Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave **36276**

Alumno: Antonio Ramos González

Matrícula: 372576

Maestro: Pedro Núñez Yépiz

Actividad No. 8 : Arreglos en c (anexos)

Tema – Unidad 5 : arreglos y cadenas

Ensenada Baja California a 1 de octubre del 2023

```c
// Antonio Ramos Gonzalez Mt: 372576
// 9/27/2023 || 10/01/2023
// Funciones e Introduccion a Arreglos en C
// RGA_Act8_932
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>

int msg();
void menu();
int validar(const char msg[], int li_in, int li_sup);
void vect_1(int vect1[], int m);
void vect_2(int vect2[], int lim_in, int lim_sup);
void vect_3(int vect1[], int vect2[], int vect3[], int m);
void imprimir_vetores(int vect1[], int vect2[], int vect3[], int m, int n);
void matriz(int vect1[], int vect2[], int mtz[][4], int m, int v);
void imprimir_matriz(int mtz[][4],int m);

int main()
{
    srand(time(NULL));//genera semilla para numeros aleatorios
    menu();
    return 0;
}

int msg()
{
    short int opc;//guardara la decision del usuario
    system("CLS");
    printf("MENU\n");
    printf("1.- LLENA VECTOR 1(MANUALMENTE)\n");
    printf("2.- LLENA VECTOR 2(ALEATORIAMENTE)\n");
    printf("3.- LLENA VECTOR 3(VECTOR 1 Y VECTOR 2)\n");
    printf("4.- IMPRIMIR VECTOES\n");
    printf("5.- LLENA MATRIZ(VECTOR 1 Y VECTOR 2)\n");
    printf("6.- IMPRIMIR MATRIZ\n");
    printf("0.- SALIR\n");
    printf("ESCOGE UNA OPCION: ");
    scanf("%hd", &opc);//lee la decicion del usuario
    return opc;//retorna opc
}
```

```
43
44  void menu()
45  {
46      int vect1[10], vect2[10], vect3[20];//definimos el tamaño de los vetores
47      int mtz[4][4];//definimos el tamaño de la matriz
48      short int opc;
49      do
50      {
51          opc = msg();//leemos el valor retornado de la funcion msg
52          switch (opc)
53          {
54          case 1:
55              vect_1(vect1, 10);//se manda vect1, asi como su tamaño a la funcion
56
57              break;
58          case 2:
59              vect_2(vect2, 1, 20);//se manda vect2, asi como su limite de datos a la funcion
60              break;
61          case 3:
62              vect_3(vect1, vect2, vect3, 20);//se manda vect1,vect2 para llenar vect3
63              break;
64          case 4:
65              imprimir_vetores(vect1, vect2, vect3, 10, 20);//se manda vect1,vect2 y vect3 para imprimirlos
66              break;
67          case 5:
68              matriz(vect1, vect2, mtz, 4, 10);//se manda vect1,vect2 para llenar mtz
69              break;
70          case 6:
71              imprimir_matriz(mtz,4);//se manda mtz para imprimirla
72              break;
73          default:
74              break;
75          }
76
77      } while (opc != 0);
78  }
79
```

```c

//valida los datos que entraran en la funcion vect_1
int validar(const char msg[], int li_in, int li_sup)
{
    char cad[3];//cadena que leera un numero
    int num;//guardara un numer

    do
    {
        printf("%s", msg);
        fflush(stdin);
        gets(cad);//lee el un numero como caracter
        num = atoi(cad);//convierte el caracter en un numero

        if (num < li_in || li_sup)//imprime mensaje en caso de que no se valido el numero
        {
            system("CLS");
            printf("SOLO NUMEROS DEL 30 AL 70\n");
        }
    } while (num < li_in || num > li_sup);//valida que el numero sea correcto
    return num;
}

//llena un vector mediante el usuario
//RGA_Act8_01_932
void vect_1(int vect1[], int m)
{
    int i, val;//variables locales
    system("CLS");
    printf("LLENA EL VECTOR\n");
    printf("SOLO NUMEROS DEL 30 al 70\n");
    system("PAUSE");
    for (i = 0; i < m; i++)//ciclo que controla el llenado del vector
    {
        system("CLS");
        printf("Vec[%d]", i + 1);
        val = validar(" =  ", 30, 70);//manda a llamar la funcion validar
        vect1[i] = val;//asigna el valor validado al vector
    }
    system("CLS");
    printf("**************************\n");
    printf("El vector ha sido llenado\n");
    printf("**************************\n");
    system("PAUSE");
}
```

```c
//llena un vector con numeros aleatorios
//RGA_Act8_02_932
void vect_2(int vect2[], int lim_in, int lim_sup)
{
    int i, x, cont, j, ran;//variables locales
