	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcon

Asignatura: Fundamentos de Programación

Grupo: 3

No. de practica: 10

Integrantes:

Jonan Gómez Mendoza 5641
Franco Inglés Carolina 2836
Lucia Nicole Rosette Hernández *2768*

Semestre: *1*

Fecha de entrega: *octubre 28, 2019*

Observaciones: Muy bien

CALIFICACIÓN: **10**

PRÁCTICA #11

- **Objetivo:**

Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

- **Introducción**:

ARREGLO UNIDIMENSIONAL

Un array (unidimensional, también denominado vector) es una variable estructurada formada de un número "n" de variables simples del mismo tipo que son denominadas los componentes o elementos del array. El número de componentes "n" es, entonces, la dimensión del array. De igual manera que en matemáticas, decimos que "A" es un vector de dimensión "n".

El formato para declarar un array unidimensional es:
tipo nombre[n];
donde: n >= 1
Para acceder a un elemento del array: nombre[i];
donde: 0 <= i < n
Por ejemplo, la declaración:
int A[4];

Define un array de tipo entero de dimensión 4. Y ya podríamos acceder al primer componente del array por medio de: A[0], al segundo elemento por: A[1] y al último elemento por A[3].

En C, un array se utiliza básicamente cuando queremos tener, por ejemplo, una secuencia de números reunidos en una sola variable.

ARREGLO MULTIDIMENSIONAL

Un array en C puede tener una, dos o más dimensiones. Por ejmplo, un array de dos dimensiones también denominado matriz, es interpretado como un array (umidimensional) de dimensión "f" (número de filas), donde cada componente es un array (unidimensional) de dimensión "c" (número de columnas). Un array de dos dimensiones, contiene, pues, "f*c" componentes.

El formato para declarar un array multidimensionales:

int nombre[f][c]...;

donde: f,c...>= 1;
Para acceder a un elemento del array multidimensional: nombre[i][j];

donde:0<=i<f; 0<=j<c;

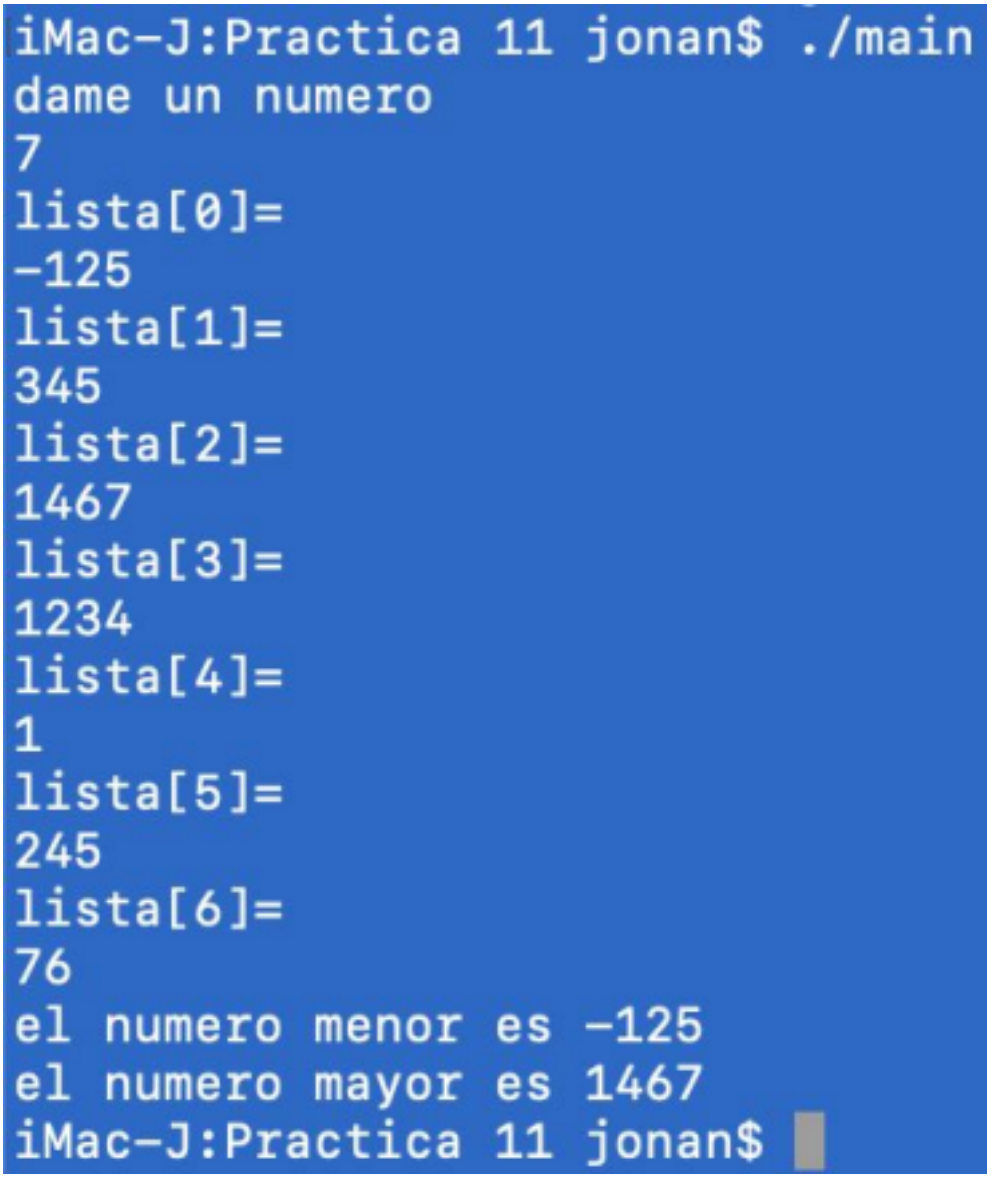
Durante la declaración de un array multidimensional también podemos inicializar sus componentes indicando la lista de los valores entre llaves. En el interior de la lista, los componentes de cada línea del array son encerrados nuevamente entre llaves. Para hacer más cara la visibilidad de los elementos del array, podemos indicarlos en varias líneas.

