```c
// Lopez Michel Jose Alonso   Matricula: 370650
// Fecha inicio: 04/10/2023   Fecha fin: 09/10/2023

// En esta libreria llamada JoseLib.h usaremos las funciones del programa numero siete
// y aparte haremos un nuevas en base a las parecidas del programa numero 8. Todo con el fin de usar toda esta
//libreria en el programa numero nueve, en fin de solo mandar a llamara toda esta libreria y ya solo mandar a
// llamar a las funciones que hice aqui, igual como lo hariamos con librerias como stdio.h por ejemplo.

// JoseLib.h

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>
#include <stdbool.h> //Son esta las librerias que usaremos en este archivo.h

#define M 15   //Aqui definimso una constante de 15 valores para que se llenen de manera aleatoria el vector, que
               //en otra funcion servira para hacer matrices como la fila

#define N 4   //Aqui declaramos otra constante de que servira en las matrices como las columnas que tendra la misma.


//-----------------------------------------------------------------------------------------------------------
void mayusculas(char cadena[60]);
void minusculas(char cadena[60]);
void capital(char cadena[60]);
int tamaño(char cadena[60]);
void voltear(char cadena[60]);
void noespacios(char cadena[60]);
char permitir(char cadena[60]);
void todas(char cadena[60]);
void palindromo(char cadena[]);    //Todas las funciones antes hechas en el programa de la semana siete.
//-----------------------------------------------------------------------------------------------------------



//-----------------------------------------------------------------------------------------------------------
int validar_numero(int ri, int rf);
bool no_repetir(int vect[], int n);
void llenar_vect_aleatorio(int vect[], int m, int ri, int rf);
void llenar_matriz(int matriz[4][4], int m, int ri, int rf);
void imprimir_mat(int matriz[4][4], int m, int n);
void imprimir_vect(int vect[], int m);
bool no_repetir_m(int matriz[4][4],int n);
void ordenar(int vect[], int m);
int busq_sec(int vect[], int n, int num);
void buscar (int vect[], int m);        //Todas las funciones que usaremos en esta libreria para el programa.
//-----------------------------------------------------------------------------------------------------------



//-----------------------------------------------------------------------------------------------------------
// FUNCION DE MAYUSCULAS //
void mayusculas(char cadena[60])
{
    int i;
    printf("\n");
    for(i = 0; cadena[i] != '\0'; i++)
    {
        if (cadena[i] >= 97)
        {
            if (cadena[i] <= 122)
            {
                cadena[i] = cadena[i] - 32;
                printf("%c" ,cadena[i]);
            }
        }
    }
}

// FUNCION DE MINUSCULAS //
void minusculas(char cadena[60])
{
    int i;
    printf("\n");
    for(i = 0; cadena[i] != '\0'; i++)
    {
        if (cadena[i] >= 65)
        {
            if (cadena[i] <= 90)
            {
                cadena[i] = cadena[i] + 32;
                printf("%c" ,cadena[i]);
            }
        }
    }
```

```
}

// FUNCION QUE CONVIERTE CADENA EN "CAPITAL" //
void capital(char cadena[60])
{
    int i;
    printf("\n");
    cadena[0] = 67;
    cadena[1] = 65;
    cadena[2] = 80;
    cadena[3] = 73;
    cadena[4] = 84;
    cadena[5] = 65;
    cadena[6] = 76;
    cadena[7] = '\0';
    for(i = 0; cadena[i] != '\0'; i++)
    {
        printf("%c", cadena[i]);
    }
}

// FUNCION QUE CUENTA LOS CARACTERES DE LA CADENA //
int tamaño(char cadena[60])
{
    printf("\n");
    int i;
    for(i = 0; cadena[i] != '\0'; i++);
    return i;
}

// FUNCION QUE VOLTEA LA CADENA AL REVES //
void voltear(char cadena[60])
{
    int i, t;
    printf("\n");
    t = tamaño(cadena);
    for (i = t-1; i >= 0; i--)
    {
        printf("%c", cadena[i]);
    }
}

// FUNCION QUE ELIMINA LOS ESPACIOS DE LA CADENA //
void noespacios(char cadena[60])
{
    int i, t, aux, espacio;