    x = (lim_sup - lim_in) + 1;

    for (i = 0; i < 10; i++)//llama a los vectores mediante su indice
    {
        do//ciclo encargado de validar que no se repitan los numeros
        {
            cont = 0;//se encargara de guardar la validacion
            ran = (rand() % x) + lim_in;//genera numero random

            for (j = 0; j < i; j++)//llama a los vectores mediante su indice
            {
                if (vect2[j] == ran)//valida que el valor random no se igual a los ya generados
                {
                    cont = 1;
                }
            }

        } while (cont != 0);//repite el ciclo en caso de que se repitan los numeros

        vect2[i] = ran;//se asigna el valor random al vecotor
    }
    system("CLS");
    printf("**************************\n");
    printf("El vector ha sido llenado\n");
    printf("**************************\n");
    system("PAUSE");
}
```

```c

//llena un vector con los vectores 1 y 2
//RGA_Act8_03_932
void vect_3(int vect1[], int vect2[], int vect3[], int m)
{
    int i, j;//variables locales
    system("CLS");
    for (i = 0, j = 10; i < m / 2; i++, j++)//se encargara de acceder a los indices de los vectores
    {
        vect3[i] = vect1[i];//asigna el valor del vector 1 al vector 3
        vect3[j] = vect2[i];//asigna el valor del vector 2 al vector 3
    }

    printf("**************************\n");
    printf("El vector ha sido llenado\n");
    printf("**************************\n");
    system("PAUSE");
}

//imprime todos los vectores
//RGA_Act8_04_932
void imprimir_vetores(int vect1[], int vect2[], int vect3[], int m, int n)
{
    int i;//variable local, se encargara de acceder a los indices de los vectores
    system("CLS");
    printf("VECTOR 1\n");
    printf("---------------------\n");
    for (i = 0; i < m; i++)//ciclo que controla la impresion del vector 1
    {
        printf("[%d]\n",vect1[i]);//imprime el vector 1
    }
    printf("---------------------");
    printf("\nVECTOR 2\n");
    printf("---------------------\n");

    for (i = 0; i < m; i++)//ciclo que controla la impresion del vector 2
    {
        printf("[%d]\n",vect2[i]);//imprime el vector 2
    }
    printf("---------------------");
    printf("\nVECTOR 3\n");
    printf("---------------------\n");
    for (i = 0; i < n; i++)//ciclo que controla la impresion del vector 3
    {
        printf("[%d]\n",vect3[i]);//imprime el vector 3
    }
    system("PAUSE");
}
```

```
205
206 //llena matriz con vector 1 y 2
207 //RGA_Act8_05_932
208 void matriz(int vect1[], int vect2[], int mtz[][4], int m, int v)
209 {
210     system("CLS");
211     int fil, colum,j=0, i = 0;//variables locales que accederan a los indices de los vectores y la matriz
212     for (fil = 0; fil < m; fil++)//ciclo que controla las filas de la matriz
213     {
214
215         for (colum = 0; colum < 4; colum++)//ciclo que controla las columnas de la matriz
216         {
217             if (i<v)//controla con que se llenara la matriz
218             {
219                 mtz[fil][colum]=vect1[i];//asigna el valor de vect1 a la matriz
220                 i++;//aumenta contador para acceder al indice del vect1
221             }
222             else
223             {
224                 mtz[fil][colum]=vect2[j];// asigna el valor de vect2 a la matriz
225                 j++;//aumenta contador para acceder al indice del vect2
226             }
227
228         }
229     }
230      printf("**************************\n");
231     printf("La matriz ha sido llenado\n");
232     printf("**************************\n");
233     system("PAUSE");
234 }
235
236 //imprime la matriz
237 //RGA_Act8_06_932
238 void imprimir_matriz(int mtz[][4],int m)
239 {
240     int fil,colum;//valiables locales
241     system("CLS");
242     printf("\tMATRIZ 4x4\n");
243     for ( fil = 0; fil < m; fil++)//ciclo que controla las filas de la matriz
244     {
245
246         for ( colum = 0; colum < 4; colum++)//ciclo que controla las columnas de la matriz
247         {
248             printf("[%d]\t",mtz[fil][colum]);//imprime matriz
249         }
250         printf("\n");
251
252     }
253     system("PAUSE");
254
255 }
256
```

```
102   //llena un vector mediante el usuario
