria, o ***casillas***, en cada una de las cuales se puede guardar un elemento de la colección. Además, es usual dibujarla como lo ilustra la figura siguiente:

	0	**1**	**2**	**3**	**4**	**5**
0						
1						
2						
3						

Esta figura representa un matriz de cuatro filas (numeradas verticalmente de 0 a 3) y seis columnas (numeradas horizontalmente de 0 a 5). En cada una de las 24 celdas o casillas se puede guardar un dato. La ***dimensión*** o tamaño de una matriz es el número filas por el número de columnas. Debe ser claro entonces que la figura anterior es la gráfica de una matriz de dimensión 4x6.

La numeración de las filas y las columnas determina que cada una de las casillas de una matriz tiene asociados dos números que la identifican de manera única. A estos números se les llama

índice de fila e ***índice de columna***, respectivamente. En el seudolenguaje, y también en C y C++, las filas y las columnas se numeran desde 0.

Los lenguajes como C y C++, permiten que el programador declare matrices de cualquier tipo y prácticamente de cualquier tamaño. En el seudolenguaje, un matriz se declara usando el siguiente formato:

<NOMBRE> : matriz [<N>][<M>] de <TIPO>

En este formato aparecen en mayúsculas y entre los caracteres **<** y **>** los componentes que el programador puede determinar. Así por ejemplo, si se quiere declarar una matriz con nombre **mat**, de dimensión **15x4** y que pueda almacenar datos de tipo **caracter**, se debe escribir la siguiente línea.

mat : matriz [15][4] de caracter

Según el formato anterior, el programador debe bautizar la matriz (ponerle un nombre significativo), debe decir cuál es su dimensión, y también debe decir de qué tipo son los elementos que almacenará.

Enseguida se dan algunos ejemplos de declaraciones de matrices.

- Si se necesita guardar la información relacionada con el tablero de un juego de *tic tac toe* (el tradicional ***triqui***), se puede declarar la siguiente matriz:

  tablero : matriz [3][3] de caracter

- Si se requiere guardar las notas que han sacado 35 estudiantes en los 5 talleres y en los 5 laboratorios del curso de Programación de Computadores se pueden declarar las siguientes matrices.

  talleres : matriz [35][5] de real
  laboratorios : matriz [35][5] de real

  Note que, en ambas matrices, cada fila guarda las notas de un estudiante del curso.

- Si se quiere guardar las letras que conforman una sopa de letras, como aquellas que vienen en los pasatiempos, se puede declarar la siguiente matriz.

  sopa : matriz [10][15] de caracter

  Note que la sopa de letras más grande que se puede guardar es de 10 filas por 15 columnas.

Los índices se crearon para permitir que el programador se pueda referir, de forma específica y directa, a una cualquiera de las casillas de la matriz, tanto para guardar un dato en esa casilla,

como para obtener el dato almacenado en ella. En el seudolenguaje, para referirse a una casilla particular de una matriz se debe seguir el siguiente formato:

<NOMBRE>[<INDICE-DE-FILA>][<INDICE-DE-COLUMNA>]

es decir, se debe escribir el nombre de la matriz seguido por el índice de fila y por el índice de columna, ambos entre paréntesis cuadrados, de la casilla que se quiere consultar.

Para los siguientes ejemplos, suponga que se declara la matriz **montos**, de la siguiente manera:

montos : matriz [6][10] de real

- La siguiente instrucción asigna o guarda el número 10,4 en la casilla de la esquina superior izquierda de esta matriz:

```
montos[0][0]:= 10,4
```

- La siguiente instrucción iterativa guarda 5,5 en cada una de las casillas de la última fila de esta matriz:

```
k:=0
MIENTRAS (k<10) HACER
        montos[5][k]:= 5,5
        k:=k+1
FIN-MIENTRAS
```

La siguiente figura muestra la matriz **montos** después de ejecutadas las instrucciones de los dos ejemplos anteriores. Las casillas vacías no tienen valores definidos.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	10,4									
1										
2										
3										
4										
5	5,5	5,5	5,5	5,5	5,5	5,5	5,5	5,5	5,5	5,5

Ejemplo

Una **matriz mágica** es una matriz cuadrada (tiene igual número de filas que de columnas) que tiene como propiedad especial que la suma de las filas, las columnas y las diagonales es igual. Por ejemplo:

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

En esta matriz las sumas son 15.

