



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.C ALEJANDRO ESTEBAN PIMENTEL ALARCON

Asignatura: FUNDAMENTOS DE PROGRAMACION

Grupo: 3

No de Práctica(s): 11

Integrante(s): MARTINEZ TRUJILLO CINTHYA ANDREA
GUTIERREZ ACOSTA CLAUDIA

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 2720
2881

Semestre: 2020-1

Fecha de entrega: 28 / OCTUBRE / 2019

Observaciones: Bien, pero tengan cuidado al dar datos al programa, si el programa espera recibir números, se confunde cuando recibe comas y otras cosas que no sean números.

CALIFICACIÓN: 10

ARREGLOS UNIDIMENSIONALES Y MULTIDIMENSIONALES

INTRODUCCION

En esta practica conoceremos los arreglos y la importancia que tienen al momento de programar, la manipulacion de datos de manera flexible.

Es importante contar con un indice y saber que existen los arreglos unidimensionales, asi como los multidimensionales. Este tipo de herramienta nos ayudara a que los codigos se efectuen de manera correcta.

OBJETIVO

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

ARREGLOS UNIDIMENSIONALES



ARREGLOS UNIDIMENSIONALES

```
1  #include<stdio.h>
2  #define TAMANO 5
3
4  int main (int argc, char *argv[])
5  {
6      int lista [TAMANO] = {23, 5, 34, 19, 0};
7
8      printf("Lista:\n");
9      for(int i=0; i< TAMANO-1; i++){
10         printf("%i\n", lista[TAMANO-1]);
11     }
12
13     printf("%i\n", lista[TAMANO-1]);
14
15     return 0;
16 }
```

```

Captura de Pantalla 2019-10-23 a la(s) 20.37.19.png
Captura de Pantalla 2019-10-23 a la(s) 21.15.57.png
Nueva carpeta con elementos
Nueva carpeta con elementos 2
Nueva carpeta con elementos 3
Nueva carpeta con elementos 4
Nueva carpeta con elementos 5
Practica 4.pdf
dulces.xlsx
ejemplo1
ejemplo1.c
ejemplo1.dSYM
main.c
practica 2
practica 3
tarea3
uni
uni.c
~$actica 4.docx
[iMac-de-andrea:desktop andreatrujillo$ gcc uni.c -o uni
[iMac-de-andrea:desktop andreatrujillo$ ./uni
Lista:
23,5,34,19,0
iMac-de-andrea:desktop andreatrujillo$

```

ARREGLOS MULTIDIMENSIONALES

```

1  #include<stdio.h>
2  #define DIM 3
3  int main(int argc, char *argv[])
4  {
5      int matriz [DIM][DIM] = {{23, 5, 34},
6                               {8, 46, 22},
7                               {3, 9, 12}};
8      printf("Matriz:\n");
9      for (int i = 0; i < DIM; i++){
10
11          for(int j=0; j<DIM; j++){
12              printf("%i\t", matriz [i][j]);
13          }
14          printf("\n");
15      }
16      return 0;
17  }