Sin embargo, es mucho más conveniente anidar dos ciclos para inicializar un array de dos dimensiones.

ACTIVIDAD 1

```c
#include<stdio.h>
int main (){
    int num;
    printf("dame un numero\n");
    scanf("%i",&num);
int lista[num];
//este for para llenar la lista
for (int i=0;i<num;i++){
    printf("lista[%i]=\n",i);
    scanf("%i",&lista[i]);
}
int menor;//menor va a ser nuestro numero menor
menor=lista[0];//por el momento vamos a suponer que el primer elemto de la lista
//Es el numero menor
//este for para sacar el numero menor
for (int i=1;i<num;i++){
    if (lista[i]<menor){//se compara el siguiente elemento de la lista (porque i=1)
        //con el numero menor relativo (que la primera vez es lista[0])
        menor=lista[i]; //si el suguiente elemento es menor este pasara a ser el
        //nuevo menor relativo
        //seguira hasta que se compar con todos los elementos de la lista
    }
}
int mayor;
mayor=lista[0];
//este for para sacar el numero mayor
for (int i=1;i<num;i++){
    if (lista[i]>mayor){
        mayor=lista[i];
    }
}
printf("el numero menor es %i\n",menor);
printf("el numero mayor es %i\n",mayor);
}
```

COMPROBACIÓN



ACTIVIDAD 2

Esta es la primer parte del código donde se asignan los valores de las listas.

```c
#include<stdio.h>
int main(){
    int N,M; //estas seran las dos variables que determinan
    //el tamaño de las matrices
    printf("Las matrices tendran una dimension NxM\n");
    printf("introduzca el valor de N: ");
    scanf("%i",&N);
    printf("introduzca el valor de M: ");
    scanf("%i",&M);
    //Tenemos 3 matrices las dos que se va a sumar y la que va a contener el
        resultado
    int sumando1[N][M],sumando2[N][M], resultado[N][M];
    //el primer for sirve para asignarl el subindice de la fila
    for(int i=0;i<N;i++){
        //el segundo for es para asignar la columna
        for(int j=0;j<M;j++){
            //con esta accion se llena la primer lista con los ij especificados
            printf("introduzca sumando1[%i][%i]: ",i,j);
            scanf("%i",&sumando1[i][j]);
            }
    }
    //sigue otros dos for iguales para la otra lista
    for(int i=0;i<N;i++){
        for(int j=0;j<M;j++){
            printf("introduzca sumando2[%i][%i]: ",i,j);
            scanf("%i",&sumando2[i][j]);
            }
    }
    //como la suma de dos listas es elemento a elemento significa que comparten los
    //mismos subindices ij asi que usamos los mismos dos for
    for(int i=0;i<N;i++){
        for(int j=0;j<M;j++){
            //se suman las dos listas con los mismos subindices
            //el resultado se guarda en otra lista con los mismos subindices
            resultado[i][j]=sumando1[i][j]+sumando2[i][j];
        }
    }
```

Esta es la segunda parte donde se muestran las listas

```
    //se muestra la primera lista
    printf("sumando1:\n");
    for(int i=0;i<N;i++){
        printf("|");
        for(int j=0;j<M;j++){
            if (j==M-1){
                printf("%i",sumando1[i][j]);
                break;
            }
            printf("%i\t",sumando1[i][j]);
        }
        printf("|");
        //este print es dejar un renglon de espacio entre las filas
        printf("\n");
    }//se muestra la segunda lista
    printf("+\nsumando2:\n");
    for(int i=0;i<N;i++){
        printf("|");
        for(int j=0;j<M;j++){
            if (j==M-1){
                printf("%i",sumando2[i][j]);
                break;
            }
            printf("%i\t",sumando2[i][j]);
        }
        printf("|");
        printf("\n");
    }
    printf("=\n");
    //este for es para imprimir el resultado
    for(int i=0;i<N;i++){
        printf("|");
        for(int j=0;j<M;j++){
            if (j==M-1){
                printf("%i",resultado[i][j]);
                break;
            }
            printf("%i\t",resultado[i][j]);
        }
        printf("|");
        printf("\n");
    }
    return 0;
}

```

COMPROBACIÓN

```
Las matrices tendran una dimension NxM
introduzca el valor de N: 3
introduzca el valor de M: 2
introduzca sumando1[0][0]: 1
introduzca sumando1[0][1]: 2
introduzca sumando1[1][0]: 3
introduzca sumando1[1][1]: 4
introduzca sumando1[2][0]: 5
introduzca sumando1[2][1]: 6
introduzca sumando2[0][0]: 7
introduzca sumando2[0][1]: 8
introduzca sumando2[1][0]: 9
introduzca sumando2[1][1]: 10
introduzca sumando2[2][0]: 11
introduzca sumando2[2][1]: 12
sumando1:
|1      2|
|3      4|
|5      6|
+
sumando2:
|7      8|
|9      10|
|11     12|
=
|8      10|
|12     14|
|16     18|
```

- **Conclusión:** Conocimos la importancia y utilidad que tiene saber el uso de los arreglos ya que con estos podemos facilitarnos el trabajo cuando en el programa es necesario que en la resolución de problemas agrupemos datos del mismo tipo. De igual manera conocer las diferencias entre unidimensional y multidimensional y cuando podemos usar cada una de ellas.