Considere el problema de construir un algoritmo que compruebe si una matriz de datos enteros es mágica o no, y en caso de que sea mágica escribir la suma. El usuario ingresa el tamaño de la matriz máximo hasta 10. Además debe guardar la suma de las filas, las columnas y las diagonales en un arreglo en el orden siguiente:

0	1	2		3	4	5		6	7
Fila 0	Fila 1	Fila 2	...	Columna 0	Columna 1	Columna 2	...	Diagonal 1	Diagonal 2

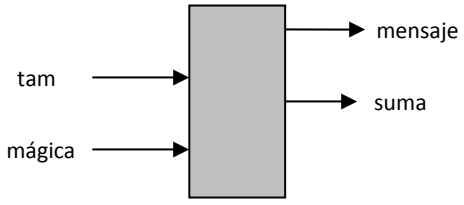
Las entradas (datos conocidos) para el algoritmo son:

- La dimensión de la matriz
- Los números que contiene la matriz

La salida esperada (datos desconocidos) es:

- La matriz es mágica o no, y si es mágica cuál es el valor de la suma.

En este problema, los arreglos son útiles para guardar los datos que conforman la matriz. Los números que contiene la matriz se pueden guardar en una variable entera.
La siguiente gráfica resume las entradas y salidas del algoritmo que se pretende diseñar. Además bautizan todas las variables mencionadas.



Las condiciones iniciales y finales se pueden expresar mediante las cláusulas REQUIERE y GARANTIZA:

REQUIERE:
La dimensión de la matriz que debe ser máximo 10x10
Cada elemento de la matriz debe ser un número entero

GARANTIZA
Muestra en pantalla si es mágica o no, y si lo es cual es el valor de la suma.

Una primera versión del algoritmo puede ser la siguiente:

Inicio

Paso 1. Leer el tamaño de la matriz

Paso 2. Leer los elementos de la matriz

Paso 3. Determinar si la matriz es mágica o no

Paso 4. Si la matriz es mágica mostrar el valor de la suma

Fin

Los pasos 1 y 2 son interacciones con el usuario que permiten capturar los datos de entrada. La versión inicial se puede refinar detallando estos pasos y además defiendo las variables para hacerlos:

Procedimiento principal

variables

i, j, aux, tam, suma: enteros //i señala las filas //j señala las columnas

magica: matriz [10][10] de enteros

Inicio

escribir("Por favor digite el número de filas de la matriz (entre 2 y 10): ")

leer(tam)

para (i=0 hasta tam-1) hacer

para(j=0 hasta tam-1) hacer

```
            escribir("Por favor digite el dato en la posición")
            escribir(i,j)
            leer(magica[i][j])
        fin_para
    fin_para
```

Paso 3. Determinar si la matriz es mágica o no

Paso 4. Si la matriz es mágica mostrar el valor de la suma

Se puede observar que el primer **ciclo para** tiene como contador la variable i, esto indica que se llenará la matriz por filas, el segundo **ciclo para** que tiene como contador la variable j, recorrerá la fila columna a columna para ubicar allí el dato correspondiente.

