UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Ingeniería en computación

Actividad 9

Materia: Programación estructurada

ALUMNO: Villalobos Ensaldo Luis Daniel
MATRÍCULA: 368617
GRUPO: 441

PROFESOR: Pedro Nunez Yepiz

Ensenada, Baja California a 9 de Abril de 2024.

```c
/* Luis Daniel Villalobos Ensaldo #368617
Actividad 9   22/03/2024    5/04/2024
Programacion estructurada   gpo:432
1.- LLENAR VECTOR .- V números aleatoriamente, de 100 al 200 (no
repetidos)
2.- LLENAR MATRIZ .- 4x4 con números aleatoriamente 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.
*/
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "luis.h"
#define N 4
#define V 15

//*** PROTOTIPOS DE FUNCIONES  ******
int valinum(int ri, int rf, char msge[]);
int msges();
void menu();
int vectrand(int vect[], int m, int ri, int rf);
int llenarmat(int mat[][N], int m, int n, int ri, int rf);
void printvect(int vect[], int m, char nombre[] = "Vector");
void printmat(int mat[][N], int m, int n, char nombre[] = "Matriz");
int ordvect(int vect[], int m);
int searchvect(int vect[], int m, int num, bool ordenado = 0);

//****  main principal  *********
int main()
{
    srand(time(NULL));
    menu();

    return 0;
}
```

```
//  *** DESARROLLO DE LAS FUNCIONES  ******
//*********************
int msges()
{
    int op;
    system("CLS");
    printf("   M  E   N   U \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 = valinum(0, 6, "ESCOGE UNA OPCION: ");
    return op;
}
//****************
void menu()
{
    int op, vect[V], mat[N][N], pos, num;
    bool vector = 0, matriz = 0, orden = 0;
    do
    {
        op = msges();
        switch (op)
        {
        case 1:
            system("CLS");
            vector = vectrand(vect, V, 100, 200);
            printf("Ya se lleno el vector\n");
            orden = 0;
            system("PAUSE");
            break;
        case 2:
            system("CLS");
            matriz = llenarmat(mat, N, N, 1, 16);
            printf("Matriz llenada\n");
            system("PAUSE");
            break;
        case 3:
            system("CLS");
            if (vector)
```

```
            {
                printvect(vect, V, "Vector aleatorio");
            }
            else
            {
                printf("No se ha llenado el vector\n");
            }
            system("PAUSE");

            break;
        case 4:
            system("CLS");
            if (matriz)
            {
                printmat(mat, N, N, "Matriz aleatoria");
            }
            else
            {
                printf("No se ha llenado la matriz\n");
            }
            system("PAUSE");

            break;
        case 5:
            system("CLS");
            if (vector)
            {
                orden = ordvect(vect, V);
                printf("Ya se ordeno el vector\n");
            }
            else
            {
                printf("No se ha llenado el vector\n");
            }
            system("PAUSE");

            break;
        case 6:
            system("CLS");
            if (vector)
            {
                num = valinum(100, 200, "Buscar valor entre
100-200:\n");
```

```
                pos = searchvect(vect, V, num, orden);
                if (pos == -1)
                {
                    printf("No se encontro el valor\n");
                }
                else
                {
                    printf("El valor esta en la posicion[%d]\n", pos);
                }
            }
            else
            {
                printf("No se ha llenado el vector\n");
            }
            system("PAUSE");

            break;
        }

    } while (op != 0);
}

//**********************
int vectrand(int vect[], int m, int ri, int rf)
{
    //  VARIALES LOCALES
    int rango, num, j, i;
    rango = (rf - ri) + 1;
    //  AQUI DESARROLLO PROGRAMA
    for (i = 0; i < m; i++)
    {
        vect[i] = (rand() % rango) + ri;
        num = vect[i];
        j = 0;
        while (j < i)
        {
            if (num == vect[j])
            {
                i--;
            }
            j++;
        }
    }
```

```
    return 1;
}

//*************************
int llenarmat(int mat[][N], int m, int n, int ri, int rf)
{

    //  VARIALES LOCALES
    int rango, num, compr[m * n], acum = 0, i, j;
    //  AQUI DESARROLLO PROGRAMA
    vectrand(compr, m * n, 1, 16);

    for (i = 0; i < m; i++)
    {
        j = 0;
        while (j < n)
        {
            mat[i][j] = compr[acum];
            acum++;
            j++;
        }
    }
    return 1;
}
//*****************************
void printvect(int vect[], int m, char nombre[])
{
    //  VARIALES LOCALES
    //  AQUI DESARROLLO PROGRAMA
    printstr(nombre);
    for (int i = 0; i < m; i++)
    {
        printf("i %d: [%d]\n", (i), vect[i]);
    }

    printf("\n");
}

void printmat(int mat[][N], int m, int n, char nombre[])
{
    //  VARIALES LOCALES
    //  AQUI DESARROLLO PROGRAMA
    printstr(nombre);
```

```
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            printf("[%d] ", mat[i][j]);
        }
        printf("\n");
    }

    printf("\n");
}

int ordvect(int vect[], int m)
{
    int temp, j;
    for (int i = 0; i < m - 1; i++)
    {
        for (j = i + 1; j < m; j++)
        {
            if (vect[j] < vect[i])
            {
                temp = vect[i];
                vect[i] = vect[j];
                vect[j] = temp;
            }
        }
    }
    return 1;
}

int searchvect(int vect[], int m, int num, bool ordenado)
{
    int i;

    if (ordenado)
    {
        int bajo = 0, centro, alto = m;
        do
        {
            centro = (bajo + alto) / 2;
            if (vect[centro] < num)
            {
```

```
                bajo = centro + 1;
            }
            else
            {
                if (vect[centro] == num)
                {
                    return centro;
                }
                else
                {
                    alto = centro - 1;
                }
            }
        } while (alto>bajo);
    }
    else
    {
        for (i = 0; i < m; i++)
        {
            if (vect[i] == num)
            {
                return i;
            }
        }
    }
    return -1;
}
```

```
   M  E   N   U
1.- llenar vector
2.- llenar matriz
3.- imprimir vector
4.- Imprimir matriz
5.- Ordenar vector
6.- Buscar valor en vector
0.- SALIR
ESCOGE UNA OPCION:
```

```
No se ha llenado el vector
Presione una tecla para continuar . . .
```

```
No se ha llenado la matriz
Presione una tecla para continuar . . .
```

```
'a se lleno el vector
'resione una tecla para continuar . . .
```

```
Matriz llenada
Presione una tecla para continuar . . .
```

