



Introducción a la Programación

Practica 4

Medina Martinez Jonathan Jason
2023640061

25 de marzo del 2023

Índice

1. Objetivo	3
2. Introducción	3
3. Desarrollo	4
3.1. Burbuja	4
3.1.1. Código 1	4
3.1.2. Ejecución	6
3.2. Calculadora de Matrices	7
3.2.1. Código 2	7
3.2.2. Ejecución	9
3.3. Sistema de Ecuaciones	10
3.3.1. Código 3	10
3.3.2. Código 3 Modificado	11
3.3.3. Ejecución	12
4. Conclusión	13

1. Objetivo

Desarrollar programas aplicando el método de la burbuja y resolver problemas con arreglos bidimensionales.

2. Introducción

Esta practica muestra los códigos y la solución de la práctica 3 de Introducción a la Programación, que consiste en el desarrollo de programas utilizando el método de la burbuja y la manipulación de arreglos bidimensionales.

3. Desarrollo

Realice los siguientes programas en C.

3.1. Burbuja

Cree un programa que solicite al usuario un numero de muestras n y genere un arreglo de números pseudoaleatorios dimensión n . Los dígitos deberán estar en el rango de 50 a 300. Una vez generado el arreglo, ordenarlo de mayor a menor utilizando el método de la burbuja.

El programa deberá mostrar ambos arreglos, el generado y ordenado.

3.1.1. Código 1

```
1  /**
2  * @file programa1.c
3  * @author Medina Martinez Jonathan Jason (jmedinam1702@alumno.ipn.mx)
4  * @brief
5  * @version 0.1
6  * @date 2023-03-23
7  *
8  * @copyrigth GPLv3
9  *
10 */
11
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <time.h>
15
16 void generar_arreglo_aleatorio(int arr[], int n);
17 void ordenar_arreglo(int arr[], int n);
18 void imprimir_arreglo(int arr[], int n);
19 void imprimir_texto(char text[]);
20
21 int main()
22 {
23     int n = 0;
24
25     char text1[] = "Ingrese el valor de n";
26     imprimir_texto(text1);
27     scanf("%d",&n);
28
29     int arr[n];
30     generar_arreglo_aleatorio(arr, n);
31
32     char text2[] = "Arreglo ordenado de mayor a menor:";
33     imprimir_texto(text2);
34     imprimir_arreglo(arr, n);
35
36     ordenar_arreglo(arr, n);
37     char text3[] = "Arreglo generado:";
38     imprimir_texto(text3);
39
40     imprimir_arreglo(arr, n);
41
42     return 0;
43 }
44
```

```

45 /// @brief Genera un arreglo aleatorio de una dimension dado
46 /// @param arr El arreglo generada
47 /// @param n La dimension del arreglo
48 void generar_arreglo_aleatorio(int arr[], int n) {
49     srand(time(NULL));
50     for (int i = 0; i < n; i++)
51     {
52         arr[i] = (rand() % 500) + 1;
53     }
54 }
55
56 /// @brief Ordena un arreglo de mayor a menor
57 /// @param arr El arreglo a ordenar
58 /// @param n La dimension del arreglo
59 void ordenar_arreglo(int arr[], int n) {
60     int aux = 0, i = 0, j = 0;
61     for (i = 0; i < n; i++)
62     {
63         for (j = 0; j < n - 1 - i; j++)
64         {
65             if (arr[j] > arr[j + 1])
66             {
67                 aux = arr[j];
68                 arr[j] = arr[j + 1];
69                 arr[j + 1] = aux;
70             }
71         }
72     }
73 }
74
75 /// @brief Imprime un arreglo
76 /// @param arr El arreglo a imprimir
77 /// @param n La dimension del arreglo
78 void imprimir_arreglo(int arr[], int n) {
79     for (int i = 0; i < n; i++) {
80         printf("%d ", arr[i]);
81     }
82     printf("\n\n");
83 }
84
85 /// @brief Imprime un texto
86 /// @param text el texto a imprimir
87 void imprimir_texto(char text[]) {
88     printf("\n\n%s\n\n", text);
89 }

```

3.1.2. Ejecución

Ingrese el valor de n

20

Arreglo ordenado de mayor a menor:

236 331 104 323 11 14 457 468 371 103 458 97 455 380 332 37 232 213 238 36

Arreglo generado:

11 14 36 37 97 103 104 213 232 236 238 323 331 332 371 380 455 457 458 468

3.2. Calculadora de Matrices

Cree un programa que

1. Defina una matriz A de tamaño 3×3 .
2. Defina una matriz B de tamaño 3×3 .
3. Solicite al usuario los elementos de la matriz A y B .
4. Calcule la suma de A mas B y muestre en pantalla el resultado.
5. Calcule la resta de A menos B y muestre en pantalla el resultado.
6. Calcule la multiplicación de A por B y muestre en pantalla el resultado.

