

Práctica 1:

Aplicaciones de arreglos

Estructuras de Datos y Algoritmos I

Autor: José Mauricio Matamoros de Maria y Campos

1. Objetivo

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

2. Material

Se asume que el alumno cuenta con una computadora con arquitectura Intel x86 o compatible, sistema operativo Linux/Unix, y compilador `gcc` instalado.

3. Instrucciones

1. Realice cada una de las actividades detalladas en la [Sección 4](#) apoyándose en el código de ejemplo del [Apéndice A](#) y responda las preguntas de las mismas.

4. Desarrollo de la práctica

Lea cuidadosamente cada una de las actividades propuestas antes de realizar el programa o las modificaciones indicadas.

4.1. Actividad 1: Máximos y mínimos

El programa del [Apéndice A.2](#) genera un arreglo unidimensional, es decir, un vector, de números aleatorios e imprime el máximo. Compile y ejecute dicho programa y reporte el número más grande del arreglo generado con semilla 69.

Máximo: _____

A continuación, modifique el programa para que localice e imprima el elemento más pequeño del arreglo y reporte dicho número.

[2 puntos] **Mínimo:** _____

4.2. Actividad 2: Suma

Modifique el programa anterior para que calcule la suma de todos los elementos del arreglo y reporte (imprima) dicha suma.

[2 puntos] **Suma:** _____

4.3. Uniendo arreglos de tamaño fijo

El propósito del programa incompleto del [Apéndice A.3](#) es el de intercalar en c los elementos de a y b , en orden, para posteriormente imprimirlos.

Complete el programa, compílelo y ejecútelo, anotando abajo la salida del mismo.

[3 puntos] $C =$ _____

4.4. Suma de matrices

El propósito del programa incompleto del [Apéndice A.4](#) es el desumar dos matrices e imprimir el resultado.

Complete el programa, compílelo y ejecútelo, anotando abajo la salida del mismo.

[3 puntos] $C =$

$$\begin{bmatrix} \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \end{bmatrix}$$

A. Código de ejemplo

A.1. Archivo **src/Makefile**

src/Makefile

```
1 CC      = gcc
2 CFLAGS  = -Wall
3 SILENT   = @
4 PROGRAMS = max merge matrix
5 .PHONY: all clean $(PROGRAMS)
6
7 all: $(PROGRAMS)
8
9 $(PROGRAMS): %.c
10     $(SILENT) $(CC) $(CFLAGS) -o $@ $<
11
12 clean:
13     rm -f *.o
```

A.2. Archivo **src/max.c**

src/max.c

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <limits.h> // INT_MAX
4
5 #define A_SIZE 1024
6
7 int a[A_SIZE];
8 int main(void);
9 void generate(void);
10
11 void generate(void){
12     srand(69);
13     for(int i = 0; i < A_SIZE; ++i)
14         a[i] = 1 + rand() % 69000;
15 }
16
17 int main(void){
18     // Init max to minimum possible value
19     int max = INT_MIN; // limits.h
20     int pmax = 0;
21     // Populate the arrays with random numbers
22     generate();
23     // Find max
24     for(int i = 0; i < A_SIZE; ++i){
25         if(a[i] > max){
26             max = a[i];
27             pmax = i;
28         }
29     }
30
31     printf("Max: %d (%d)\n", max, pmax);
32     return 0;
33 }
```

A.3. Archivo `src/merge.c`

src/merge.c

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <limits.h>
4
5 #define A_SIZE 10
6
7 int a[A_SIZE];
8 int b[A_SIZE];
9 int c[2 * A_SIZE];
10 int main(void);
11 void populate(int* array, int seed);
12 int main(void){
13     // Populate the arrays with random numbers
14     populate(a, 69);
15     populate(b, 80);
16
17     // Merge
18     /*          Student's code here          */
19
20
21     printf("a:");
22     for(int i = 0; i < A_SIZE; ++i) printf(" %d", a[i]);
23     printf("\n");
24
25     printf("b:");
```

A.4. Archivo `src/matrix.c`

src/matrix.c

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <limits.h>
4
5 #define ROWS 3
6 #define COLS 3
7
8 int main(void);
9 void populate(int matrix[ROWS][COLS]);
10 void matrix_print(int matrix[ROWS][COLS]);
11 void matrix_sum(int a[ROWS][COLS], int b[ROWS][COLS], int c[ROWS][COLS]);
12
13 void populate(int matrix[ROWS][COLS]) {
14     for (int row = 0; row < ROWS; ++row) {
15         for (int col = 0; col < COLS; ++col)
16             matrix[row][col] = 1 + rand() % 100;
17     }
18 }
19
20 void matrix_print(int matrix[ROWS][COLS]) {
21     for (int row = 0; row < ROWS; ++row) {
22         for (int col = 0; col < COLS; ++col)
23             printf("%6d", matrix[row][col]);
24         printf("\n");
25     }
26 }
27
28 void matrix_sum(int a[ROWS][COLS], int b[ROWS][COLS], int c[ROWS][COLS]) {
29     for (int i = 0; i < ROWS; ++i) {
30         for (int j = 0; j < COLS; ++j) {
31             c[i][j] = a[i][j] + b[i][j];
32         }
33     }
34 }
35
36 int main(void) {
37     int a[ROWS][COLS];
38     int b[ROWS][COLS];
39     int c[ROWS][COLS];
40
41     // Set the seed
42     srand(69);
43     // Sum
44     matrix_sum(a, b, c);
45
46     // Print
47     // printf("a:\n");
```

B. Reporte Escrito

El reporte de la práctica deberá ser entregada en un archivo en formato PDF siguiendo las siguientes especificaciones:

- La primera página del documento deberá ser la carátula oficial para prácticas de laboratorio disponible en lcp02.fi-b.unam.mx/
- El nombre del documento PDF deberá ser `nn-XXXX-L01.pdf`, donde:
 - `nn` es el número de lista del alumno a dos dígitos forzosos (ej. 01, 02, etc.).
 - `XXXX` corresponden a las dos primeras letras del apellido paterno seguidas de la primera letra del apellido materno y la primera letra del nombre, en mayúsculas y evitando cacofonías; es decir, los cuatro primeros caracteres de su RFC o CURP.
- El reporte consiste en un documento de redacción propia donde el alumno explica de forma concisa y a detalle las actividades realizadas en la práctica, y reportando los resultados obtenidos.
- La longitud del reporte no deberá exceder las 3 páginas, sin contar la carátula.
- El reporte deberá seguir todos los lineamientos para documentos escritos establecidos al inicio del curso.
- Todas las referencias deberán estar debidamente citadas.

IMPORTANTE: No se aceptan archivos en otros formatos ni con nombres distintos a los especificados.