103   //RGA_Act8_01_932
104   void vect_1(int vect1[], int m)
105   {
106       int i, val;//variables locales
107       system("CLS");
108       printf("LLENA EL VECTOR\n");
109       printf("SOLO NUMEROS DEL 30 al 70\n");
110       system("PAUSE");
111       for (i = 0; i < m; i++)//ciclo que controla el llen
112       {
113           system("CLS");
114           printf("Vec[%d]", i + 1);
115           val = validar(" =  ", 30, 70);//manda a llamar
116           vect1[i] = val;//asigna el valor validado al ve
117       }
118       system("CLS");
119       printf("**************************\n");
120       printf("El vector ha sido llenado\n");
121       printf("**************************\n");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
MENU
1.- LLENA VECTOR 1(MANUALMENTE)
2.- LLENA VECTOR 2(ALEATORIAMENTE)
3.- LLENA VECTOR 3(VECTOR 1 Y VECTOR 2)
4.- IMPRIMIR VECTOES
5.- LLENA MATRIZ(VECTOR 1 Y VECTOR 2)
6.- IMPRIMIR MATRIZ
0.- SALIR
ESCOGE UNA OPCION: 1
```

```
102  //llena un vector mediante el usuario
103  //RGA_Act8_01_932
104  void vect_1(int vect1[], int m)
105  {
106      int i, val;//variables locales
107      system("CLS");
108      printf("LLENA EL VECTOR\n");
109      printf("SOLO NUMEROS DEL 30 al 70\n");
110      system("PAUSE");
111      for (i = 0; i < m; i++)//ciclo que controla el llenado del vector
112      {
113          system("CLS");
114          printf("Vec[%d]", i + 1);
115          val = validar(" =  ", 30, 70);//manda a llamar la funcion validar
116          vect1[i] = val;//asigna el valor validado al vector
117      }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
LLENA EL VECTOR
SOLO NUMEROS DEL 30 al 70
Presione una tecla para continuar . . .
```

```
113         system("CLS");
114         printf("Vec[%d]", i + 1);
115         val = validar(" =  ", 30, 70);//manda a l
116         vect1[i] = val;//asigna el valor validado
117     }
    system("CLS");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Vec[4] =  22
```

```
110     system("PAUSE");
111     for (i = 0; i < m; i++)//ciclo que controla el llena
112     {
113         system("CLS");
114         printf("Vec[%d]", i + 1);
115         val = validar(" =  ", 30, 70);//manda a llamar
116         vect1[i] = val;//asigna el valor validado al vec
117     }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
SOLO NUMEROS DEL 30 AL 70
 =
```

```
111     for (i = 0; i < m; i++)//ciclo que controla el llenado
112     {
113         system("CLS");
114         printf("Vec[%d]", i + 1);
115         val = validar(" =  ", 30, 70);//manda a llamar la
116         vect1[i] = val;//asigna el valor validado al vecto
117     }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
**************************
El vector ha sido llenado
**************************
Presione una tecla para continuar . . .
```

```
125 //llena un vector con numeros aleatorios
126   //RGA_Act8_02_932
127 void vect_2(int vect2[], int lim_in, int lim_sup)
128   {
129       int i, x, cont, j, ran;//variables locales
130       x = (lim_sup - lim_in) + 1;
131
132       for (i = 0; i < 10; i++)//llama a los vectores media
133       {
134           do//ciclo encargado de validar que no se repitan
135           {
136               cont = 0;//se encargara de guardar la valida
137               ran = (rand() % x) + lim_in;//genera numero
138
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
MENU
1.- LLENA VECTOR 1(MANUALMENTE)
2.- LLENA VECTOR 2(ALEATORIAMENTE)
3.- LLENA VECTOR 3(VECTOR 1 Y VECTOR 2)
4.- IMPRIMIR VECTOES
5.- LLENA MATRIZ(VECTOR 1 Y VECTOR 2)
6.- IMPRIMIR MATRIZ
0.- SALIR
ESCOGE UNA OPCION: 2
```

```
125   //llena un vector con numeros aleatorios
126   //RGA_Act8_02_932
127   void vect_2(int vect2[], int lim_in, int lim_
128   {
129       int i, x, cont, j, ran;//variables locale
130       x = (lim_sup - lim_in) + 1;
131
132       for (i = 0; i < 10; i++)//llama a los vec
133       {
134           do//ciclo encargado de validar que no
135           {
136               cont = 0;//se encargara de guarda
137               ran = (rand() % x) + lim_in;//gen
138
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
**********
El vector ha sido llenado
**********
Presione una tecla para continuar . . .
