



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Vazquez Guzman Jorge Antonio

Matrícula: 372504

Maestro: Pedro Núñez Yépiz

Actividad No. : 8

Tema - Unidad : Teoria Arreglos y Funciones

Ensenada Baja California a 1 de octubre del 2023



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ejercicio 1:

```
82 // Funcion que cuando es llamada le pide al usuario llenar el vector de manera manual con un rango inferior y superior
83 int vectorman(int vector1[], int size)
84 {
85     int i, val, sal;
86
87     for (i = 0; i < size; i++)
88     {
89         sal = 1;
90         do
91         {
92             printf("Ingrese un valor entre el 30 y el 70: ");
93             scanf("%d", &val);
94             if (val >= 30 && val <= 70) // Valida que este dentro del rango
95             {
96                 vector1[i] = val;
97                 sal = 1;
98             }
99             else
100             {
101                 printf("Ingrese un valor valido\n");
102                 system("pause");
103                 sal = 0;
104             }
105         } while (sal != 1);
106     }
107 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
3. Llenado de vector con los primeros 2.
4. Imprimir vectores.
5. Llenado de matriz 4x4.
6. Imprimir matriz 4x4.
0. Salir.
1
Ingrese un valor entre el 30 y el 70: 56
Ingrese un valor entre el 30 y el 70: 54
Ingrese un valor entre el 30 y el 70: 32
Ingrese un valor entre el 30 y el 70: 1
Ingrese un valor valido
Presione una tecla para continuar . . .
```

Ejercicio 2:



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
113 int vectorrand(int vector2[], int size)
114 {
115     int i, n, j, sal;
116
117     for (i = 0; i < size; i++)
118     {
119         sal = 1;
120         do
121         {
122             n = rand() % 20 + 1;
123             for (j = 0; j < i; j++) // Ciclo que se encarga que no se repita
124             {
125                 if (vector2[j] == n)
126                 {
127                     sal = 0;
128                 }
129             }
130             if (sal == 1)
131             {
132                 vector2[i] = n;
133             }
134         } while (sal != 1);
135     }
136     printf("\nProceso terminado con exito...");
137     system("pause");
138     return 0;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

MENU

1. Llenado de vector manualmente.
2. Llenado de vector aleatorio.
3. Llenado de vector con los primeros 2.
4. Imprimir vectores.
5. Llenado de matriz 4x4.
6. Imprimir matriz 4x4.
0. Salir.

2



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ejercicio 3:

```
140
141 // Funcion que cuando es llamada llena un vector con los 2 anteriores
142 int vector1y2(int vector1[], int vector2[], int vector3[], int size)
143 {
144     int i, j;
145
146     for (i = 0; i < size; i++) // Llena con vector 1
147     {
148         vector3[i] = vector1[i];
149     }
150     for (j = 0; j < size; j++) // Llena con vector 2
151     {
152         vector3[j + 10] = vector2[j];
153     }
154     printf("\nProceso terminado con exito...");
155     system("pause");
156     return 0;
157 }
158
159 // Funcion que cuando es llamada imprime los 3 vectores
160 int imprimir(int vector[], int size)
161 {
162     int i;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

MENU

1. Llenado de vector manualmente.
 2. Llenado de vector aleatorio.
 3. Llenado de vector con los primeros 2.
 4. Imprimir vectores.
 5. Llenado de matriz 4x4.
 6. Imprimir matriz 4x4.
 0. Salir.
- 3

Proceso terminado con exito...Presione una tecla para continuar . . .



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ejercicio 4:

```
159 // Funcion que cuando es llamada imprime los 3 vectores
160 int imprimir(int vector[], int size)
161 {
162     int i;
163
164     printf("\n Vector: \n");
165     for (i = 0; i < size; i++)
166     {
167         printf("%d\n", vector[i]);
168     }
169     printf("\nProceso terminado con exito...");
170     system("pause");
171     return 0;
172 }
173
174 // Funcion que cuando es llamada llena la matriz 4x4 con los vectores 1 y 2
175 int matriz4(int matriz[][4], int vector1[], int vector2[], int size)
176 {
177     int i, j, sup1 = 0, sup2 = 0;
178
179     for (i = 0; i < size; i++)
180     { // Ciclos para recorrer la matriz
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Vector:

56
56
54
54
58
59
57
53
52
51

Proceso terminado con exito...Presione una tecla para continuar . . .



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ejercicio 5:

```
174 // Funcion que cuando es llamada llena la matriz 4x4 con los vectores 1 y 2
175 int matriz4(int matriz[][4], int vector1[], int vector2[], int size)
176 {
177     int i, j, sup1 = 0, sup2 = 0;
178
179     for (i = 0; i < size; i++)
180     { // Ciclos para recorrer la matriz
181         for (j = 0; j < size; j++)
182         {
183             if (sup1 < 10)
184             { // Esto verifica que no se sobrepase de los 10 elementos del vector 1
185                 matriz[i][j] = vector1[sup1++];
186             }
187             else
188             {
189                 if (sup2 < 10)
190                 { // Si ya paso los 10 elementos, pasa al vector 2
191                     matriz[i][j] = vector2[sup2++];
192                 }
193             }
194         }
195     }
196     printf("\nProceso terminado con exito...");
197     system("pause");
198     return 0;
199 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

MENU

1. Llenado de vector manualmente.
 2. Llenado de vector aleatorio.
 3. Llenado de vector con los primeros 2.
 4. Imprimir vectores.
 5. Llenado de matriz 4x4.
 6. Imprimir matriz 4x4.
 0. Salir.
- 5

Proceso terminado con exito...Presione una tecla para continuar . . .



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ejercicio 6:

```
201 // Funcion que cuando es llamada imprime la matriz 4x4
202 int imprimirmat(int matriz[][4], int size)
203 {
204     int i, j;
205
206     printf("Elementos de la matriz:\n");
207     for (i = 0; i < size; i++) // Estos ciclos recorren la matriz
208     {
209         for (j = 0; j < size; j++)
210         {
211             printf("%d \t", matriz[i][j]);
212         }
213         printf("\n");
214     }
215     printf("\nProceso terminado con exito...");
216     system("pause");
217
218     return 0;
219 }
220
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
3. Llenado de vector con los primeros 2.
4. Imprimir vectores.
5. Llenado de matriz 4x4.
6. Imprimir matriz 4x4.
0. Salir.
```

6

Elementos de la matriz:

56	56	54	54
58	59	57	53
52	51	1120474088	397
-1970492949	32760	1696195180	0

Proceso terminado con exito...Presione una tecla para continuar . . .



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Salir:

```
MENU
1. Llenado de vector manualmente.
2. Llenado de vector aleatorio.
3. Llenado de vector con los primeros 2.
4. Imprimir vectores.
5. Llenado de matriz 4x4.
6. Imprimir matriz 4x4.
0. Salir.
```

```
0
```

```
Repetiste el menu un total de 6 veces. Fue un placer...
```

```
D:\C:\Users\user\OneDrive\Documents\Github\Programación Fundamentos\Artículo 10\ejercicios
```