3.2.1. Código 2

```
1  /**
2  * @file programa2.c
3  * @author Medina Martinez Jonathan Jason (jmedinam1702@alumno.ipn.mx)
4  * @brief
5  * @version 0.1
6  * @date 2023-03-24
7  *
8  * @copyright GPLv3
9  *
10 */
11
12 #include <stdio.h>
13
14 void pedir_elementos(int matriz[3][3]);
15 void suma_matrices(int matrizA[3][3], int matrizB[3][3], int
    matrizC[3][3]);
16 void resta_matrices(int matrizA[3][3], int matrizB[3][3], int
    matrizC[3][3]);
17 void multiplicacion_matrices(int matrizA[3][3], int matrizB[3][3], int
    matrizC[3][3]);
18 void imprimir_matriz(int matrizC[3][3]);
19 void imprimir_texto(char text[]);
20
21 int main() {
22     int A[3][3], B[3][3], C[3][3];
23
24     char text1[] = "Ingrese los elementos de la matriz A:";
25     imprimir_texto(text1);
26     pedir_elementos(A);
27
28     char text2[] = "Ingrese los elementos de la matriz B:";
29     imprimir_texto(text2);
30     pedir_elementos(B);
31
32     char text3[] = "La suma de A + B es:";
33     imprimir_texto(text3);
34     suma_matrices(A, B, C);
35     imprimir_matriz(C);
36
37     char text4[] = "La resta de A - B es:";
38     imprimir_texto(text4);
```

```

39     resta_matrices(A, B, C);
40     imprimir_matriz(C);
41
42     char text5[] = "La multiplicacion de A x B es:";
43     imprimir_texto(text5);
44     multiplicacion_matrices(A, B, C);
45     imprimir_matriz(C);
46
47     return 0;
48 }
49
50 /// @brief Pide los elementos de la matriz
51 /// @param matriz La matriz en la que se guardan los elementos
52 void pedir_elementos(int matriz[3][3]) {
53     for (int i = 0; i < 3; i++) {
54         for (int j = 0; j < 3; j++) {
55             scanf("%d", &matriz[i][j]);
56         }
57     }
58 }
59
60 /// @brief Suma 2 matrices
61 /// @param matrizA Una de las matrices a sumar
62 /// @param matrizB Una de las matrices a sumar
63 /// @param matrizC La matriz resultante de la suma
64 void suma_matrices(int matrizA[3][3], int matrizB[3][3], int
matrizC[3][3]) {
65     for (int i = 0; i < 3; i++) {
66         for (int j = 0; j < 3; j++) {
67             matrizC[i][j] = matrizA[i][j] + matrizB[i][j];
68         }
69     }
70 }
71
72 /// @brief Resta 2 matrices
73 /// @param matrizA Una de las matrices a restar
74 /// @param matrizB Una de las matrices a restar
75 /// @param matrizC La matriz resultante de la resta
76 void resta_matrices(int matrizA[3][3], int matrizB[3][3], int
matrizC[3][3]) {
77     for (int i = 0; i < 3; i++) {
78         for (int j = 0; j < 3; j++) {
79             matrizC[i][j] = matrizA[i][j] - matrizB[i][j];
80         }
81     }
82 }
83
84 /// @brief Multiplica 2 matrices
85 /// @param matrizA Una de las matrices a multiplicar
86 /// @param matrizB Una de las matrices a multiplicar
87 /// @param matrizC La matriz resultante de la multiplicacion
88 void multiplicacion_matrices(int matrizA[3][3], int matrizB[3][3], int
matrizC[3][3]) {
89     for (int i = 0; i < 3; i++) {
90         for (int j = 0; j < 3; j++) {
91             matrizC[i][j] = 0;
92             for (int k = 0; k < 3; k++) {
93                 matrizC[i][j] += matrizA[i][k] * matrizB[k][j];
94             }
95         }
96     }
97 }

```



```

95     }
96 }
97 }
98
99 /// @brief Imprime una matriz
100 /// @param matriz C
101 void imprimir_matriz(int matrizC[3][3]) {
102     for (int i = 0; i < 3; i++) {
103         for (int j = 0; j < 3; j++) {
104             printf("%4d ", matrizC[i][j]);
105         }
106         printf("\n");
107     }
108     printf("\n");
109 }
110
111 /// @brief Imprime un texto
112 /// @param text el texto a imprimir
113 void imprimir_texto(char text[]) {
114     printf("\n\n%s\n\n", text);
115 }