```

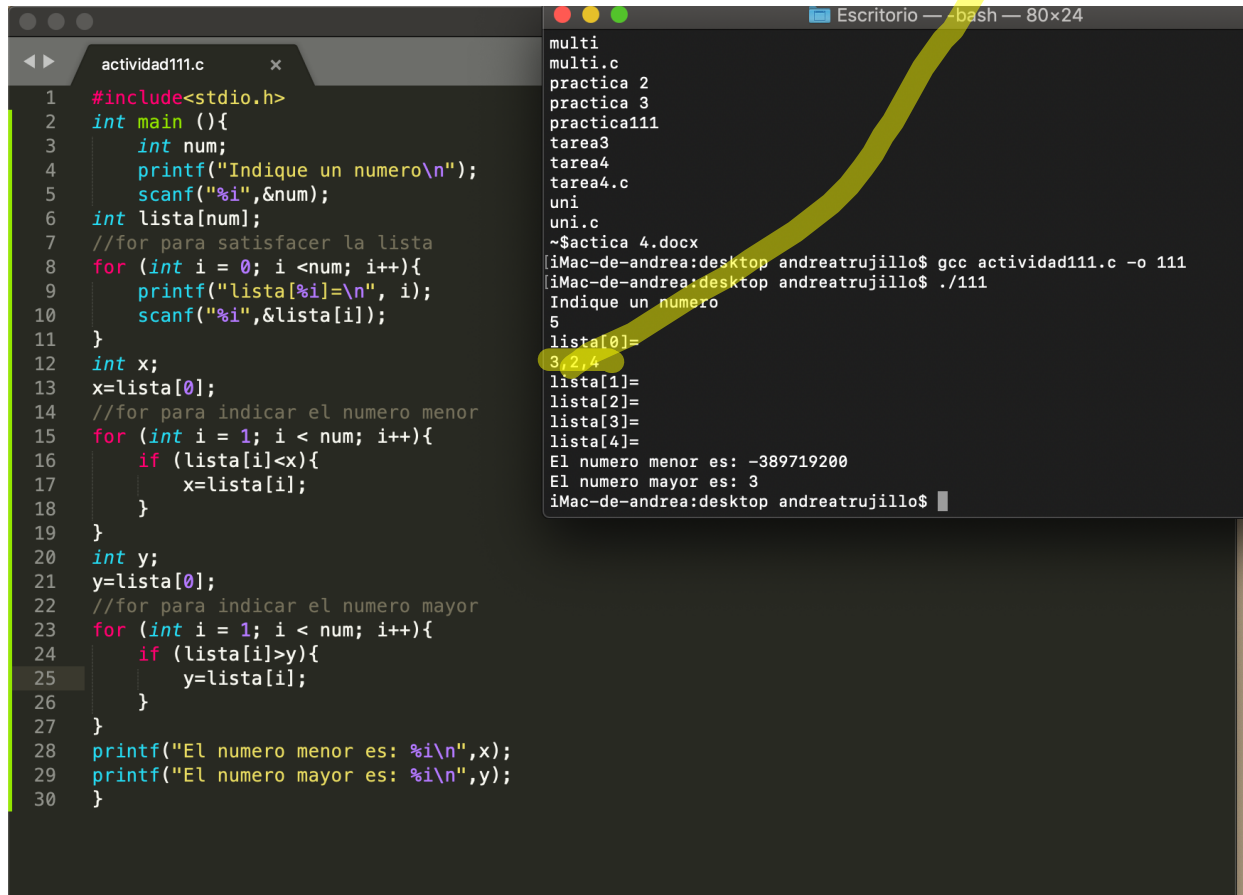
```
Nueva carpeta con elementos 3
Nueva carpeta con elementos 4
Nueva carpeta con elementos 5
Practica 4.pdf
dulces.xlsx
ejemplo1
ejemplo1.c
ejemplo1.dSYM
main.c
multi
multi.c
practica 2
practica 3
tarea3
uni
uni.c
~$actica 4.docx
[iMac-de-andrea:desktop andreatrujillo$ gcc multi.c -o multi
[iMac-de-andrea:desktop andreatrujillo$ ./multi
Matriz:
23      5      34
8        46     22
3        9      12
iMac-de-andrea:desktop andreatrujillo$ █
```

ACTIVIDAD 1

Hacer un programa que:

- Pida al usuario un numero
- Genere un arreglo de esa longitud
- Pida al usuario numeros suficientes para llenar el arreglo
- Muestre al usuario el numero menor y el mayor de dicho arreglo

