

	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 PRACTICA 11

Integrante(s): 1

*No. de Equipo de
cómputo* Equipo 3

No. de Lista o 420054913

Semestre: 2020-1

Fecha de 27 DE OCTUBRE 2019

Observaciones:

CALIFICACIÓN: _____

EN ESTA PRACTICA SE APRENDERÁ EL USO DE ARREGLOS TANTO UNIDIMENSIONALES COMO MATRICES (SIENDO ESTAS DE CUALQUIER TAMAÑO POSIBLE), Y PONIÉNDOLO EN ACCIÓN CUANDO SE REALICEN LOS EJERCICIOS QUE ESTARÁN AUNADOS A ESTO.

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.

Actividad 1

Hacer un program que:

- Pida al usuario un número.
- Genere un arreglo de esa longitud.
- Pida al usuario números suficientes para llenar el arreglo.
- Muestre al usuario el número menor y el mayor de dicho arreglo.

En la siguiente captura se puede ver el código y las condiciones que usa para evaluar cada uno de los datos ingresados; con esto se comprueba cada valor mostrando al final solo el más grande y el más pequeño.

En estas condiciones se prueba los valores de las posiciones dentro del arreglo, comenzando con el valor de la posición hacia adelante, siendo en los 2 casos, que si el valor es mayor en un caso o menor en el otro sustituye al valor que se tenía hasta ese momento y sigue probando con los demás valores hasta terminar de recorrer el arreglo.

Replit - Relevant

RelevantShimmeringUnit

invite

run

share

+ new repl

my repls

talk

AmaranthaH (0)

main.c

saved

1 #include <stdio.h>

2 int main(){

3 int n=0;

4 int nu;

5 int i;

6

7 printf("Tamaño de arreglo\n");

8 scanf("%i",&n);

9 int a[n];

10

11 //llenar arreglo

12 for(i = 0; i<n; i++){

13 | scanf("%i",&nu);

14 | a[i]=nu;

15 }

16

17 //mostrar arreglo

18 | for(i=0; i<n;i++){

19 printf("%i..",a[i]);

20

21 int mi = a[0];

22 int ma= a[0];

23 for(i= 0; i<n;i++){

24 | if(a[i]>ma){

25 | | ma= a[i];

26 | }

27 | }

28 | printf("\n Mayor.. %i \n",ma);

29

30

31 for(i=0; i<n;i++){

32 | if(a[i]<=mi){

33 | | mi=a[i];

34 | }

35 | }

36

37 printf("Menor.. %i",mi);

38 }

https://RelevantShimmeringUnit.amaranthah.repl.run

clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)

clang-7 -pthread -lm -o main main.c

./main

Tamaño de arreglo

6

9

7

8

3

2

11

9..7..8..3..2..11..