```

```
158   //llena un vector con los vectores 1 y 2
159   //RGA_Act8_03_932
160   void vect_3(int vect1[], int vect2[], int vect3[], int
161   {
162       int i, j;//variables locales
163       system("CLS");
164       for (i = 0, j = 10; i < m / 2; i++, j++)//se encar
165       {
166           vect3[i] = vect1[i];//asigna el valor del vect
167           vect3[j] = vect2[i];//asigna el valor del vect
168       }
169
170       printf("*****************************\n");
171       printf("El vector ha sido llenado\n");
172       printf("*****************************\n");
173       system("PAUSE");
174   }
175
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
MENU
1.- LLENA VECTOR 1(MANUALMENTE)
2.- LLENA VECTOR 2(ALEATORIAMENTE)
3.- LLENA VECTOR 3(VECTOR 1 Y VECTOR 2)
4.- IMPRIMIR VECTOES
5.- LLENA MATRIZ(VECTOR 1 Y VECTOR 2)
6.- IMPRIMIR MATRIZ
0.- SALIR
ESCOGE UNA OPCION: 3
```

```
158   //llena un vector con los vectores 1 y 2
159   //RGA_Act8_03_932
160   void vect_3(int vect1[], int vect2[], int ve
161   {
162       int i, j;//variables locales
163       system("CLS");
164       for (i = 0, j = 10; i < m / 2; i++, j++)
165       {
166           vect3[i] = vect1[i];//asigna el valo
167           vect3[j] = vect2[i];//asigna el valo
168       }
169
170       printf("*****************************\n");
171       printf("El vector ha sido llenado\n");
172       printf("*****************************\n");
173       system("PAUSE");
174   }
175
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
*****************************
El vector ha sido llenado
*****************************
Presione una tecla para continuar . . .
```

```
176    //imprime todos los vectores
177    //RGA_Act8_04_932
178    void imprimir_vetores(int vect1[], int vec
179    {
180        int i;//variable local, se encargara d
181        system("CLS");
182        printf("VECTOR 1\n");
183        printf("----------------------\n");
184        for (i = 0; i < m; i++)//ciclo que con
185        {
186            printf("[%d]\n",vect1[i]);//imprim
187        }
188        printf("----------------------");
189        printf("\nVECTOR 2\n");
190        printf("----------------------\n");
191
192        for (i = 0; i < m; i++)//ciclo que con
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
MENU
1.- LLENA VECTOR 1(MANUALMENTE)
2.- LLENA VECTOR 2(ALEATORIAMENTE)
3.- LLENA VECTOR 3(VECTOR 1 Y VECTOR 2)
4.- IMPRIMIR VECTOES
5.- LLENA MATRIZ(VECTOR 1 Y VECTOR 2)
6.- IMPRIMIR MATRIZ
0.- SALIR
ESCOGE UNA OPCION: 4
```

```c
//imprime todos los vectores
//RGA_Act8_04_932
void imprimir_vetores(int vect1[], int vect2[], int vect3[], int m, int n)
{
    int i;//variable local, se encargara de acceder a los indices de los vectores
    system("CLS");
    printf("VECTOR 1\n");
    printf("--------------------\n");
    for (i = 0; i < m; i++)//ciclo que controla la impresion del vector 1
    {
        printf("[%d]\n",vect1[i]);//imprime el vector 1
    }
    printf("----------------------");
    printf("\nVECTOR 2\n");
    printf("--------------------\n");

    for (i = 0; i < m; i++)//ciclo que controla la impresion del vector 2
    {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
VECTOR 1
--------------------
[35]
[65]
[68]
[49]
[58]
[70]
[30]
[61]
[50]
[44]
--------------------
VECTOR 2
--------------------
[2]
[20]
[7]
[9]
[5]
[3]
[8]
[10]
[16]
[11]
--------------------
```

```
187     }
188     printf("---------------------");
189     printf("\nVECTOR 2\n");
190     printf("---------------------\n");
191
192     for (i = 0; i < m; i++)//ciclo que controla la impresion de
193     {
194         printf("[%d]\n",vect2[i]);//imprime el vector 2
195     }
196     printf("---------------------");
197     printf("\nVECTOR 3\n");
198     printf("---------------------\n");
199     for (i = 0; i < n; i++)//ciclo que controla la impresion de
200     {
201         printf("[%d]\n",vect3[i]);//imprime el vector 3
202     }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
--------------------
VECTOR 3
--------------------
[35]
[65]
[68]
[49]
[58]
[70]
[30]
[61]
[50]
[44]
[2]
[20]
[7]
[9]
[5]
[3]
[8]
[10]
[16]
[11]
Presione una tecla para continuar . . .
