



# **Universidad Autónoma de Baja California**

## **Facultad de Ingeniería, Arquitectura y Diseño**



### **PROGRAMACIÓN ESTRUCTURADA**

#### **ARREGLOS - ANEXO**

#### **PRÁCTICA 8**

### **Ingeniero en Software y Tecnologías Emergentes**

**Brayan Ivan Perez Ventura**

**372781**

## CÓDIGO

```
52 void menu()
53 {
54     int vector[10], vector_2[10], vector_3[20], matrix_1[4][4];
55     int op, n, m, p;
56     n = 10;
57     m = 10;
58     p = 20;
59     do
60     {
61         op = msge_menu();
62         switch (op)
63         {
64             case 1:
65                 vector1(vector, n);
66                 break;
67             case 2:
68                 vector2(vector_2, m);
69                 break;
70             case 3:
71                 vector3(vector_3, vector_2, m, vector, n);
72                 break;
73             case 4:
74                 printVectors(vector, n, vector_2, m, vector_3, p);
75                 break;
76             case 5:
77                 matrix(matrix_1, vector_2, vector, n);
78                 break;
79             case 6:
80                 printMatrix(matrix_1);
81                 break;
82         }
83     } while (op != 0);
84 }
85 // ***** MESSAGE MENU *****
86 int msge_menu()
87 {
88     int op;
89     system("CLS");
90     printf("M E N U\n");
91     printf("1.- Llenar vector 1 (Manualmente)\n");
92     printf("2.- Llenar vector 2 aleatoriamente\n");
93     printf("3.- Llenar Vector 3 (Con Vector 1 y Vector 2)\n");
94     printf("4.- Imprimir vectores\n");
95     printf("5.- Llenar matriz 4x4\n");
96     printf("6.- Imprimir matriz\n");
97     printf("0.- Salir\n");
98     op = valid("Selecciona una opcion: ", 0, 6);
99     return op;
100 }
```

## SALIDA

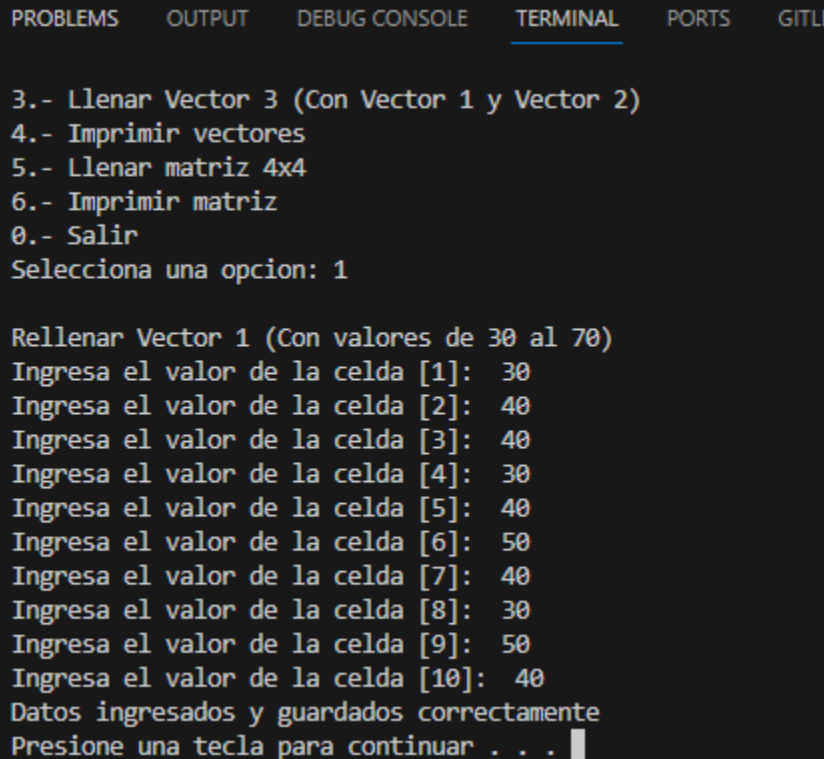
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