La parte nuclear de la solución es el paso 3. En este problema en particular se sabe que el número de filas y de columnas es igual y que hay dos diagonales. Para el ejemplo mostrado al inicio sería 3 filas, 3 columnas y dos diagonales. Para almacenar las sumas en un arreglo este tendrá una dimensión de 2*tam+2. La declaración del arreglo sumas es:

sumas: arreglo [22] de enteros

Ahora para calcular las sumas se puede hacer lo siguiente:

```
Paso 3.1: Inicializar el arreglo de sumas en cero
Paso 3.2: Sumar fila por fila, columna por columna y las diagonales y guardar
su valor en el arreglo.
para(i=0 hasta 2*tam+2) hacer
        sumas[i]:=0
fin_para
//Sumas correspondientes a las filas
para(i=0 hasta tam-1) hacer
        para(j=0 hasta tam-1) hacer
                sumas[i]=magica[i][j]+sumas[i]
        fin_para
```

```
fin_para

//Sumas correspondientes a las columnas

para(j=0 hasta tam-1) hacer

        para(i=0 hasta tam-1) hacer

                sumas[j+tam]=magica[i][j]+sumas[j+tam]

        fin_para

fin_para

//Sumas correspondientes a las diagonales

para(i=0 hasta tam-1) hacer

        sumas[2*tam]=magica[i][i]+sumas[2*tam]

fin_para

para(i=0 hasta tam-1) hacer

         sumas[2*tam+1]=magica[i][(tam-1)-i]+sumas[2*tam+1];

fin_para
```

Paso 4: Para determinar si la matriz es mágica se va a recorrer y comparar el vector sumas, si en algún momento se encuentra un valor diferente se muestra en pantalla que la matriz no es mágica y se lleva el contador **i** más allá del final del arreglo, si por el contrario se llega al final del arreglo, es decir que todo este contiene el mismo valor y la matriz si cumple con las características evaluadas, se muestra en pantalla que la matriz es mágica.

```
//Comparar el vector suma y muestra el resultado

int con=0;

con=sumas[0];

para(i=1 hasta 2*tam+1) hacer

      si(con<>sumas[i])

            escribir("la matriz no es mágica)
```

```
                i=2*tam+3;
        fin_si
    fin-para
    si(i=2*tam+2)
        escribir("la matriz es mágica y la suma es:")
        escribir(con);
    fin_si
```

El algoritmo completo se presenta enseguida.

```
    Procedimiento principal
    variables
    i, j, aux, tam, suma: entero          //i señala las filas   //j señala las columnas
    con=0: entero
    magica: matriz [10][10] de enteros
    sumas: arreglo [22] de enteros
    Inicio
    escribir("Por favor digite el número de filas de la matriz (entre 2 y 10): ")
    leer(tam)
    para (i=0 hasta tam-1) hacer
        para(j=0 hasta tam-1) hacer
            escribir("Por favor digite el dato en la posición")
            escribir(i,j)
            leer(magica[i][j])
        fin_para
```

```
fin_para
para(i=0 hasta 2*tam+2) hacer
        sumas[i]:=0
fin_para
//Sumas correspondientes a las filas
para(i=0 hasta tam-1) hacer
        para(j=0 hasta tam-1) hacer
                sumas[i]=magica[i][j]+sumas[i]
        fin_para
fin_para
//Sumas correspondientes a las columnas
para(j=0 hasta tam-1) hacer
         para(i=0 hasta tam-1) hacer
                  sumas[j+tam]=magica[i][j]+sumas[j+tam]
          fin_para
fin_para
//Sumas correspondientes a las diagonales
para(i=0 hasta tam-1) hacer
        sumas[2*tam]=magica[i][i]+sumas[2*tam]
fin_para
para(i=0 hasta tam-1) hacer
        sumas[2*tam+1]=magica[i][(tam-1)-i]+sumas[2*tam+1];
fin_para
con=sumas[0];
```

```
para(i=1 hasta 2*tam+1) hacer
        si(con<>sumas[i])
                escribir("la matriz no es mágica)
                i=2*tam+3;
        fin_si
fin-para
si(i=2*tam+2)
        escribir("la matriz es mágica y la suma es:")
        escribir(con);
fin_si
fin-procedimiento
```

Nota: ver anexo

Problemas para desarrollar en clase

1. El dueño de un restaurante entrevista a cinco clientes de su negocio y les pide que califiquen de 1 a 10 los siguientes aspectos: (1 es pésimo y 10 es excelente o inmejorable)

- Atención de parte de los empleados
- Calidad de la comida
- Justicia del precio (el precio que pagó le parece justo?)
- Ambiente (muebles cómodos?, música adecuada?, iluminación suficiente?, decoración, etc.)