```

```
206    //llena matriz con vector 1 y 2
207    //RGA_Act8_05_932
208    void matriz(int vect1[], int vect2[], int mtz[][4],
209    {
210        system("CLS");
211        int fil, colum,j=0, i = 0;//variables locales q
212        for (fil = 0; fil < m; fil++)//ciclo que contro
213        {
214
215            for (colum = 0; colum < 4; colum++)//ciclo
216            {
217                if (i<v)//controla con que se llenara l
218                {
219                    mtz[fil][colum]=vect1[i];//asigna e
220                    i++;//aumenta contador para acceder
221                }
222                else
223                {
224                    mtz[fil][colum]=vect2[j];// asigna
225                    i++;//aumenta contador para acceder
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
MENU
1.- LLENA VECTOR 1(MANUALMENTE)
2.- LLENA VECTOR 2(ALEATORIAMENTE)
3.- LLENA VECTOR 3(VECTOR 1 Y VECTOR 2)
4.- IMPRIMIR VECTOES
5.- LLENA MATRIZ(VECTOR 1 Y VECTOR 2)
6.- IMPRIMIR MATRIZ
0.- SALIR
ESCOGE UNA OPCION: 5
```

```
206  //llena matriz con vector 1 y 2
207  //RGA_Act8_05_932
208  void matriz(int vect1[], int vect2[], int mtz[][4], i
209  {
210      system("CLS");
211      int fil, colum,j=0, i = 0;//variables locales que
212      for (fil = 0; fil < m; fil++)//ciclo que controla
213      {
214
215          for (colum = 0; colum < 4; colum++)//ciclo qu
216          {
217              if (i<v)//controla con que se llenara la
218              {
219                  mtz[fil][colum]=vect1[i];//asigna el
220                  i++;//aumenta contador para acceder a
221              }
222              else
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
**************************
La matriz ha sido llenada
**************************
Presione una tecla para continuar . . .
```

```
236   //imprime la matriz
237   //RGA_Act8_06_932
238   void imprimir_matriz(int mtz[][4],int m)
239   {
240       int fil,colum;//valiables locales
241       system("CLS");
242       printf("\tMATRIZ 4x4\n");
243       for ( fil = 0; fil < m; fil++)//ciclo que
244       {
245
246           for ( colum = 0; colum < 4; colum++)//
247           {
248               printf("[%d]\t",mtz[fil][colum]);
249           }
250           printf("\n");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
MENU
1.- LLENA VECTOR 1(MANUALMENTE)
2.- LLENA VECTOR 2(ALEATORIAMENTE)
3.- LLENA VECTOR 3(VECTOR 1 Y VECTOR 2)
4.- IMPRIMIR VECTOES
5.- LLENA MATRIZ(VECTOR 1 Y VECTOR 2)
6.- IMPRIMIR MATRIZ
0.- SALIR
ESCOGE UNA OPCION: 6
```

```
235
236 //imprime la matriz
237 //RGA_Act8_06_932
238 void imprimir_matriz(int mtz[][4],int m)
239 {
240     int fil,colum;//valiables locales
241     system("CLS");
242     printf("\tMATRIZ 4x4\n");
243     for ( fil = 0; fil < m; fil++)//ciclo que controla las
244     {
245
246         for ( colum = 0; colum < 4; colum++)//ciclo que co
247         {
248             printf("[%d]\t",mtz[fil][colum]);//imprime mat
249         }
250         printf("\n");
251
252     }
253     system("PAUSE");
254
255 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
        MATRIZ 4x4
[35]    [65]    [68]    [49]
[58]    [70]    [30]    [61]
[50]    [44]    [2]     [20]
[7]     [9]     [5]     [3]
Presione una tecla para continuar . . .
```