    printf("\n");
    t = tamaño(cadena);
    for(i = 0; i < t; i++)
    {
        if (cadena[i] == 32)
        {
            espacio = i;
            aux = i;
            while (cadena[aux] == 32 && aux < t - 1)
            aux ++;
            cadena[espacio] = cadena[aux];
            cadena[aux] = 32;
        }
    }

    for(i = 0; cadena[i] != '\0'; i++)
    {
        printf("%c", cadena[i]);
    }
}

// FUNCION QUE PERMITE O NO LA CADENA //
char permitir(char cadena[60])
{
    int i, t;
    t = tamaño(cadena);
    printf("\n");
    for(i = 0; cadena[i] != '\0'; i++)
    {

    if (cadena[0] == ' ' || cadena[t] == ' ')
    {
        printf("\nNO PERMITIDA, debe ser una cadenas sin espacios al inicio ni al final");
        cadena[i] = '\0';
    }

    if (cadena[i] == 32 && cadena[i-1] == 32)
    {
        printf("NO PERMITIDO, debe ser cadena sin espacios dobles");
        cadena[i] = '\0';
    }
```

```
    if (cadena[i] != 32 && (cadena[i] < 65 || cadena[i] > 122))
    {printf("\nNO PERMITIDO, solo caracteres alfabeticos");
        cadena[i] = '\0';
    }

    }printf("\n");
    for(i = 0; cadena[i] != '\0'; i++)
    {
        printf("%c", cadena[i]);
    }
    return cadena[60];

}

// FUNCION QUE REALIZA 5 ACCIONES DE LAS FUNCIONES ANTERIORES //
void todas(char cadena[60])
{
    int i;
    char original[60], original2[60], original3[60], original4[60];
    for (i = 0; cadena[i] != '\0'; i++)
    {
        original[i] = cadena[i];
        original2[i] = cadena[i];
        original3[i] = cadena[i];
        original4[i] = cadena[i];
    }
    noespacios(cadena);
    mayusculas(original);
    minusculas(original2);
    voltear(original3);
    capital(original4);

}
                                                                                //Todas las funciones antes hechas en el programa de la semana siete.
//-------------------------------------------------------------------------------------------------------


//*******************************Todas las funciones para el programa nueve*************************************************
//**************************************************************************************************************************

// FUNCION QUE VALIDA NUMEROS //
int validar_numero(int ri, int rf)
{
     system("CLS");
    int n;
    char xnum[30];  //Aqui declaramos las variables junto con la cadena de tipo char que usaremos.

    printf("\nDame un numero entre el %d y %d: ", ri, rf);
        scanf("%c", &xnum);     //Aqui pedimos un valor al usuario que este entre los limtes de los valores establecidos.

    fflush(stdin);  //Aqui limpiamos el valor del buffer, en caso de que se quiera llenar de basura la cadena.
    gets(xnum); //Aqui pedimos al usaurio un valor para la cadena.
    n = atoi(xnum);  //Aqui convertimos la cadena de valor caracter a valor entero on la variable xnum.

    if (n > rf)  //Si el valor convertido a entero es mayor al rango final establecido, estara fuera del rango.
    {
        printf("\nNUMERO FUERA DE RANGO");
        n = 70; //Y automaticamente nuestro valor entero n sera igual a 70.
    }

    if (n < ri) //Si el valor convertido a entero es menor al rango inicial establecido, sera un numero muy pequeñp para poder usar en el
rango.
    {
        printf("\nNUMERO MUY PEQUEÑO PARA EL RANGO");
        n = 30; //Y automaticamente nuestro valor entero n sera igual a 30.
    }

    return n; //Y como ocuparemos el valor n en posteriores programas lo retornamos.
    system("PAUSE");
}

// FUNCION QUE EVITA NUMEROS REPETIDOS EN VECTOR //
bool no_repetir(int vect[],int n) //Implementamos una funcion que sirva con bandera solo para que nos erepitan los numeros.
{
    int i;
    for (i = 0; i < 15; i++) //El programa comienza en 0,seguira mientras sea menor a 15 y aumenta de uno en uno.
    {
        if (n == vect[i]) //Aqui le decimos que el valor retornado n si es igual al tamaño si seran iguales.