M E N U
1.- Llenar vector 1 (Manualmente)
2.- Llenar vector 2 aleatoriamente
3.- Llenar Vector 3 (Con Vector 1 y Vector 2)
4.- Imprimir vectores
5.- Llenar matriz 4x4
6.- Imprimir matriz
0.- Salir
Selecciona una opcion: 1
```

## CÓDIGO

```
131 void vector1(int vector[], int n)
132 {
133     int i;
134     printf("\nRellenar Vector 1 (Con valores de 30 al 70) \n");
135     for (i = 0; i < n; i++)
136     {
137         printf("Ingresa el valor de la celda [%d]: ", i + 1);
138         vector[i] = valid(" ", 30, 70);
139     }
140     printf("Datos ingresados y guardados correctamente\n");
141     system("PAUSE");
142 }
```

## SALIDA



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITL

3.- Llenar Vector 3 (Con Vector 1 y Vector 2)
4.- Imprimir vectores
5.- Llenar matriz 4x4
6.- Imprimir matriz
0.- Salir
Selecciona una opcion: 1

Rellenar Vector 1 (Con valores de 30 al 70)
Ingresa el valor de la celda [1]: 30
Ingresa el valor de la celda [2]: 40
Ingresa el valor de la celda [3]: 40
Ingresa el valor de la celda [4]: 30
Ingresa el valor de la celda [5]: 40
Ingresa el valor de la celda [6]: 50
Ingresa el valor de la celda [7]: 40
Ingresa el valor de la celda [8]: 30
Ingresa el valor de la celda [9]: 50
Ingresa el valor de la celda [10]: 40
Datos ingresados y guardados correctamente
Presione una tecla para continuar . . .
```

## CÓDIGO

```
145 void vector2(int vector_2[], int m)
146 {
147     int i, num;
148     srand(time(NULL));
149     for (i = 0; i < m; i++)
150     {
151         while (existElem(vector_2, m, num = ((rand() % 20) + 1)))
152             ;
153         vector_2[i] = num;
154     }
155     printf("\nValores aleatorios generados correctamente en el vector.\n");
156     system("PAUSE");
157 }
```

## SALIDA

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   GITLENS   COMMENTS

M E N U

- 1.- Llenar vector 1 (Manualmente)
- 2.- Llenar vector 2 aleatoriamente
- 3.- Llenar Vector 3 (Con Vector 1 y Vector 2)
- 4.- Imprimir vectores
- 5.- Llenar matriz 4x4
- 6.- Imprimir matriz
- 0.- Salir

Selecciona una opcion: 2

Valores aleatorios generados correctamente en el vector.

Presione una tecla para continuar . . . █

## CÓDIGO

```
160 void vector3(int vector3[], int vector_2[], int m, int vector[], int n)
161 {
162     int i;
163     for (i = 0; i < n; i++)
164     {
165         vector3[i] = vector[i];
166         vector3[i + m] = vector_2[i];
167     }
168     printf("Datos guardados en el vector 3 correctamente\n");
169     system("PAUSE");
170 }
```

## SALIDA

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  COMMENTS

M E N U
1.- Llenar vector 1 (Manualmente)
2.- Llenar vector 2 aleatoriamente
3.- Llenar Vector 3 (Con Vector 1 y Vector 2)
4.- Imprimir vectores
5.- Llenar matriz 4x4
6.- Imprimir matriz
0.- Salir
Selecciona una opcion: 3
Datos guardados en el vector 3 correctamente
Presione una tecla para continuar . . .
```

## CÓDIGO

