



Tema 5: Abstracción Procedimental

Nombre: Carlos Antonio Alvarez Angulo

Matrícula: 00366182

Grupo: Grupo 432

Fecha: 10/10/2023

ACTIVIDAD 9

Realiza programa en C utilizando librería propia, el programa deberá tener el siguiente menú.

MENÚ

- 1.- LLENAR VECTOR
- 2.- LLENAR MATRIZ
- 3.- IMPRIMIR VECTOR
- 4.- IMPRIMIR MATRIZ
- 5.- ORDENAR VECTOR
- 6.- BUSCAR VALOR EN VECTOR
- 0.- SALIR

NOTA: El programa deberá repetirse cuantas veces lo desee el usuario, Validado el menú con la función vali_num

INSTRUCCIONES

- 1.- **LLENAR VECTOR .-** Llenar vector con 15 números, los números generados aleatoriamente, los números entre el rango de 100 al 200 (**no repetidos**)
- 2.- **LLENAR MATRIZ .-** Llenar la matriz de 4x4 con con números generados aleatoriamente, números entre el rango de 1 al 16 (**no repetidos**)
- 3.- **IMPRIMIR VECTOR .-** Imprime el vector que se envíe, donde la función recibe como parámetro el vector,tamaño, nombre del vector.
- 4.- **IMPRIMIR MATRIZ.-** Imprime la matriz sin importar el tamaño de la matriz recibiendo como parámetros la matriz, la cantidad de renglones y columnas, así como nombre que se le dará a la matriz
- 5.- **ORDENAR VECTOR.-** Usar función que ordene el vector por el método de ordenación de la Burbuja mejorada.
- 6.- **BUSCAR VALOR EN VECTOR.-** Buscar un valor en el vector usando el método de búsqueda secuencial.
- 0.- SALIR



Universidad Autónoma de Baja California Programación Estructurada

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4  #include<random>
5  #include "C:\Users\Carlos\Desktop\Anthony\Cursos\Programacion Estructurada\CAAA_PE_ACT10\anthony_h\anthony.h"
6  int msg();
7  void menu();
8  void add_vect_rand(int vect1[], int m, int ri, int rf);
9  void imp_vect(int vect1[], int m);
10 void add_mat(int mat[4][4], int m, int n, int ri, int rf);
11 void imp_mat(int mat[4][4], int m, int n);
12 void buscar(int vect1[], int m);
13 int main()
14 {
15     menu();
16     return 0;
17 }
18 int msg()
19 {
20     int op;
21     system("CLS");
22     printf("Menu\n");
23     printf("1.- Llenar un vector aleatoriamente\n");
24     printf("2.- Llenar una matriz 4x4 aleatoriamente\n");
25     printf("3.- Imprimir un vector\n");
26     printf("4.- Imprimir una matriz 4x4\n");
27     printf("5.- Ordenar vector\n");
28     printf("6.- Buscar un valor en el vector\n");
29     printf("0.- Salir\n");
30     printf("Escoje una opcion: ");
31     scanf("%d",&op);
32     return op;
33 }
34 void menu()
35 {
36     int op, vect1[15],mat[4][4], m, n, ri, rf;
37     do
38     {
39         op=msg();
40         switch (op)
41         {
42             case 1:
43                 add_vect_rand(vect1, 15, 100, 200);
44                 break;
45             case 2:
46                 add_mat(mat, 4, 4, 1, 16);
47                 break;
48             case 3:
```



```
49         imp_vect(vect1, 15);
50         break;
51     case 4:
52         imp_mat(mat, 4, 4);
53         break;
54     case 5:
55         ordenar(vect1, 15);
56         break;
57     case 6:
58         buscar(vect1, 15);
59         break;
60     }
61 }
62 while (op!=0);
63 }
64 void add_vect_rand(int vect1[], int m, int ri, int rf)
65 {
66     int i, j, rango, num, band;
67     system("CLS");
68     ri=100;
69     rf=200;
70     m=15;
71     rango=(rf-ri)+1;
72     for(i=0;i<m;i++)
73     {
74         do
75         {
76             num=(rand()%rango)+ri;
77             j=0;
78             band=false;
79             while (j<i && band==false)
80             {
81                 if (num==vect1[j])
82                 {
83                     band=true;
84                 }
85                 j++;
86             }
87         }
88         while (band==true);
89         vect1[i]=num;
90     }
91     system("PAUSE");
92 }
93 void imp_vect(int vect1[], int m)
94 {
95     int i;
```



```
196     system("CLS");
197     printf("Vector 1: \n");
198     m=10;
199     for(i=0;i<m;i++)
200     {
201         printf("POS [%d]: %d, ",i,vect1[i]);
202     }
203     system("PAUSE");
204 }
205 void add_mat(int mat[4][4], int m, int n, int ri, int rf)
206 {
207     int i, j, k, l, rango, num, band;
208     system("CLS");
209     ri=1;
210     rf=16;
211     m=4;
212     n=4;
213     rango=(rf-ri)+1;
214     for (i=0;i<m;i++)
215     {
216         for (j=0;j<n;j++)
217         {
218             do
219             {
220                 num=(rand()%(rf-ri+1))+ri;
221                 band=0;
222                 for (k=0;k<i;k++)
223                 {
224                     for (l=0;l<j;l++)
225                     {
226                         if (num==mat[k][l])
227                         {
228                             band=1;
229                             break;
230                         }
231                     }
232                 }
233             } while (band);
234             mat[i][j] = num;
235         }
236     }
237     system("PAUSE");
238 }
239 void imp_mat(int mat[4][4], int m, int n)
240 {
```



```
141     int i, j;
142     system("CLS");
143     m=4;
144     n=4;
145     for(i=0;i<m;i++)
146     {
147         for(j=0;j<n;j++)
148         {
149             printf("%d ",mat[i][j]);
150         }
151         printf("\n");
152     }
153     system("PAUSE");
154 }
155 void buscar(int vect1[], int m)
156 {
157     int i, num, x;
158     system("CLS");
159     num=vali_num(100, 200, "Inserte el numero que busca: ");
160     ordenar(vect1, m);
161     x=-1;
162     for (i=0;i<m;i++)
163     {
164         if (vect1[i] == num)
165         {
166             x = i;
167             break;
168         }
169     }
170     if (x != -1)
171     {
172         printf("Si existe\n");
173         printf("%d esta en el indice: %d\n", num, x);
174     }
175     else
176     {
177         printf("No existe\n");
178     }
179     system("PAUSE");
180 }
```



Universidad Autónoma de Baja California Programación Estructurada

Menu

- 1.- Llenar un vector aleatoriamente
- 2.- Llenar una matriz 4x4 aleatoriamente
- 3.- Imprimir un vector
- 4.- Imprimir una matriz 4x4
- 5.- Ordenar vector
- 6.- Buscar un valor en el vector
- 0.- Salir

Escoje una opcion: █

Vector 1:

POS [0]: 141, POS [1]: 185, POS [2]: 172, POS [3]: 138, POS [4]: 180, POS [5]: 169, POS [6]: 165, POS [7]: 168, POS [8]: 196, POS [9]: 122,

```
12 4 7 12
13 8 13 15
10 5 15 14
13 7 15 3
```

Inserte el numero que busca: 180

Si existe

180 esta en el indice: 12