        {
            return true;
        }
    }
    return false;  //pero si son diferentes el programa continuara igual
}

// FUNCION QUE LLENA VECTOR ALEATORIAMENTE //
```

```c
void llenar_vect_aleatorio(int vect[], int m, int ri, int rf)
{
    system("CLS");
    int rango, i, n;
    rango = (rf - ri) + 1;  //Declaramos las varibales y declaramos el rango como la resta del rango final menos le rango incial mas uno.

    srand(time(NULL)); //declaramos el tiempo para poder hacer que se escogan las variables aleatoriamente.

    for(i = 0; i < m; i++) //Declaramos que comienze en 0,que siga mientras nuestro valor i iniciado en 0 sea menor que 15 o m y haremos que
avance de uno en uno.
    {
        do //Y hacemos un do while para que haga que el rango sea aleatorio.
        {
            n = (rand()%rango) + ri; //Nuestra variable n sera el resultado de un valor aleatorio mas el
                                     // rango inicial.

        } while (no_repetir(vect, n)); //y declaramos en el while nuestra funcion para que no se repitan
                            // los valores aleatorios del vector.
        vect[i] = n;  //Y diremos que el tamaño del vector[i] sera igual a nuestra variable de tipo entero n.
    }
    printf("Los valores aleatorios ya se ha puesto correctamente.\n");

    system("PAUSE");
}

// FUNCION QUE LLENA MATRIZ DE FORMA ALEATORIA //
void llenar_matriz(int matriz[4][4], int m, int ri, int rf)
{
    system("CLS");
    int rango, i, n, j;
    rango = (rf - ri) + 1; //Como en la funcion anterior igual declaramo la formula para el rango.

    srand(time(NULL));  //y declaramos una funcion que nos ayudara para que funcionen los valores aleatorios.

    //Declaramos una matriz de cuatro por cuatro:
    for(j = 0; j < 4; j++) //Declaramos que comienze en 0,que siga mientras nuestro valor j iniciado en 0 sea menor que 4 y haremos que avance
de uno en uno.
    {
        for(i = 0; i < 4; i++) //Declaramos que comienze en 0,que siga mientras nuestro valor i iniciado en 0 sea menor que 4 y haremos que
avance de uno en uno.
        {
            do{
                n = (rand()%rango) + ri;

              } while (no_repetir_m(matriz, n));//y declaramos en el while nuestra funcion para que no se repitan
                            // los valores aleatorios del vector.

        matriz[j][i] = n;  //Le damos la forma de matriz, y le decimos que sea igual a n.
        }
    }

    printf("Los valores aleatorios ya se ha puesto correctamente en la Matriz.\n");
    system("PAUSE");
}

// FUNCION QUE IMPRIME LA MATRIZ //
void imprimir_mat(int matriz[4][4], int m, int n)
{
     system("CLS");

    printf("\nMATRIZ");
    int j, i;
    printf("\n");
    for (j = 0; j < m; j++) //Declaramos que comienze en 0,que siga mientras nuestro valor j iniciado en 0 sea menor que m o nuestra fila y
haremos que avance de uno en uno.
    {
        printf("\n["); //hacemos que sea la forma de matriz los valores.
        for(i = 0; i < n; i++) //Declaramos que comienze en 0,que siga mientras nuestro valor i iniciado en 0 sea menor que n o nuestra
columna y haremos que avance de uno en uno.
        {
            printf("%d, ", matriz[j][i]); //Imprimimos dicha matriz para poder visualizar los valores.
        }
        printf("]");
    }
    printf("\n");

    system("PAUSE");
}

// FUNCION QUE IMPRIME VECTOR //
void imprimir_vect(int vect[], int m)
{
    system("CLS");

    int i;
    printf("\nVECTOR");
```

```
    printf("\n[");
    for (i = 0; i < m; i++) //Declaramos que comienze en 0,que siga mientras nuestro valor i iniciado en 0 sea menor que m o nuestra fila y haremos que avance de uno en uno.
    {
        printf("%d, ", vect[i]); //imprimimos los valores aleatorios del vector.