```
173 void printVectors(int vector[], int n, int vector_2[], int m, int vector_3[], int p)
174 {
175     int i;
176     printf("VECTOR 1\n");
177     for (i = 0; i < n; i++)
178     {
179         printf("Vector [%d]: [%2d]\n", i, vector[i]);
180     }
181     printf("VECTOR 2\n");
182     for (i = 0; i < m; i++)
183     {
184         printf("Vector [%d]: [%2d]\n", i, vector_2[i]);
185     }
186     printf("VECTOR 3\n");
187     for (i = 0; i < p; i++)
188     {
189         printf("Vector [%d]: [%2d]\n", i, vector_3[i]);
190     }
191     system("PAUSE");
192 }
```

## SALIDA

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

M E N U
1.- Llenar vector 1 (Manualmente)
2.- Llenar vector 2 aleatoriamente
M E N U
1.- Llenar vector 1 (Manualmente)
2.- Llenar vector 2 aleatoriamente
3.- Llenar Vector 3 (Con Vector 1 y Vector 2)
4.- Imprimir vectores
5.- Llenar matriz 4x4
6.- Imprimir matriz
0.- Salir
Selecciona una opcion: 4
VECTOR 1
Vector [0]: [30]
Vector [1]: [40]
Vector [2]: [40]
Vector [3]: [30]
Vector [4]: [40]
Vector [5]: [50]
Vector [6]: [40]
Vector [7]: [30]
Vector [8]: [50]
Vector [9]: [40]
VECTOR 2
Vector [0]: [18]
Vector [1]: [11]
Vector [2]: [ 3]
Vector [3]: [12]
Vector [4]: [15]
Vector [5]: [ 4]
Vector [6]: [ 1]
Vector [7]: [10]
Vector [8]: [ 2]
Vector [9]: [17]
VECTOR 3
Vector [0]: [30]
Vector [1]: [40]
Vector [2]: [40]
Vector [3]: [30]
Vector [4]: [40]
Vector [5]: [50]
Vector [6]: [40]
Vector [7]: [30]
Vector [8]: [50]
Vector [9]: [40]
Vector [10]: [18]
Vector [11]: [11]
Vector [12]: [ 3]
Vector [13]: [12]
Vector [14]: [15]
Vector [15]: [ 4]
Vector [16]: [ 1]
Vector [17]: [10]
Vector [18]: [ 2]
Vector [19]: [17]
Presione una tecla para continuar . . .
```

## CÓDIGO

```
195 void matrix(int matrix_1[][4], int vector_2[], int vector[], int n)
196 {
197     int i, j, k = 0;
198     for (i = 0; i < 4; i++)
199     {
200         for (j = 0; j < 4; j++)
201         {
202             if (k < n)
203             {
204                 matrix_1[i][j] = vector[k];
205             }
206             else
207             {
208                 matrix_1[i][j] = vector_2[k - n];
209             }
210             k++;
211         }
212     }
213     printf("Matriz llenada correctamente\n");
214     system("PAUSE");
215 }
```

## SALIDA

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   GITLENS   COMMENTS

M E N U

- 1.- Llenar vector 1 (Manualmente)
- 2.- Llenar vector 2 aleatoriamente
- 3.- Llenar Vector 3 (Con Vector 1 y Vector 2)
- 4.- Imprimir vectores
- 5.- Llenar matriz 4x4
- 6.- Imprimir matriz
- 0.- Salir

Selecciona una opcion: 5

Matriz llenada correctamente

Presione una tecla para continuar . . .

## CÓDIGO

```
218 void printMatrix(int matrix_1[][4])
219 {
220     printf("Matrix 4x4:\n");
221     for (int i = 0; i < 4; i++)
222     {
223         for (int j = 0; j < 4; j++)
224         {
225             printf("%d\t", matrix_1[i][j]);
226         }
227         printf("\n");
228     }
229     system("PAUSE");
230 }
```

## SALIDA

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   GITLENS   COMMENTS

M E N U

- 1.- Llenar vector 1 (Manualmente)
- 2.- Llenar vector 2 aleatoriamente
- 3.- Llenar Vector 3 (Con Vector 1 y Vector 2)
- 4.- Imprimir vectores
- 5.- Llenar matriz 4x4
- 6.- Imprimir matriz
- 0.- Salir

Selecciona una opcion: 6

Matrix 4x4:

30	40	40	30
40	50	40	30
50	40	18	11
3	12	15	4

Presione una tecla para continuar . . .