



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. : 9

Tema - Unidad : Funciones y metodos de ordenacion y busqueda

Ensenada Baja California a 8 de octubre del 2023



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Anexo

Menu que llama en su mayoría a la librería:

```
//Funcion que se encarga de leer la seleccion del usuario
void menu()
{
    int op, vect[15], mat[4][4];

    do
    {
        printf("\t MENU\n");
        printf("1. Llenar vector \n");
        printf("2. Llenar matriz \n");
        printf("3. Imprimir vector \n");
        printf("4. Imprimir matriz \n");
        printf("5. Ordenar vector \n");
        printf("6. Buscar valor en vector \n");
        printf("0. Salir\n");
        op = validar("Ingrese la opcion que desee: ", 0, 6); //Llama a una funcion de la libreria e impide que se rompa el programa

        switch (op)
        { //Todas a excepcio de la 6 son funcion de la libreria que se mandan a llamar
            case 1:
                fillnorep(vect, 15, 100, 200);
                break;
            case 2:
                fillmatnorep(mat, 4, 4, 1, 16);
                break;
            case 3:
                printvect(vect, 15);
                break;
            case 4:
                printmat(mat, 4, 4);
                break;
            case 5:
                ordenar(vect, 15);
                break;
            case 6:
                buscar(vect);
                break;
            case 0:
                printf("\nHasta luego...");
                break;
            default:
                printf("\nOpcion no valida");
                break;
        }
    } while (op != 0);
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ejercicio 1:

```
117
118 // Esta funcion rellena un vector con numeros aleatorios pero sin repeticion, para esto llama a la funcion busq_seq pa
119 void fillnorep(int vect[], int n, int ri, int rf)
120 {
121     int i, num, rango;
122     rango = rf - ri + 1;
123
124     for (i = 0; i < n; i++)
125     {
126         do
127         {
128             num = (rand() % rango) + ri;
129         } while (busq_seq(vect, i, num) != -1); //Llama a la otra funcion para confirmar que no se encuentra
130
131         vect[i] = num;
132     }
133 }
134
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Presti\OneDrive\Documentos\GitHub\ProgramacionEstructurada> cd 'c:\Users\Presti\OneDrive\Documentos\GitHub\Progra
PS C:\Users\Presti\OneDrive\Documentos\GitHub\ProgramacionEstructurada\Actividad9\output> & .\JAVG_Act9_932.exe'
MENU
1. Llenar vector
2. Llenar matriz
3. Imprimir vector
4. Imprimir matriz
5. Ordenar vector
6. Buscar valor en vector
0. Salir
Ingrese la opcion que desee: 1
MENU
1. Llenar vector
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ejercicio 2:

```
135 // Esta funcion rellena una matriz con numeros aleatorios sin repeticion
136 void fillmatnorep(int mat[][4], int m, int n, int ri, int rf)
137 {
138     int i, j, cont, largo;
139     largo = m * n;
140
141     int vect[largo];
142     fillnorep(vect, largo, ri, rf); //llena el vector para despues llenar la matriz
143
144     for (i = 0, cont = 0; i < 4; i++)
145     {
146         for (j = 0; j < 4; j++, cont++)
147         {
148             mat[i][j] = vect[cont];
149         }
150     }
151 }
152
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
6. Buscar valor en vector
0. Salir
Ingrese la opcion que desee: 1
MENU
1. Llenar vector
2. Llenar matriz
3. Imprimir vector
4. Imprimir matriz
5. Ordenar vector
6. Buscar valor en vector
0. Salir
Ingrese la opcion que desee: 2
MENU
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ejercicio 3:

```
257 void printvect(int vect[], int n)
258 {
259     int i;
260
261     for (i = 0; i < n; i++)
262     {
263         printf("%d \n", vect[i]);
264     }
265 }
266
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
6. Buscar valor en vector
0. Salir
Ingrese la opcion que desee: 3
141
185
172
138
180
169
165
168
196
122
```

Ejercicio 4:

```
242 // Funcion que imprime la matriz
243 void printmat(int mat[][4], int m, int n)
244 {
245     int i, j;
246
247     for (i = 0; i < m; i++)
248     {
249         for (j = 0; j < n; j++)
250         {
251             printf("%2d", mat[i][j]);
252         }
253         printf("\n");
254     }
255 }
256
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
2. Llenar matriz
3. Imprimir vector
4. Imprimir matriz
5. Ordenar vector
6. Buscar valor en vector
0. Salir
Ingrese la opcion que desee: 4
[12][ 4][ 7][13]
[ 8][15][10][ 5]
[14][ 3][ 9][16]
[11][ 6][ 1][ 2]
MENU
1. Llenar vector
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ejercicio 5:

```
44 // Funcion que ordena el vector de menor a mayor
45 void ordenar(int vect[], int n)
46 {
47     int i, j, aux;
48     for (i = 0; i < n - 1; i++)
49     {
50         for (j = i + 1; j < n; j++)
51         {
52             if (vect[j] <= vect[i])
53             {
54                 aux = vect[i]; //Metodo de burbuja
55                 vect[i] = vect[j];
56                 vect[j] = aux;
57             }
58         }
59     }
60 }
61
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
138
141
149
151
161
163
165
167
168
169
172
180
185
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ejercicio 6

```
85 // Funcion que busca si hay n elemento en el un vector
86 int busq_seq(int vect[], int n, int num)
87 {
88     int i;
89     for (i = 0; i < n; i++)
90     {
91         if (vect[i] == num)
92         {
93             return i; // Regresa esto si encontro un numero igual
94         }
95     }
96     return -1; // regresa esto si es que no hay ninguno igual
97 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

MENU

1. Llenar vector
2. Llenar matriz
3. Imprimir vector
4. Imprimir matriz
5. Ordenar vector
6. Buscar valor en vector
0. Salir

Ingrese la opcion que desee: 6

Ingrese un numero que desee buscar entre 100 y 200: 196

El numero 196 se encuentra en la posicion 14 MENU