

Vector manual

```
ACT8.cpp 1 X
G+ ACT8.cpp > ...
107
108 // llenado del vector manualmente
109 void vect1(int vector1[])
110 {
111     int num;
112     int i;
113     for (i = 0; i < 10; i++)
114     {
115         printf(" Posicion [%d] del vector\n", i);
116         num = validacion("Ingresa un numero entre 30 y 70: ", 30, 70);
117         vector1[i] = num;
118         system("CLS");
119     }
120 }
121
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Vector 1
Vector[0] 34
Vector[1] 32
Vector[2] 35
Vector[3] 34
Vector[4] 35
Vector[5] 36
Vector[6] 65
Vector[7] 56
Vector[8] 44
Vector[9] 66
Presione una tecla para continuar . . .
```

Vector aleatorio

```
ACT8.cpp > vect2(int [])
120
121
122 //Funcion que verifica si el numero ya se encuentra en el vector
123 int num_repetido(int num, int i, int vector2[])
124 {
125     for (int j = 0; j < i; j++)...
126     return 0; // si el numero no esta en el vector retorna 0
127 }
128
129 //llenar vector de manera aleatoria sin repetir numeros
130 void vect2(int vector2[])
131 {
132     int num;
133     int i;
134     for (i = 0; i < 10; i++)
135     {
136         do
137         {
138             num = rand() % 20 + 1;
139         } while (num_repetido(num, i, vector2));
140         vector2[i] = num;
141     }
142     printf("El vector se ha llenado con exito\n");
143 }
144
```

Vector 2

```
Vector[0] 1
Vector[1] 9
Vector[2] 19
Vector[3] 17
Vector[4] 4
Vector[5] 13
Vector[6] 6
Vector[7] 8
Vector[8] 10
Vector[9] 14
Presione una tecla para continuar . . .
```

Vector usando vector manual y vector aleatorio

```
ACT8.cpp > vect2(int [])
152 void vect1_vect2(int vector3[], int vect2[], int vect1[])
153 {
154     for (int i = 0; i < 20; i++)
155     {
156         if (i < 10) // Se llenan las primeras 10 posiciones con el vect
157         {
158             vector3[i] = vect1[i];
159         }
160         else // se llenan las ultimas 10 posiciones con el vector 2
161         {
162             vector3[i] = vect2[i - 10];
163         }
164     }
165     printf("Vector de 20 espacios completo exitosamente!\n");
166 }
167
168 // Funcion para imprimir los vectores
169 > void imprimir(int vect1[], int vect2[], int vect3[])...
192
```

Vector 3
Vector[0] 1
Vector[1] 0
Vector[2] 0
Vector[3] 0
Vector[4] 1117133904
Vector[5] 32759
Vector[6] 16
Vector[7] 0
Vector[8] 0
Vector[9] 0
Vector[10] 1117132080
Vector[11] 32759
Vector[12] 8
Vector[13] 0
Vector[14] 0
Vector[15] 0
Vector[16] 268501009
Vector[17] 0
Vector[18] -377485472
Vector[19] 437
Presione una tecla para continuar . . .

Matriz 4x4

```
ACT8.cpp > vect2(int [])
151 //llenado de vector con datos de otros vectores
152 void vect1_vect2(int vector3[], int vect2[], int vect1[])
153 {
154     for (int i = 0; i < 20; i++)
155     {
156         if (i < 10) // Se llenan las primeras 10 posiciones con el vect
157         {
158             vector3[i] = vect1[i];
159         }
160         else // se llenan las ultimas 10 posiciones con el vector 2
161         {
162             vector3[i] = vect2[i - 10];
163         }
164     }
165     printf("Vector de 20 espacios completo exitosamente!\n");
166 }
167
168 // Funcion para imprimir los vectores
169 > void imprimir(int vect1[], int vect2[], int vect3[])...
192
193 //Funcion para llenar una matriz con 2 vectores
194 void matriz_4x4(int matriz[][4], int vect1[], int vect2[])
195 {
196     int i, j, k = 0;
197     for (i = 0; i < 4; i++)
198     {
199         for (j = 0; j < 4; j++)
200         {
201             if (k < 10) // Se llenan los primeros 10 espacios con el ve
202             {
203                 matriz[i][j] = vect1[k];
204             }
205             else // Se llenan los ultimos 6 espacios con el vector 2
206             {
207                 matriz[i][j] = vect2[k - 6];
208             }
209             k++;
210         }
211     }
212     printf("La matriz 4x4 se lleno exitosamene!\n");
213 }
214
```

Matriz 4x4
1 0 0 0
-897442736 32759 16 0
0 0 0 0
268501009 0 897058656 747
Presione una tecla para continuar . . .