Esta no es la forma correcta de ingresar datos, por eso no funcionó bien su programa



```
1 #include<stdio.h>
2 int main (){
3     int num;
4     printf("Indique un numero\n");
5     scanf("%i",&num);
6     int lista[num];
7     //for para satisfacer la lista
8     for (int i = 0; i < num; i++){
9         printf("lista[%i]=\n", i);
10        scanf("%i",&lista[i]);
11    }
12    int x;
13    x=lista[0];
14    //for para indicar el numero menor
15    for (int i = 1; i < num; i++){
16        if (lista[i]<x){
17            x=lista[i];
18        }
19    }
20    int y;
21    y=lista[0];
22    //for para indicar el numero mayor
23    for (int i = 1; i < num; i++){
24        if (lista[i]>y){
25            y=lista[i];
26        }
27    }
28    printf("El numero menor es: %i\n",x);
29    printf("El numero mayor es: %i\n",y);
30 }
```

```
multi
multi.c
practica 2
practica 3
practica111
tarea3
tarea4
tarea4.c
uni
uni.c
~$actica 4.docx
iMac-de-andrea:desktop andreatrujillo$ gcc actividad111.c -o 111
iMac-de-andrea:desktop andreatrujillo$ ./111
Indique un numero
5
lista[0]=
3,2,4
lista[1]=
lista[2]=
lista[3]=
lista[4]=
El numero menor es: -389719200
El numero mayor es: 3
iMac-de-andrea:desktop andreatrujillo$
```

ACTIVIDAD 2

Hacer un programa que:

- Pida al usuario un numero N
- Genere dos matrices de N X N
- Pida al usuario numeros suficientes para llenar ambas matrices
- Muestre al usuario la matriz resultado de sumar las dos de entrada

```
actividad112.c
1 #include<stdio.h>
2 int main(int argc, char *argv[]){
3     int n, m, x;
4
5     printf("Indique dos numeros enteros positivos: ");
6     scanf("%d %d", &n, &m);
7
8     int matrizx1[n][m];
9     int matrizx2[n][m];
10    int matrizsuma[n][m];
11
12    printf("Indique %d numeros enteros positivos para crear la primera matriz:\n", n*m);
13
14    for (int j = 0; j < m; j++){
15        for (int i = 0; i < n; i++){
16            scanf("%d", &x);
17            matrizx1[i][j]=x;
18        }
19    }
20    printf("Indique nuevos %d numeros enteros positivos para crear la segunda matriz:\n",n*m);
21
22    for (int j=0; j<m; j++){
23        for(int i=0; i<n; i++){
24            scanf("%d", &x);
25            matrizx2[i][j]=x;
26        }
27    }
28    printf("\nmatriz %dx%d que es efecto de la suma de ambas matrices: \n", n*m);
29
30    for (int j = 0; j < m; j++){
31        for (int i = 0; i < n; i++){
32            matrizsuma[i][j] = matrizx1[i][j] + matrizx2[i][j];
33        }
34    }
35
36    for (int j = 0; j < m; j++){
37        for (int i = 0; i < n; i++){
38            printf("%d\t", matrizsuma[i][j]);
39        }
40        printf("\n");
41    }
42
43    return 0;
44 }
```

```
Escritorio — -bash — 80x24
1 warning generated.
iMac-de-andrea:desktop andreatrujillo$ ./112
Indique dos numeros enteros positivos: 2 3
Indique 6 numeros enteros positivos para crear la primera matriz:
4 5 6 1 2 7
Indique nuevos 6 numeros enteros positivos para crear la segunda matriz:
1 3 7 8 9 4

matriz 6x-326993380 que es efecto de la suma de ambas matrices:
5      8
13     9
11     11
iMac-de-andrea:desktop andreatrujillo$ ./112
Indique dos numeros enteros positivos: 4 2
Indique 8 numeros enteros positivos para crear la primera matriz:
2 3 10 4 6 5 7 12
Indique nuevos 8 numeros enteros positivos para crear la segunda matriz:
1 4 87 4 2 9 0 3

matriz 8x-383649240 que es efecto de la suma de ambas matrices:
3      7      97      8
8      14     7      15
iMac-de-andrea:desktop andreatrujillo$
```

CONCLUSIONES

Ocupamos de manera acertada las indicaciones que se indican en la practica para poder realizar las actividades como anteriormente demostramos, comprobamos que los codigos se efectuaran acertadamente para no tener inconvenientes y de esta manera conocer mas a fondo la importancia de este programa.