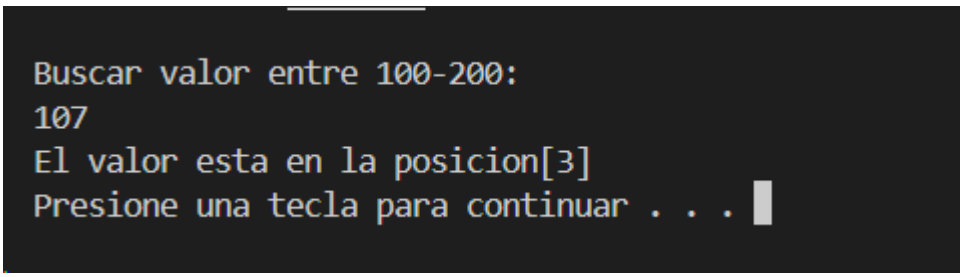
```
Vector aleatorio
i 0: [106]
i 1: [129]
i 2: [184]
i 3: [163]
i 4: [148]
i 5: [155]
i 6: [103]
i 7: [144]
i 8: [175]
i 9: [102]
i 10: [107]
i 11: [182]
i 12: [135]
i 13: [128]
i 14: [126]

Presione una tecla para continuar . . .
```

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

Presione una tecla para continuar . . .
```

```
Ya se ordeno el vector
Presione una tecla para continuar . . .
```

```
Vector aleatorio
i 0: [102]
i 1: [103]
i 2: [106]
i 3: [107]
i 4: [126]
i 5: [128]
i 6: [129]
i 7: [135]
i 8: [144]
i 9: [148]
i 10: [155]
i 11: [163]
i 12: [175]
i 13: [182]
i 14: [184]

Presione una tecla para continuar . . .
```

```
Buscar valor entre 100-200:
200
No se encontro el valor
Presione una tecla para continuar . . .
```


Buscar valor entre 100-200:
107
El valor esta en la posicion[3]
Presione una tecla para continuar . . .

Programa1

```
int vectrand(int vect[], int m, int ri, int rf)
{
    //  VARIALES LOCALES
    int rango, num, j, i;
    rango = (rf - ri) + 1;
    //  AQUI DESARROLLO PROGRAMA
    for (i = 0; i < m; i++)
    {
        vect[i] = (rand() % rango) + ri;
        num = vect[i];
        j = 0;
        while (j < i)
        {
            if (num == vect[j])
            {
                i--;
            }
            j++;
        }
    }
    return 1;
}
```

Programa2

```
int llenarmat(int mat[][N], int m, int n, int ri, int rf)
{
    //  VARIALES LOCALES
    int rango, num, compr[m * n], acum = 0, i, j;
    //  AQUI DESARROLLO PROGRAMA
    vectrand(compr, m * n, 1, 16);
    for (i = 0; i < m; i++)
    {
        j = 0;
        while (j < n)
        {
            mat[i][j] = compr[acum];
            acum++;
            j++;
        }
    }
    return 1;
}
```

Programa3

```
void printvect(int vect[], int m, char nombre[])
{
    //  VARIALES LOCALES
    //  AQUI DESARROLLO PROGRAMA
    printstr(nombre);

    for (int i = 0; i < m; i++)
    {
        printf("i %d: [%d]\n", (i), vect[i]);
    }
    printf("\n");
}
```

Programa4

```
void printmat(int mat[][N], int m, int n, char nombre[])
{
    //  VARIALES LOCALES
    //  AQUI DESARROLLO PROGRAMA
    printstr(nombre);

    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            printf("[%d] ", mat[i][j]);
        }
        printf("\n");
    }
    printf("\n");
}
```

Programa5

```
int ordvect(int vect[], int m)
{
    int temp, j;
    for (int i = 0; i < m - 1; i++)
    {
        for (j = i + 1; j < m; j++)
        {
            if (vect[j] < vect[i])
            {
                temp = vect[i];
                vect[i] = vect[j];
                vect[j] = temp;
            }
        }
    }
    return 1;
}
```

Programa6

```
int searchvect(int vect[], int m, int num, bool ordenado)
{
    int i;
    if (ordenado)
    {
        int bajo = 0, centro, alto = m;
        do
        {
            centro = (bajo + alto) / 2;
            if (vect[centro] < num)
            {
                bajo = centro + 1;
            }
            else
            {
                if (vect[centro] == num)
                {
                    return centro;
                }
                else
                {
                    alto = centro - 1;
                }
            }
        } while (alto>bajo);
    }
    else
    {
        for (i = 0; i < m; i++)
        {
            if (vect[i] == num)
            {
                return i;
            }
        }
```

```
    }
    return -1;
}
```