



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* M.C. JOSE MAURICIO MATAMOROS DE MARIA  
Y CAMPOS

*Asignatura:* ESTRUCTURA DE DATOS Y ALGORITMOS I

*Grupo:* 12

*No de Práctica(s):* 01

*Integrante(s):* ARELLANES CONDE ESTEBAN

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:* 01

*Semestre:* 2022-2

*Fecha de entrega:* 21 feb, 23:59

*Observaciones:*

CALIFICACIÓN: \_\_\_\_\_

# Práctica 1:

## Aplicaciones de arreglos

### 1. Objetivo

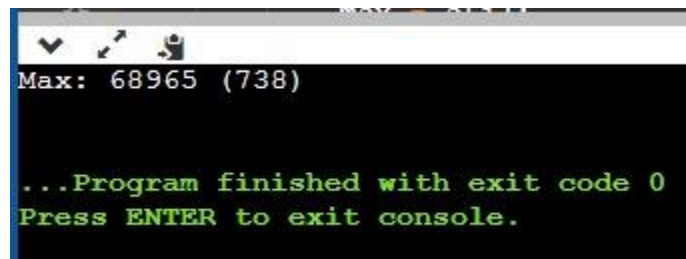
El alumno aprenderá a utilizar arreglos unidimensionales y multidimensionales para dar soluciones a problemas.

### Actividad 1: Máximos y Mínimos

En el programa de la actividad uno consistía en imprimir el número máximo de un arreglo de una sola dimensión en base a números aleatorios generados con la función **srand** con semilla 69, el número máximo de dicho programa es:

**Máximo: 68965 (Posición: 738)**

Y comprobándolo en consola del compilador online:



```
Max: 68965 (738)

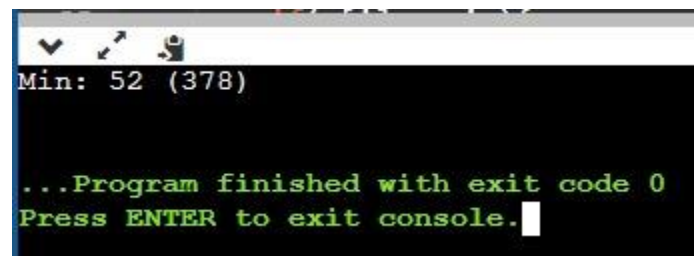
...Program finished with exit code 0
Press ENTER to exit console.
```

Imagen 1. max.c

Y modificando el programa de máximo para imprimir el mínimo, el mínimo es:

**Mínimo: 52 (Posición: 378)**

Y comprobándolo en consola del compilador online:



```
Min: 52 (378)

...Program finished with exit code 0
Press ENTER to exit console.
```

Imagen 2. min.c

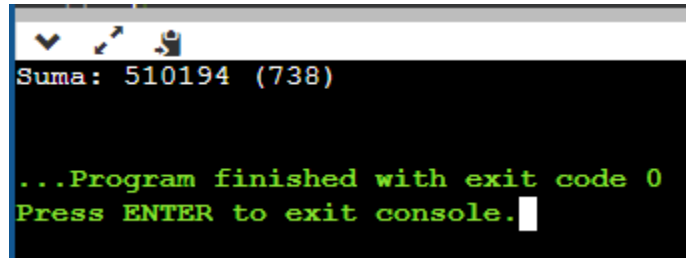
### Actividad 2: Suma

Este programa consistía en base al programa anterior de máximos y mínimos, mostrar e imprimir la suma de todos los elementos generados aleatoriamente con semilla

69 mediante un ciclo **for** y un condicional **if**, la suma es:

**Suma: 510194 (Posición: 738)**

Y comprobándolo en consola del compilador online:



```
Suma: 510194 (738)

...Program finished with exit code 0
Press ENTER to exit console.
```