Escriba un algoritmo que pida las calificaciones de los cinco clientes a cada uno de estos aspectos, y luego escriba el promedio obtenido en cada uno de ellos. La lista debe aparecer ordenada del aspecto mejor calificado al peor calificado.

2. En una hacienda hay un hato que se compone de **N** vacas. Diseñe un algoritmo que guarde en una matriz de dimensión **7xN** la producción de leche diaria (en litros) de cada una de las vacas, durante una semana. Además, el algoritmo debe calcular la producción total del hato en cada uno de los siete días, y el número de la vaca que dio más leche en cada día.

Ejercicios para desarrollar en casa

1. Los siguientes ejercicios tienen como propósito que usted escriba ciclos que recorran la matriz completa o partes de ella. Suponga que se ha definido una constante positiva entera **N** y una matriz **mat**, de dimensión **NxN**.

   - Escriba un algoritmo que ponga cero en ambas diagonales de la matriz.
   - Escriba un algoritmo que ponga cero en la primera y la última fila, y en la primera y la última columna de la matriz.
   - Escriba un algoritmo que llene de números la matriz de tal forma que mat[i][j] sea igual a i+j.
   - Escriba un algoritmo que llene la diagonal principal de la matriz con los números 1,2,3,...N. La diagonal principal de una matriz está formada por las casillas en las cuales el índice de fila y de columna son iguales.
   - Escriba un algoritmo que llene todas las filas pares con los números 1,2,3,...N, y las filas impares con los números N,N-1,N-2,...1.

2. Diseñe un algoritmo que permita guardar en un arreglo las sumas de las filas de una matriz. Esto es, la suma de los elementos de la primera fila deberá quedar guardada en la primera posición del arreglo, la suma de los elementos de la segunda fila en la segunda posición, y así sucesivamente para todas las filas de la matriz. **La máxima dimensión** de la matriz es 100x50 (100 filas y 50 columnas) y la del vector es 100.

   Por ejemplo, si el usuario ingresa la siguiente matriz de 3x5 (3 filas, 5 columnas)

3,5	**6,5**	**30**	**8,2**	**0**
4	**0**	**-1**	**3,6**	**1,4**
10	**-1,5**	**3,4**	**6,6**	**2**

   El resultado sería un arreglo siguiente:

48,2	8	20,5

   porque

   3,5 + 6,5 + 30 + 8,2 + 0 = 48,2
   4 + 0 + (-1) + 3,6 + 1,4 = 8 y
   10 + (-1,5) + 3,4 + 6,6 + 2 = 20,5

3. En álgebra lineal las matrices son tema central. Sobre ellas se definen varias operaciones, como por ejemplo:

   - La suma de dos matrices. Si A y B son matrices de igual dimensión, la matriz C=A+B se calcula haciendo que C[i][j] = A[i][j]+B[i][j], para todo i y j válidos.

- La traspuesta de una matriz. Si A es una matriz de dimensión NxM, la matriz B=At se calcula haciendo que B[i][j] = A[j][i], para todo i y j válidos. Note que esto quiere decir que las filas se convierten en columnas y que la dimensión de B es MxN.
- La traza de una matriz cuadrada. Si A es una matriz de dimensión NxN, la matriz traza es la suma de todos los elementos de la diagonal principal.
- La multiplicación de dos matrices. Si A y B son matrices de dimensiones nxm y mxk, respectivamente, la matriz C=A*B , de dimensión nxk, se calcula haciendo que:

$$C[i][j] = \sum_{p=0}^{m-1} A[i][p] * B[p][j]$$

Especifique y escriba un algoritmo para cada una de estas operaciones.

Anexos

En esta sección se mostrara como es la codificación de la teoría vista en clase, para ello se utilizara como herramienta de codificación: C++.

