Matrices.

Programación I – Laboratorio I.
Tecnicatura Superior en Programación.
UTN-FRA

Autores: *Pablo Gil*
Hector Farina

Revisores: *Ing. Ernesto Gigliotti*
Lic. Mauricio Dávila

Versión : 1

Índice de contenido

1 Matrices

1.1 Introducción

En el lenguaje C es posible declarar un array que contenga más de una dimensión, se conoce como **matriz** al array de dos dimensiones, el mismo puede ser pensado como una tabla con filas y columnas:

[0][0]	[0][1]	[0][2]
[1][0]	[1][1]	[1][2]

El mapa de memoria donde se almacenarán los valores tendrá direcciones de memoria consecutivas, agrupando las columnas de una misma fila:

Dirección de memoria variable	Valor de: [0][0]
Dirección+Z	Valor de: [0][1]
Dirección+2Z	Valor de: [0][2]
Dirección+3Z	Valor de: [1][0]
Dirección+4Z	Valor de: [1][1]
Dirección+5Z	Valor de: [1][2]

Siendo Z el tamaño del tipo de dato del que se construyó la matriz.

1.2 Declaración de una matriz

La declaración de arrays de más de una dimensión se realiza de forma parecida a la de una dimensión, la sintaxis de la declaración de un array multidimensional es:

```
tipo nombre[tam1][tam2]...[tamN];
```

Y su indexación, etc., se realiza de forma similar al array unidimensional.
Veamos un ejemplo:

```
float matriz[2][3];
int i,j;

for(i=0;i<2;i++)
{
	for(j=0;j<3;j++)
	{
		printf("M[%d][%d]: ",i,j);
		scanf("%f",&matriz[i][j]);
	}
}
```

1.3 Matrices con cadenas de caracteres

Sabemos que para guardar varios nombres necesitamos varios vectores o sea una matriz. La cantidad de filas que tenga la matriz va a depender de la cantidad de nombres que se quieran ingresar y la cantidad de columnas esta asociada directamente con el número máximo de letras del texto a guardar.
Tener en cuenta que para cargar una matriz con cadenas de caracteres debemos posicionarnos al comienzo de cada fila, ya que es a partir de allí donde se carga la cadena.

Por lo tanto cuando se este cargando la matriz debemos movernos de fila en fila y la forma de posicionarnoa al comienzo de cada fila es:

`mat[i]` o sea la dirección de comienzode la fila "i"

Si estamos trabajando con una matriz de caracteres definida como:

```
char mat [FIL][COL];
```

Al escribir mat[0] se indica al dirección de comienzo de la primer fila , mat[1] indica la dirección de comienzo de la segunda fila y en forma genérica mat [i] es la dirección de comienzo de la fila i.

Significa que cuando debemos cargar una matriz se usará un ciclo *for* y se cargarán fila por fila de la siguiente forma;

```
for (i=0;i<FIL;i++)
{
	printf("\nIngrese nombre: ");
	gets(mat[i]);
}
```

Ejemplo tipo: Cargar 20 nombres, mostrarlos ordenados alfabéticamente y luego mostrar el nombre que más letras tiene.

```
#include <stdio.h>
#include <string.h>
#define FIL 20
#define COL 25

void main(void)
{
	int i,j,x,lmax=0,imax;
	char mat[FIL][COL],aux[COL];

	for(i=0;i<FIL;i++)
	{
		printf("Ingrese nombre: ");
		gets(mat[i]);
	}

	// Ordeno alfabeticamente la matriz

	for(i=0;i<FIL-1;i++)
	{
		for(j=i+1;j<FIL;j++)
		{
			if((strcmp(mat[i],mat[j]))>0)
			{
				strcpy(aux,mat[i]);
				strcpy(mat[i],mat[j]);
				strcpy(mat[j],aux);
			}
		}
	}

	// Busqueda del nombre mas largo


	for(i=0;i<FIL;i++)
	{
		x=strlen(mat[i]);
		if(x>lmax)
		{
			lmax=x;
			imax=i;
		}
	}
	printf("El nombre mas largo es %s",mat[imax]);
}
```

1.4 Ordenamiento de una matriz de cadenas

El algoritmo para ordenar una matriz que contiene cadenas de caracteres tiene el mismo formato que el ordenamiento de un vector.
La razón es simple, una matriz que contiene cadenas de caracteres no es más que un vector donde cada elemento (una fila de la matriz) es una palabra.
Entonces en resumen debemos considerar que ordenamos un vector de palabras y lo tratamos como tal.

Como se ve en el ejemplo anterior, la forma del ordenamiento es exactamente la misma que se usa para ordenar un vector, solo cambia como se tratan los distintos tipos de datos.

Por ejemplo cuando en el ordenamiento de números enteros se llegaba a la comparación, se analizaba si el elemento "i" era mayor o menor que el "j". Con cadenas de caracteres quien hace la comparación es la función strcmp a la cual le pasamos la fila "i" y la fila "j".

En el ordenamiento de números enteros se usaba el auxiliar y las asignaciones para realizar el cambio de datos. Con cadenas de caracteres el tratamiento es el mismo, la diferencia es que para realizar asignaciones se debe usar la función strcpy y que el auxiliar debe ser una cadena.