```

3.2.2. Ejecución

<p>Ingrese los elementos de la matriz A:</p> <p>56 25 86 45 62 62 58 25 2</p> <p>Ingrese los elementos de la matriz B:</p> <p>152 123 228 2 4 5 2 2 6</p>	<p>La suma de A + B es:</p> <table> <tr><td>208</td><td>148</td><td>314</td></tr> <tr><td>47</td><td>66</td><td>67</td></tr> <tr><td>60</td><td>27</td><td>8</td></tr> </table> <p>La resta de A - B es:</p> <table> <tr><td>-96</td><td>-98</td><td>-142</td></tr> <tr><td>43</td><td>58</td><td>57</td></tr> <tr><td>56</td><td>23</td><td>-4</td></tr> </table> <p>La multiplicacion de A x B es:</p> <table> <tr><td>8734</td><td>7160</td><td>13409</td></tr> <tr><td>7088</td><td>5907</td><td>10942</td></tr> <tr><td>8870</td><td>7238</td><td>13361</td></tr> </table>	208	148	314	47	66	67	60	27	8	-96	-98	-142	43	58	57	56	23	-4	8734	7160	13409	7088	5907	10942	8870	7238	13361
208	148	314																										
47	66	67																										
60	27	8																										
-96	-98	-142																										
43	58	57																										
56	23	-4																										
8734	7160	13409																										
7088	5907	10942																										
8870	7238	13361																										

3.3. Sistema de Ecuaciones

1. Tomando como base la definición anterior, cree un programa que permita resolver el siguiente sistema de ecuaciones:

$$2x - 4y - 3z = 15$$

$$x + 5y - 5z = 5$$

$$4x + 2y + 67z = 20$$

3.3.1. Código 3

```
1  /**
2  * @file programa3.c
3  * @author Medina Martinez Jonathan Jason
4  * (jmedinam1702@alumno.ipn.mx)
5  * @brief
6  * @version 0.1
7  * @date 2023-03-24
8  *
9  * @copyrigth GPLv3
10 */
11
12 #include <stdio.h>
13
14 int main() {
15
16     float A[3][3] = {{2, -4, -3}, {1, 5, -5}, {4, 2, 67}};
17     float B[3] = {15, 5, 20};
18
19     float invA[3][3];
20     float detA = A[0][0] * (A[1][1] * A[2][2] - A[2][1] * A[1][2])
21 - A[0][1] * (A[1][0] * A[2][2] - A[2][0] * A[1][2])
22 + A[0][2] * (A[1][0] * A[2][1] - A[2][0] * A[1][1]);
23
24     invA[0][0] = (A[1][1] * A[2][2] - A[2][1] * A[1][2]) / detA;
25     invA[0][1] = (A[0][2] * A[2][1] - A[2][2] * A[0][1]) / detA;
26     invA[0][2] = (A[0][1] * A[1][2] - A[1][1] * A[0][2]) / detA;
27
28     invA[1][0] = (A[1][2] * A[2][0] - A[2][2] * A[1][0]) / detA;
29     invA[1][1] = (A[0][0] * A[2][2] - A[2][0] * A[0][2]) / detA;
30     invA[1][2] = (A[0][2] * A[1][0] - A[1][2] * A[0][0]) / detA;
31
32     invA[2][0] = (A[1][0] * A[2][1] - A[2][0] * A[1][1]) / detA;
33     invA[2][1] = (A[0][1] * A[2][0] - A[2][1] * A[0][0]) / detA;
34     invA[2][2] = (A[0][0] * A[1][1] - A[1][0] * A[0][1]) / detA;
35
36     float X[3];
37     X[0] = invA[0][0] * B[0] + invA[0][1] *
38     B[1] + invA[0][2] * B[2];
39     X[1] = invA[1][0] * B[0] + invA[1][1] *
40     B[1] + invA[1][2] * B[2];
41     X[2] = invA[2][0] * B[0] + invA[2][1] *
42     B[1] + invA[2][2] * B[2];
43
44     printf("La solucion es: \n");
45     printf("x = %f \n", X[0]);
```

```

46     printf("y = %f \n", X[1]);
47     printf("z = %f \n", X[2]);
48
49     return 0;
50 }

```

2. Compruebe los resultados obtenidos haciendo el producto punto de cada fila de A con el vector columna X.

```

La solucion es:
x = 6.579670
y = -0.398352
z = -0.082418

