

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Ingeniería en Software y Tecnologías Emergentes
Programación Estructurada

Actividad 9.

ALUMNO: Fernando Haro Calvo

MATRICULA: 372106

GRUPO: 932

PROFESOR: Pedro Núñez Yépiz

8 de octubre del 2023

Funciones Auxiliares:

```
21 // Valida la entrada del usuario en un rango de numeros enteros.
22 int valiNum(int ri, int rf)
23 {
24     char cadena[50];
25     int num;
26
27     do
28     {
29         fflush(stdin);
30         gets(cadena);
31         num = atoi(cadena);
32     } while (num < ri || num > rf);
33
34     return num;
35 }
```

```
37 // Genera un numero aleatorio en un rango de numeros enteros.
38 int numAleatorio(int ri, int rf)
39 {
40     int rango = (rf - ri + 1);
41
42     return rand() % rango + ri;
43 }
```

Ejercicio 1:

```
45 // Llena el vector con valores aleatorio dentro de un rango de numeros sin repetir.
46 void llenarVectSinRep(int vect[], int n, int ri, int rf)
47 {
48     int i, num, rango;
49     rango = rf - ri + 1;
50
51     for (i = 0; i < n; i++)
52     {
53         do
54         {
55             num = (rand() % rango) + ri;
56         } while (busqSeqVect(vect, i, num) != -1);
57
58         vect[i] = num;
59     }
60 }
```

Salida:

Presione una tecla para continuar . . .

Ejercicio 2:

```
62 // Llena una matriz con numeros aleatorios sin repetir.
63 void llenarMatSinRep(int mat[][4], int m, int n, int ri, int rf)
64 {
65     int i, j, cont, num, largo;
66     largo = m * n;
67
68     int vect[largo];
69     llenarVectSinRep(vect, largo, ri, rf);
70
71     for (i = 0, cont = 0; i < 4; i++)
72     {
73         for (j = 0; j < 4; j++, cont++)
74         {
75             mat[i][j] = vect[cont];
76         }
77     }
78 }
```

Salida:

Presione una tecla para continuar . . .

Ejercicio 3:

```
115 // Imprime un vector.
116 void imprVect(int vect[], int n)
117 {
118     int i;
119
120     for (i = 0; i < n; i++)
121     {
122         printf("[%3d]\n", vect[i]);
123     }
124 }
```

Salida:

```
[101]
[147]
[142]
[160]
[143]
[181]
[183]
[174]
[109]
[114]
```

Presione una tecla para continuar . . . █

Ejercicio 4:

```
126 // Imprime una matriz.
127 void imprMat(int mat[][4], int m, int n)
128 {
129     int i, j;
130
131     for (i = 0; i < m; i++)
132     {
133         for (j = 0; j < n; j++)
134         {
135             printf("[%2d]", mat[i][j]);
136         }
137         printf("\n");
138     }
139 }
```

Salida:

```
[ 6][16][ 2][14]
[ 7][15][13][ 5]
[ 9][ 4][11][ 3]
[ 1][12][ 8][10]
```

Presione una tecla para continuar . . .

Ejercicio 5:

```
141 // Ordena el vector usando una mezcla de bubble e insertion sort.
142 void ordVect(int vect[], int n)
143 {
144     int i, j;
145     int temp;
146
147     for (i = 0; i < n - 1; i++)
148     {
149         for (j = i + 1; j < n; j++)
150         {
151             if (vect[j] <= vect[i])
152             {
153                 temp = vect[i];
154                 vect[i] = vect[j];
155                 vect[j] = temp;
156             }
157         }
158     }
159 }
```

Salida:

```
Presione una tecla para continuar . . . █
```

Imprimir vector:

```
[135]
[142]
[143]
[147]
[160]
[166]
[169]
[173]
[174]
[181]
[183]
```

```
Presione una tecla para continuar . . . █
```

Ejercicio 6:

```
86 void buscarVect(int vect[])
87 {
88     int num, indice;
89     printf("Ingrese un numero entre 100-200 para buscar en el vector: ");
90     num = valiNum(100, 200);
91     indice = busqSeqVect(vect, 15, num);
92
93     if (indice != -1)
94     {
95         printf("El numero %d se encuentra en el vector en el indice %d", num, indice);
96     }
97     else
98     {
99         printf("El numero %d no se encuentra en el vector", num);
100     }
101 }
```

```
80 // Busca un número en un arreglo de forma secuencial.
81 int busqSeqVect(int vect[], int n, int num)
82 {
83     int i;
84
85     for (i = 0; i < n; i++)
86     {
87         if (vect[i] == num)
88         {
89             return i; // Retorna índice
90         }
91     }
92
93     return -1; // No se encontró
94 }
```

Salida:

```
Ingrese un numero entre 100-200 para buscar en el vector: 181
El numero 181 se encuentra en el vector en el indice 13
Presione una tecla para continuar . . . █
```



```
Ingrese un numero entre 100-200 para buscar en el vector: 120  
El numero 120 no se encuentra en el vector  
Presione una tecla para continuar . . . █
```