Mayor.. 11

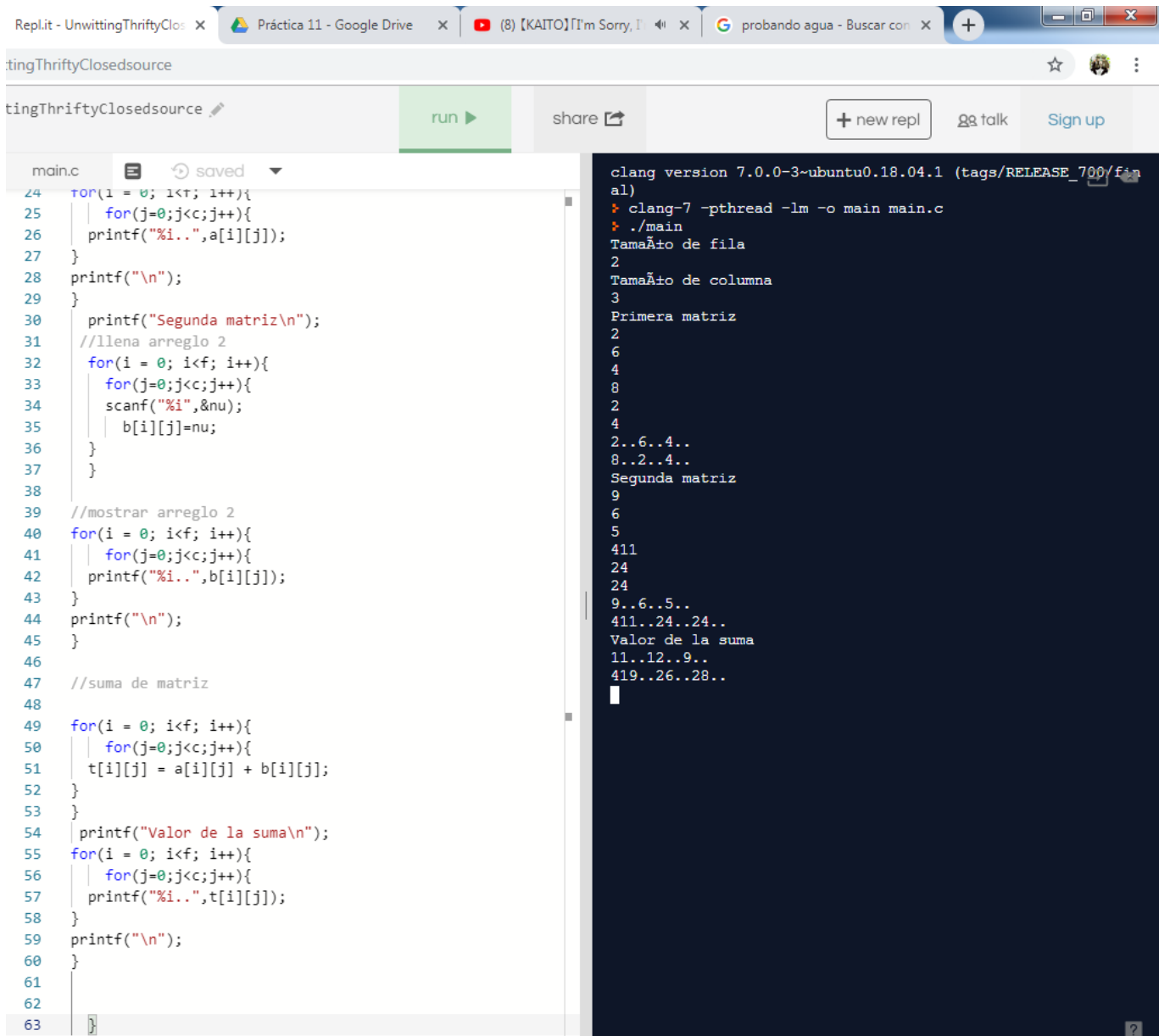
Menor.. 2

Actividad 2

- Pida al usuario un número N Y M
- Genere dos matrices de N*M
- Pida al usuario números suficientes para llenar ambas matrices.
- Muestre al usuario la matriz resultado de sumar las dos de entrada.

Lo que hace el programa es que primero pide el tamaño de las filas y columnas que tendrán las matrices que se harán a continuación, una vez que tenemos la forma de las matrices estas se llenarán con valores y para evitar problemas este mostrara las matrices creadas con los valores.

Una vez que se tengan las 2 matrices llenas, se realizara la suma de estas 2. Se suman los valores de la misma posición para dar un tercer valor nuevo que se guardara en un nuevo arreglo con la posición misma que tenían los valores que se sumaron anteriormente y de esta manera la nueva matriz se llena solo con el resultado de los números sumados.



The screenshot shows a Replit IDE with a C program in a file named `main.c`. The program prompts the user for the number of rows and columns, then fills two matrices with values. It then calculates the sum of the two matrices and displays the result.

```
main.c
24 for(i = 0; i < r; i++){
25     for(j=0; j < c; j++){
26         printf("%i..", a[i][j]);
27     }
28     printf("\n");
29 }
30 printf("Segunda matriz\n");
31 //llena arreglo 2
32 for(i = 0; i < f; i++){
33     for(j=0; j < c; j++){
34         scanf("%i", &nu);
35         b[i][j]=nu;
36     }
37 }
38
39 //mostrar arreglo 2
40 for(i = 0; i < f; i++){
41     for(j=0; j < c; j++){
42         printf("%i..", b[i][j]);
43     }
44     printf("\n");
45 }
46
47 //suma de matriz
48
49 for(i = 0; i < f; i++){
50     for(j=0; j < c; j++){
51         t[i][j] = a[i][j] + b[i][j];
52     }
53 }
54 printf("Valor de la suma\n");
55 for(i = 0; i < f; i++){
56     for(j=0; j < c; j++){
57         printf("%i..", t[i][j]);
58     }
59     printf("\n");
60 }
61
62
63 }
```

The output of the program is as follows:

```
clang version 7.0.0-3-ubuntu0.18.04.1 (tags/RELEASE_700/final)
clang-7 -pthread -lm -o main main.c
./main
Tamaño de fila
2
Tamaño de columna
3
Primera matriz
2
6
4
8
2
4
2..6..4..
8..2..4..
Segunda matriz
9
6
5
411
24
24
9..6..5..
411..24..24..
Valor de la suma
11..12..9..
419..26..28..
```

ESTA PRÁCTICA SIRVIÓ EN COMPRENDER A GRANDES RASGOS EL FUNCIONAMIENTO DE LOS ARREGLOS UNIDIMENSIONALES, Y ENSEÑA QUE PASA CUANDO SE JUNTAN MÁS DE ESTOS CREAN MATRICES QUE FUNCIONAN DE FORMAS MUY VARIADAS. DE ESTA MANERA LOS EJERCICIOS MUESTRAN COMO ESTOS SE VEN, SE USAN Y SE MUEVEN PARA CUMPLIR SUS FUNCIONES.