    }
    printf("]");

    printf("\n");

    system("PAUSE");
}

// FUNCION QUE EVITA NUMEROS REPETIDOS EN MATRIZ //
bool no_repetir_m(int matriz[4][4],int n)
{
    //Ahora parecida a la funcion de antes, ahora haremos que no se repitan los valores pero en la matriz.
    int i, j; //Declaramos las variables para la matriz.

    for (j = 0; j < 4; j++) //Declaramos que comienze en 0,que siga mientras nuestro valor j iniciado en 0 sea menor que 4 osea los valores de las filas y haremos que avance de uno en uno.
    {
        for (i = 0; i < 4; i++) //Declaramos que comienze en 0,que siga mientras nuestro valor i iniciado en 0 sea menor que 4 osea los valores de las columnas y haremos que avance de uno en uno.
        {
            if (n == matriz[j][i]) //Ahora declaramos una condicion que diga que si nuestra variable n es igual a la matriz 4+4 de las variables i y j entonces la bandera lo reconocera como identicos.
            {
                return true; //La bandera los reconocera como numeros identicos.
            }
        }
    }
    return false; //De lo contrario el programa continua normalmente.
}

// FUNCION QUE ORDENA EL VECTOR //
void ordenar(int vect[], int m)
{
    system("CLS");
    int temp, i, j;
    for (i = 0; i < m - 1; i++)//Declaramos que comienze en 0,que siga mientras nuestro valor i iniciado en 0 sea menor que m o nuestra fila menos uno y haremos que avance de uno en uno.
    {
        for (j = i + 1; j < m; j++) //Declaramos que comienze en la variable i o 0 á 1,que siga mientras nuestro valor j sea menor que m o nuestra fila y haremos que avance de uno en uno.
        {
            if (vect[j] < vect[i]) //Si el vector de j es menor que el de i
            {
                temp = vect[i];   //haremos que nuestra varaible temp sea igual a el vect[i]
                vect[i] = vect[j]; //, tambien haremos que el vector de i sea igual que el de j
                vect[j] = temp; //y por ultimo haremos que el vector de jo sea igual a nuestra variable temp.
            }
        }
    }

    printf("\nVECTOR ORDENADO");
    printf("\n[");
    for (i = 0; i < m; i++) //Declaramos que comienze en 0,que siga mientras nuestro valor i iniciado en 0 sea menor que m o nuestra fila y haremos que avance de uno en uno.
    {
        printf("%d, ", vect[i]); //Imprimimos nuestro vector.
    }
    printf("]");
    printf("\n");

     printf("Los valores del vector se han ordenado correctamente.\n");
    system("PAUSE");
}

// FUNCION PARA BUSQUEDA SECUENCIA //
int busq_sec(int vect[], int n, int num)
{  //Usamos el metodo de busqueda secuancial visto enteriormente en clase.
    system("CLS");
    int i;
    n = num;  //Declaramos las variables y decimos que nuestra variable n de las columnas sera igual a la variable num.

    for (i = 0; i < n; i++) //Declaramos que comienze en 0,que siga mientras nuestro valor i iniciado en 0 sea menor que n o nuestra columna y haremos que avance de uno en uno.
    {
        if(num == vect[i]) //Decimos que si numero es igual al vector de i
        {
            return i; //retorne la i.
        }
    }
    return -1; //Ahora haremos que retorne la i pero negativa.

    system("PAUSE");
```

```
}

// FUNCION PARA BUSCAR NUMERO //
void buscar (int vect[], int m)
{
     system("CLS");
    int num, x, i; //Declaramos las variables.

    num = validar_numero(100, 200); //Decimos que nuestra variable numero sea igual a la funcion validar numeros
    x = busq_sec(vect, M, num); //Y declaramos a x como que sea igual a la funcion busqueda secuencial.

    if (x != -1) //Y condicionamos que si la funcion busqueda secuencial es diferente de -1
    {

        printf("\nSi existe"); //Entonces si es un numero existente
        printf("\n%d esta en el indice %d", num, x); //y especificaremos el indice en donde estara.

    }

    else
    {
        //Y en su defecto si es == a -1 entonces no existira.
        printf("\nNo existe");

    }
    system("PAUSE");
}


//*****************************************************************************************************************************
```