



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Alejandro Pimentel Alarco

Profesor:

Fundamentos de Programación

Asignatura:

3

Grupo:

28

No de Práctica(s):

Badillo Ruiz Evangelina

Integrante(s):

*No. de Equipo de
cómputo empleado:*

36

3069

No. de Lista o Brigada:

1

Semestre:

28 de Octubre del 2019

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____



Práctica 11

BADILLO RUIZ EVANGELINA NO.CUENTA 3069

OBJETIVO

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 Unidimensionales:

Un arreglo unidimensional es un tipo de datos estructurado que está formado por una colección finita y ordenada de datos del mismo tipo. Es la estructura natural para modelar listas de elementos iguales. Los datos que se guarden en los arreglos todos deben ser del mismo tipo.

Un arreglo es un objeto que , cuando es creado por el compilador, se le asignan automáticamente valores iniciales predeterminados a cada uno de sus elementos,

de acuerdo a los siguientes criterios:

- ☐ Si el tipo del arreglo es numérico, a sus elementos se les asigna el valor cero.
- ☐ Si el tipo del arreglo es char, a sus elementos se les asigna el valor '\u0000'.
- ☐ Si el tipo del arreglo es bool, a sus elementos se les asigna el valor false.
- ☐ Si el tipo del arreglo es una clase, a sus elementos se les asigna el valor null.

Arreglo Unidimensional

Finito: Contiene N elementos.
Homogéneo: del mismo tipo.
Ordenado: con una posición.
Referenciado: con un índice.

PRIMERO	SEGUNDO	TERCERO	CUARTO	QUINTO	SEXTO	SÉPTIMO	...	ENÉSIMO
C_1	C_2	C_3	C_4	C_5	C_6	C_7	...	C_N
i_0	i_1	i_2	i_3	i_4	i_5	i_6	...	C_{N-1}

► Arreglo bidimensionales

- Un arreglo bidimensional tiene dos dimensiones y es un caso particular de los arreglos multidimensionales. Un arreglo bidimensional también puede definirse como: "Arreglo De Arreglos", "Matrices", "Vector de Vectores", "Conjunto de Vectores".
- Este tipo de arreglos es un tipo de dato estructurado, finito ordenado y homogéneo. El acceso a ellos también es en forma directa por medio de un par de índices.
- Los arreglos bidimensionales se usan para representar datos que pueden verse como una tabla con filas y columnas.

Arreglos bidimensionales


¿Cómo se recorren los arreglos bidimensionales con ciclos repetitivos?

matriz	0	1	2
0	10	5	1
1	23	9	7

Es necesario utilizar dos ciclos repetitivos para recorrer un arreglo bidimensional, uno para las filas y uno para las columnas.

Un ciclo se incluye dentro del otro.

```
for (int i=0; i<2 ; i++)  
    for (int j=0; j<3 ; j++)  
        System.out.print(matriz[i][j]);
```

Para el ejemplo del arreglo llamado **matriz**, los datos que imprime son: 

10
5
1
23
9
7

Arreglo unidimensional

Lo que trataba de mostrarnos el programa es como funciona un arreglo unidimensional, que es lo mas común donde usamos lista de números o caracteres.

```
1  #include<stdio.h>
2  #define TAMANO 5
3  int main() {
4      int a, lista[TAMANO] = { 23, 5, 34, 19, 0 };
5      printf("Lista: \n");
6
7      for(a=0; a<TAMANO-1; a++) {
8          printf("%i, ", lista[a]);
9      }
10     printf("%i \n", lista[TAMANO-1]);
11     return 0;
12 }
13
```

```
C:\Users\evang\OneDrive\Escritorio\arreglo1.exe
Lista:
23,5,34,19,0

-----
Process exited with return value 0
Press any key to continue . . .
```

Arreglo bidimensional

Lo que trataba de mostrarnos el programa es como funciona un arreglo bidimensional, es decir imprimir filas x columnas

```
1 #include<stdio.h>
2 #define DIM 3
3 int main(int argc, char* argv[]) {
4     int a, c, matriz[DIM][DIM] = {{ 23, 5, 34, 19, 0},
5     { 8, 46, 22}, { 3, 9, 12}};
6     printf("Matriz: \n");
7     for(a=0; a<DIM; a++) {
8         for(c=0; c<DIM; c++) {
9             printf("%i \t", matriz[a][c]);
10        }
11        printf("\n");
12    }
13    return 0;
14 }
```

```
C:\Users\evang\OneDrive\Escritorio\arreglo2.exe
Matriz:
23      5      34
8       46     22
3       9      12

-----
Process exited with return value 0
Press any key to continue . . .
```

Actividad 1

```
1 #include<stdio.h>
2 int main() {
3     int chico, grande, c, arreglo, a, lista[arreglo];
4     printf("Ingrese el valor de su Arreglo\n");
5     scanf("%i", &arreglo);
6     printf("Lista:\n");
7     for(a=0; a<=arreglo-1; a++) {
8         printf("Ingrese un numero: \n");
9         scanf("%i", &lista[a]);
10    }
11 }
12 chico=lista[0];
13 for(a=1; a<arreglo; a++) {
14     if(lista[a]<chico) {
15         chico=lista[a];
16     }
17 }
18 grande=lista[0];
19 for(a=1; a<arreglo; a++) {
20     if(lista[a]>grande) {
21         grande=lista[a];
22     }
23 }
24 printf("El numero menor es: %i \n", chico);
25 printf("El numero mayor es: %i \n", grande);
26 return 0;
27 }
```

C:\Users\evang\OneDrive\Escritorio\Project1.exe

Ingrese el valor de su Arreglo

10

Lista:

Ingrese un numero:

2

Ingrese un numero:

