



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en computación

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Kevin Alejandro Gonzalez Torres

Matrícula: 372354

Maestro: Pedro Núñez Yépiz

Actividad Numero: 8

Tema - Unidad: Vectores y matrices

Ensenada Baja California a 1 de octubre del 2022



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1. INTRODUCCIÓN

En este reporte se mostrará el uso de vectores y matrices, de sus distintos usos en diversas aplicaciones diarias y de cómo funcionan.

2. COMPETENCIA

Se practicará y se aprenderá el uso de los vectores y las matrices.

3. FUNDAMENTOS

Vectores:

<https://info64.ro/en/Vectors/>

https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C/Vectores

Matrices:

<https://beginnersbook.com/2014/01/2d-arrays-in-c-example/>

<https://www.programiz.com/c-programming/c-multi-dimensional-arrays>

4. PROCEDIMIENTO

Realiza programa en C el programa deberá tener el siguiente menú.

MENÚ

- 1.- LLENAR VECTOR 1 (MANUALMENTE)
- 2.- LLENAR VECTOR 2 ALEATORIAMENTE
- 3.- LLENAR VECTOR 3 (CON VECTOR1 Y VECTOR2)
- 4.- IMPRIMIR VECTORES
- 5.- LLENA MATRIZ 4 X 4
- 6.- IMPRIMIR MATRIZ
- 0.- SALIR

NOTA: EL PROGRAMA DEBERÁ REPETIRSE CUANTAS VECES LO DESEE EL USUARIO

NOTA 2: EL VECTOR 1 DE 10 POSICIONES, NÚMEROS DEL 30 AL 70

NOTA 3: EL VECTOR 2 DE 10 POSICIONES CON NÚMEROS GENERADOS ALEATORIAMENTE DEL 1 AL 20 (SIN REPETIR)

NOTA 4: EL VECTOR 3 DE 20 POSICIONES, CON LOS DATOS DEL ARREGLO1 Y ARREGLO2

NOTA 5: MATRIZ 4 X 4 LLENARLA CON LOS DATOS DEL VECTOR1 Y VECTOR2,



5. RESULTADOS Y CONCLUSIONES

1.-

```
83 void fillVector1(int vector[], int size)
84 {
85     printf("Fill it only with numbers between 30 to 70\n");
86
87     for (int i = 0; i < size; i++)
88     {
89         printf("Vector[%i]: ", i);
90         scanf("%i", &vector[i]);
91
92         if (vector[i] < 30 || vector[i] > 70)
93         {
94             printf("Value out of range. Please enter another number.\n");
95             i--; // repeat last (incorrect) index
96         }
97     }
98 }
```

2.-

```
100 void fillVector2(int vector[], int size)
101 {
102     printf("Vector filled with random numbers between 1 to 20");
103
104     int checkRepeated;
105
106     srand(time(NULL));
107     for (int i = 0; i < size; i++)
108     {
109         // function to avoid repeated number
110         do
111         {
112             checkRepeated = 0; // initialize to assume the number is not repeated
113             vector[i] = rand() % 20 + 1;
114
115             // check if the random number already exists in the vector
116             for (int j = 0; j < i; j++)
117             {
118                 if (vector[j] == vector[i])
119                 {
120                     checkRepeated = 1; // set to 1 if the number is repeated
121                     break;
122                 }
123             }
124         } while (checkRepeated); // repeat until a unique number is generated
125     }
126 }
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

3.-

```
128 void fillVector3(int vector[], int vector1[], int vector2[], int size)
129 {
130     for (int i = 0; i < size; i++)
131     {
132         vector[i] = vector1[i];
133         vector[i + 10] = vector2[i];
134     }
135 }
```

4.-

```
137 void fillMatrix(int matrix[][5], int size)
138 {
139     for (int r = 0; r < size; r++)
140     {
141         for (int c = 0; c < size; c++)
142         {
143             printf("Matrix[%i, %i]: ", r, c);
144             scanf("%i", &matrix[r][c]);
145         }
146     }
147 }
```

5.-

```
149 void printVector(int vector[], int size)
150 {
151     printf("Vector elements: {");
152     for (int i = 0; i < size; i++)
153     {
154         printf("%d", vector[i]);
155         if (i < size - 1)
156         {
157             printf(", ");
158         }
159     }
160     printf("}");
161 }
```

6.-



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
163 void printMatrix(int matrix[][5], int size)
164 {
165     int r, c;
166
167     printf("Matrix elements: \n{\n");
168     for (r = 0; r < size; r++)
169     {
170         for (c = 0; c < size; c++)
171         {
172             printf("%d\t", matrix[r][c]);
173         }
174         printf("\n");
175     }
176     printf("}");
177 }
```

6. ANEXOS

GTKA_RP08_PE_ANEXOS



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

7. REFERENCIAS

Diseño de algoritmos y su codificación en lenguaje C

Corona, M.A. y Ancona, M.A. (2011)..

España: McGraw-Hill.

ISBN: 9786071505712

Programación estructurada a fondo: implementación de algoritmos en C

:Pearson Educación. Sznajdleder, P. A. (2017)..

Buenos Aires, Argentina: Alfaomega

Como programar en C/C++

H.M. Deitel/ P.J. Deitel

Segunda edición

Editorial: Prentice Hall.

ISBN: 9688804711

Programación en C. Metodología, estructura de datos y objetos

Joyanes, L. y Zahonero, I. (2001)..

España: McGraw-Hill.

ISBN: 8448130138