```

3. Modifique su programa para que permita resolver cualquier sistema de tres ecuaciones.

3.3.2. Código 3 Modificado

```

1  /**
2  * @file programa3.c
3  * @author Medina Martinez Jonathan Jason
4  * (jmedinam1702@alumno.ipn.mx)
5  * @brief
6  * @version 0.1
7  * @date 2023-03-24
8  *
9  * @copyrigth GPLv3
10 *
11 */
12 #include <stdio.h>
13
14 int main() {
15     float A[3][3];
16     float B[3];
17
18     printf("Calculadora de sistemas de ecuaciones lineales
19           3x3\n\n");
20     printf("Ejemplo de sistema de ecuaciones lineales 3x3:\n");
21     printf("A[0][0]x + A[0][1]y + A[0][2]z = B[1]\n");
22     printf("A[1][0]x + A[1][1]y + A[1][2]z = B[2]\n");
23     printf("A[2][0]x + A[2][1]y + A[2][2]z = B[3]\n\n");
24
25     printf("Ingrese los elementos de la matriz A: \n");
26     for(int i=0; i<3; i++) {
27         for(int j=0; j<3; j++) {
28             printf("A[%d][%d] = ", i, j);
29             scanf("%f", &A[i][j]);
30         }
31     }
32
33     printf("Ingrese los elementos del vector B: \n");
34     for(int i=0; i<3; i++) {
35         printf("B[%d] = ", i);
36         scanf("%f", &B[i]);
37     }

```

```

37
38 float invA[3][3];
39 float detA = A[0][0] * (A[1][1] * A[2][2] - A[2][1] * A[1][2])
40 - A[0][1] * (A[1][0] * A[2][2] - A[2][0] * A[1][2])
41 + A[0][2] * (A[1][0] * A[2][1] - A[2][0] * A[1][1]);
42
43 invA[0][0] = (A[1][1] * A[2][2] - A[2][1] * A[1][2]) / detA;
44 invA[0][1] = (A[0][2] * A[2][1] - A[2][2] * A[0][1]) / detA;
45 invA[0][2] = (A[0][1] * A[1][2] - A[1][1] * A[0][2]) / detA;
46
47 invA[1][0] = (A[1][2] * A[2][0] - A[2][2] * A[1][0]) / detA;
48 invA[1][1] = (A[0][0] * A[2][2] - A[2][0] * A[0][2]) / detA;
49 invA[1][2] = (A[0][2] * A[1][0] - A[1][2] * A[0][0]) / detA;
50
51 invA[2][0] = (A[1][0] * A[2][1] - A[2][0] * A[1][1]) / detA;
52 invA[2][1] = (A[0][1] * A[2][0] - A[2][1] * A[0][0]) / detA;
53 invA[2][2] = (A[0][0] * A[1][1] - A[1][0] * A[0][1]) / detA;
54
55 float X[3];
56 X[0] = invA[0][0] * B[0] + invA[0][1] * B[1] + invA[0][2] *
57     B[2];
58 X[1] = invA[1][0] * B[0] + invA[1][1] * B[1] + invA[1][2] *
59     B[2];
60 X[2] = invA[2][0] * B[0] + invA[2][1] * B[1] + invA[2][2] *
61     B[2];
62
63 printf("La solucion es: \n");
64 printf("x = %f \n", X[0]);
65 printf("y = %f \n", X[1]);
66 printf("z = %f \n", X[2]);
67
68 return 0;
69 }

```

3.3.3. Ejecución

```

Calculadora de sistemas de ecuaciones lineales 3x3

Ejemplo de sistema de ecuaciones lineales 3x3:
A[0][0]x + A[0][1]y + A[0][2]z = B[1]
A[1][0]x + A[1][1]y + A[1][2]z = B[2]
A[2][0]x + A[2][1]y + A[2][2]z = B[3]

Ingrese los elementos de la matriz A:
A[0][0] = 10
A[0][1] = 2
A[0][2] = 58
A[1][0] = 46
A[1][1] = 1
A[1][2] = 2
A[2][0] = 7
A[2][1] = 69
A[2][2] = 4
Ingrese los elementos del vector B:
B[0] = 12
B[1] = 26
B[2] = 3
La solucion es:
x = 0.560828
y = -0.019845
z = 0.110886

```

4. Conclusión

La práctica 3 de Introducción a la Programación ha permitido desarrollar habilidades en la manipulación de arreglos y el uso del método de la burbuja para ordenarlos. Los programas creados en C han sido efectivos en la generación y ordenamiento de los arreglos, y demuestran la aplicación de los conceptos teóricos aprendidos en clase.