3

Ingrese un numero:

4

Ingrese un numero:

5

Ingrese un numero:

6

Ingrese un numero:

1

Ingrese un numero:

2

Ingrese un numero:

3

El numero menor es:2

El numero mayor es:4

Process exited with return value 0

Press any key to continue . . .

```
#include<stdio.h>
int main() {
    int chico,grande,c,arreglo,a,lista[arreglo];/*declarar las variables y sus tipo de variable*/
    printf("Ingrese el valor de su Arreglo\n"); /*pedirle al usuario el tamaño de su lista*/
    scanf("%i", & arreglo); /*escanear la lista*/
    printf("Lista:\n");
    for(a=0; a<=arreglo-1;a++){/*usamos for para que dependiendo el tamaño de la lista, registre "n" , los números
que desea el usuario*/
        printf("Ingrese un numero:\n");
        scanf("%i",& lista[a]); /*va almacenando en la lista los números que el usuario ingresa*/
    }
    chico=lista[0]; /*comprobación del numero mas pequeño de la lista*/
    for(a=1;a<arreglo;a++){/*usamos un for para que vaya recorriendo la lista*/
        if(lista[a]<chico){/*usamos un condicional para que compruebe cual es el numero mas pequeño de la lista*/
            chico=lista[a]; /*una vez que lo encuentra se almacena en la variable chico*/
        }
    }
    grande=lista[0]; /* comprobación del numero mas grande de la lista */
    for(a=1;a<arreglo;a++){/* usamos un for para que vaya recorriendo la lista */
        if(lista[a]>grande){/* usamos un condicional para que compruebe cual es el numero mas grande de la lista
*/
            grande=lista[a]; /* una vez que lo encuentra se almacena en la variable grande*/
        }
    }
    printf("El numero menor es:%i\n",chico); /*imprimos el numero mas pequeño de la lista*/
    printf("El numero mayor es:%i\n",grande); /*imprimos el numero mas grande de la lista*/
    return 0;
}
```


► Actividad 2

```
1 #include <stdio.h>
2 /* este programa suma 2 matrices de NxM, que el usuario desea que sean sus tamaños */
3 int main() {
4     int N,M; /*estas seran las dos variables que dan el tamaño de las matrices*/
5     printf("Las matrices son de dimension NxM\n");
6     printf("Escribe el valor de N:");
7     scanf("%i", &N);
8     printf("Escribe el valor de M:");
9     scanf("%i", &M);
10    /*declaramos 3 matrices, las 2 primeras son las que el usuario ingrese y la tercera el resultado de sumar ambas*/
11    int i=0,j=0, matriz1[N][M], matriz2[N][M], suma[N][M];
12    /*1er for sirve para darle un subindice a la fila*/
13    for(i;i<N;i++){
14        /*este for sirve para asignarle una columna*/
15        for(j;j<M;j++){
16            printf("introduzca la matriz1\n");
17            scanf("%i", &matriz1[i][j]);
18        }
19    }
20    /*3er for sirve para darle un subindice a la fila*/
21    for(i;i<N;i++){
22        /*este for sirve para asignarle una columna*/
23        for(j;j<M;j++){
24            printf("introduzca la matriz2");
25            scanf("%i", &matriz2[i][j]);
26        }
27    }
28    /*4to for sirve para darle un subindice a la fila*/
29    for(i;i<N;i++){
30        /*este for sirve para asignarle una columna*/
31        for(j;j<M;j++){
32            /*operacion de sumar ambas matrices, almacenadas*/
33            suma[i][j] = matriz1[i][j] + matriz2[i][j];
34        }
35    }
36    printf("matriz1:\n");
37    for(i;i<N;i++){
38        printf(" |");
39        for(j;j<M;j++){
40            if(j==M-1){
41                printf(" %i", matriz1[i][j]);
42                break;
```

```
36 printf("matriz1:\n");
37 for(i;i<N;i++){
38     printf("|");
39     for(j;j<M;j++){
40         if(j==M-1){
41             printf("%i",matriz1[i][j]);
42             break;
43         }
44         printf("%i\t",matriz1[i][j]);
45     }
46     printf("|");
47     printf("\n");
48 }
49 printf("matriz2:\n");
50 for(i;i<N;i++){
51     printf("|");
52     for(j;j<M;j++){
53         if(j==M-1){
54             printf("%i",matriz2[i][j]);
55             break;
56         }
57         printf("%i\t",matriz2[i][j]);
58     }
59     printf("|");
60     printf("\n");
61 }
62 printf("=\n");
63 for(i;i<N;i++){
64     printf("|");
65     for(j;j<M;j++){
66         if(j==M-1){
67             printf("%i",suma[i][j]);
68             break;
69         }
70         printf("%i\t",suma[i][j]);
71     }
72     printf("|");
73     printf("\n");
74 }
75 return 0;
76 }
```

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

- ❖ En esta practica aprendimos hacer un arreglo unidimensional, el cual se traba de declarar una lista que almacenara “n” valores que ingresara el usuario. Después revisaría cual de todos esos números era el menor y mayor.
- ❖ El segundo arreglo que hicimos fue uno bidimensional, el cual se trato de la suma de 2 matrices .Cuando el usuario tenia que ingresar el numero de filas y el numero de columnas de ambas matrices. El programa tenia que registrar dos lista con “n” filas y “n” columnas , a partir de eso, el programa registraba los números que el usuario deseaba ingresar. Por ultimo sumaba ambas matrices.
- ❖ Aprendí a utilizar las listas como medio me almacenamiento de información para ser aplicado a diferentes resolución de problemas.
- ❖ Utilice conocimientos como el tipo de variable, secuencias y funciones, para agilizar mis programas.