**Imagen 3. min.c**

### **Actividad 3: Unir arreglos de tamaño fijo**

Para este programa, el problema consistía en intercalar dos arreglos (a y b) de tamaño predefinido por una macro de tamaño 10 y llenados con números aleatorios en base a dos semillas (69 y 80 respectivamente) con la función **srand** e imprimirlos en consola ya intercalados. Esto se puede lograr de varias maneras, pero la realizada en clase era mediante un ciclo **for** de dos variables (i y j) para cada uno de los arreglos y con un tercer arreglo, con la excepción de ser del doble de tamaño que a y b, podemos igualar estos dos a las variables "i" y "j" donde j sería la variable "principal" por decirlo de alguna manera e "i" dentro del ciclo aumentaría dos veces para así intercalar a "a" y a "b" de manera que si comenzáramos con "a" todas sus posiciones serían pares y por ende las de "b" serían impares. Como se muestra a continuación:

```
      i      j
c[0] <- a[0]
c[1] <- b[0]
c[2] <- a[1]
c[3] <- b[1]
```

...

La salida del programa es:

**C = 90 16 30 98 15 49 81 11 5 25 45 17 17 61 93 97 84 79 74 57**

Y comprobándolo en consola del compilador online:

```
c[1] <- b[0] = 16
c[2] <- a[1] = 30
c[3] <- b[1] = 98
c[4] <- a[2] = 15
c[5] <- b[2] = 49
c[6] <- a[3] = 81
c[7] <- b[3] = 11
c[8] <- a[4] = 5
c[9] <- b[4] = 25
c[10] <- a[5] = 45
c[11] <- b[5] = 17
c[12] <- a[6] = 17
c[13] <- b[6] = 61
c[14] <- a[7] = 93
c[15] <- b[7] = 97
c[16] <- a[8] = 84
c[17] <- b[8] = 79
c[18] <- a[9] = 74
c[19] <- b[9] = 57
a: 90 30 15 81 5 45 17 93 84 74
b: 16 98 49 11 25 17 61 97 79 57
c: 90 16 30 98 15 49 81 11 5 25 45 17 17 61 93 97 84 79 74 57

...Program finished with exit code 0
Press ENTER to exit console.
```

Imagen 4. merge.c

## Actividad 4: Suma de matrices

De manera similar a los programas anteriores, el programa realizado durante el laboratorio consistía en realizar una simple y común suma de matrices, cosa que como bien sabemos, una suma de matrices se realiza término por término sumando los homólogos en cada una de las matrices siempre y cuando estas sean del mismo orden. Esto al igual que los programas anteriores, con el uso de arreglos.

```
a:
  90   30   15
  81    5   45
  17   93   84
b:
  74   13   85
  13   65   55
  23   18   80
c:
 164   43  100
  94   70  100
  40  111  164

...Program finished with exit code 0
Press ENTER to exit console.
```

Imagen 5. matrix.c

## Conclusiones:

El uso de arreglos unidimensionales o multidimensionales es parte del conocimiento básico de c y sumamente indispensable en algunos casos. Además, en los programas realizados utilizamos la biblioteca **limits.h** ya que sin esta no podríamos haber hecho uso de la función **srand** o en su defecto **rand** cosa que en lo personal desconocía.

## Referencias y bibliografía:

### Códigos realizados en el laboratorio:

Código de Máximo (max.c) en el compilador online gdb:

<https://onlinegdb.com/dVjqrsSCs>

Código de Mínimo (min.c) en el compilador online gdb:

<https://onlinegdb.com/bosUdgYrC>

Código de Unión de arreglos (merge.c) en el compilador online gdb:

<https://onlinegdb.com/2PLVHbs9b>

Código de Suma de matrices (matrix.c) en el compilador online gdb:

<https://onlinegdb.com/Ye2wXSnFX>

### Fuentes de consulta:

Función srand

cpp reference. (2021). std::srand. Recuperado: 15/02/2022, de cppreference.com

Sitio web: <https://en.cppreference.com/w/cpp/numeric/random/srand>

Biblioteca limits.h

cplusplus. (2000-2021). (limits.h). Recuperado: 15/02/2022, de cplusplus Sitio

web: <https://www.cplusplus.com/reference/climits/>