Codificación en C++ de arreglos y matrices

	Seudocódigo	C++
Matriz	<NOMBRE> : matriz [<N>][<M>] de <TIPO>	<TIPO> <NOMBRE>[<N>][<M>];

- **Ejemplo En C++**

```
#include <iostream.h>

#include <stdlib.h>

#include <conio.h>

   /****Este programa lee los datos de una matriz 3x4 y muestra en pantalla la suma de los
                                    datos de cada fila *****/
main(){

     int matriz[3][4];

     int arreglo[3];

     int i,j;

     //Ingreso de los datos
```

```
for (i=0;i<3;i++){

   for (j=0;j<4;j++){

     cout << "Ingrese el numero entero correspondiente a la posicion ["<<i<<"] ["<<j<<"]:";
     cin >> matriz[i][j];

   }

}
```

//Muestra en pantalla la matriz ingresada

```
cout << "\nLa matriz que usted ingreso es: \n\n";

for (i=0;i<3;i++){

   for (j=0;j<4;j++){

      cout << matriz[i][j]<<" ";

   }

   cout << "\n";

}
```

//Suma los datos de cada fila

```
for (i=0;i<3;i++){

   arreglo[i]=0;

   for (j=0;j<4;j++){

      arreglo[i]=arreglo[i]+matriz[i][j];

   }

}
```

//Muestra en pantalla los resultados

```
for (i=0;i<3;i++){

   cout << "\nLa suma de los datos de la fila "<<i<<" es: " << arreglo[i];
```

```
    }
    getch();
}
```

//Programa de la matriz mágica

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    int i, j, aux, tam, suma;
    int con=0;
    int magica [10][10];
    int sumas [22];
```

//Ingreso de datos

```
 cout<< "Por favor digite el numero de filas de la matriz (entre 2 y 10)";
 cin>>tam;
 for (i=0; i<=tam-1; i++){
    for (j=0; j<=tam-1; j++){
       cout<<"Por favor digite el dato de la posicion: "<< i << ","<< j <<": ";
       cin>> magica [i][j];
    }
 }
```

//Inicializar el arreglo sumas

```
 for (i=0; i<=2*tam+2; i++ ){
```

```
    sumas [i]=0;
 }
//filas
 for (i=0; i<= tam-1; i++){
    for (j=0; j<=tam-1; j++){
       sumas[i]=magica[i][j]+sumas[i];
    }
}
//columnas
  for (j=0; j<= tam-1; j++){
    for (i=0; i<=tam-1; i++){
       sumas[i+tam]=magica[i][j]+sumas[i+tam];
    }
 }
//Diagonales
 for (i=0; i<=tam-1; i++){
    sumas[2*tam]=magica [i][i]+sumas[2*tam];
 }
 for (i=0; i<=tam-1; i++){
    sumas[2*tam+1]=magica [i][(tam-1)-i]+sumas[2*tam+1];
 }
 con= sumas[0];
//imprimir arreglo sumas
 cout<<"arreglo suma \n" ;
```

```
 for (i=0; i<= 2*tam+1; i++){
    cout<< i << "= " <<sumas [i] << endl;
 }
 // Verificar la suma en el arreglo
 for (i=1; i<= 2*tam+1; i++){
    if (con != sumas [i]){
          cout<< "la matriz no es magica \n";
          i=2*tam+3;
     }
}
 if (i == 2*tam+2)
    cout << "la matriz es mágica y la suma es: "<< con << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Lectura de Profundización

- Texto tomado de http://dis.unal.edu.co/~programacion/book/modulo3.pdf

Imágenes:

Las imágenes utilizadas en este documento fueron tomadas de www.google.com

Fuentes:

- http://www.slideshare.net/Edw1a/guia8-programacion-14671287
- http://www.slideshare.net/gonmrod/vectores-matrices-i
- http://www.atc.us.es/asignaturas/fi/curso_de_c/Array_bidimensional_o_matriz.html
- http://www2.caminos.upm.es/departamentos/matematicas/Fdistancia/PIE/matlab/temas matlab/TEMA%203.pdf
- http://trevinca.ei.uvigo.es/~jgarcia/TO/cursilloCpp/5_manejo_matrices.html