

TITLE

AUTHOR
Version 1.0.0
CREATEDATE

Tabla de contenidos

Table of contents

Indice de archivos

Lista de archivos

Lista de todos los archivos con descripciones breves:

C:/Users/INSPIRON/Desktop/SII/menu1/menu.c3
---	--------

Documentación de archivos

Referencia del Archivo

C:/Users/INSPIRON/Desktop/SII/menu1/menu.c

```
#include <stdio.h>
#include <string.h>
#include <windows.h>
```

Funciones

- int **main** ()
- void **cesar1** ()
- void **cesar2** ()
- void **Escitala** (char cadena[])
- void **Escitala2** (char codpal[])
- void **ascii** ()
- void **ascci2** ()
- void **morse** ()
- void **morse2** ()

Variables

- FILE * **dat1**
- FILE * **dat2**
- char **nombre1** [50]
- char **nombre2** [50]
- char **c**
- char **l**
- int **trans**

Documentación de las funciones

void ascci2 ()

Definición en la línea 485 del archivo menu.c.

```
{
    system("cls");

    printf("Nombre del fichero a codificar a ascii: ");
    scanf("%s", &nombre1);
    printf("Nombre del fichero donde guardar codificado: ");
    scanf("%s", &nombre2);

    if (fopen(nombre1,"r")==NULL)
        printf("\nEl fichero no existe.\n\n");

    else{

        dat1=fopen(nombre1,"r");/*Se abre en modo lectura*/
        dat2=fopen(nombre2,"w");/*Se abre en modo escritura*/
        fscanf(dat1,"%d",&c);

        while (!feof(dat1)){
```

```

        l = toascii(c);

        /*Se escribe el caracter codificado*/
        fprintf(dat2,"%c",l);

        /*Se lee un nuevo caracter*/
        fscanf(dat1,"%d",&c);

    }//end of while

    fclose(dat1);
    fclose(dat2);/*Se cierran ambos ficheros*/

}//end of else
}

```

void ascii ()

Definición en la línea 451 del archivo menu.c.

```

{

    system("cls");

    printf("Nombre del fichero a codificar a ascii: ");
    scanf("%s", &nombre1);
    printf("Nombre del fichero donde guardar codificado: ");
    scanf("%s", &nombre2);

    if (fopen(nombre1,"r")==NULL)
        printf("\nEl fichero no existe.\n\n");
    else{

        dat1=fopen(nombre1,"r");/*Se abre en modo lectura*/
        dat2=fopen(nombre2,"w");/*Se abre en modo escritura*/
        fscanf(dat1,"%c",&c);

        while (!feof(dat1)){// lee el archivo hasta que finalice el texto

            /*Se escribe el caracter codificado*/
            fprintf(dat2,"%d ",c);

            /*Se lee un nuevo caracter*/
            fscanf(dat1,"%c",&c);

        }//end of while

        fclose(dat1);
        fclose(dat2);/*Se cierran ambos ficheros*/

    }//end of
}

```

void cesar1 ()

Definición en la línea 297 del archivo menu.c.

```

{

    system("cls");//limpiador de pantalla

    printf("Seleccione translacion de encriptacion: ");//impresiones
    scanf("%d", &trans);
    printf("Nombre del fichero a codificar: ");
    scanf("%s", &nombre1);
}

```

```

printf("Nombre del fichero donde guardar codificado: ");
scanf("%s", &nombre2);

if (fopen(nombre1,"r")==NULL)
    printf("\nEl fichero no existe.\n\n");
else{

    dat1=fopen(nombre1,"r");/*Se abre en modo lectura*/
    dat2=fopen(nombre2,"w");/*Se abre en modo escritura*/
    fscanf(dat1,"%c",&c);//un salto de linea

    while (!feof(dat1)){
        /*Se codifica el caracter*/
        if ((c>='A')&&(c<='Z')){
            c=((c+trans-'A')%('Z'-'A'+1))+ 'A';
        }//end of if
        else if ((c>='a')&&(c<='z')){
            c=((c+trans-'a')%('z'-'a'+1))+ 'a';
        }//end of if

        /*Se escribe el caracter en el codificado*/
        fprintf(dat2,"%c",c);

        /*Se lee un nuevo caracter*/
        fscanf(dat1,"%c",&c);
    }//end of while

    fclose(dat1);
    fclose(dat2);/*Se cierran ambos ficheros*/

} //end of else
}

```

void cesar2 ()

Definición en la línea 338 del archivo menu.c.

```

{

system("cls");

printf("Seleccione translacion de encriptacion: ");//impresion
scanf("%d", &trans);//mandar
printf("Nombre del fichero a codificar: ");//impresion
scanf("%s", &nombre1);
printf("Nombre del fichero donde guardar codificado: ");
scanf("%s", &nombre2);

if (fopen(nombre1,"r")==NULL)
    printf("\nEl fichero no existe.\n\n");
else{

    dat1=fopen(nombre1,"r");/*Se abre en modo lectura*/
    dat2=fopen(nombre2,"a");/*Se abre en modo escritura*/
    fscanf(dat1,"%c",&c);

    while (!feof(dat1)){

        /*Se codifica el caracter*/
        if ((c>='A')&&(c<='Z')){
            c=((c-trans+'A')%('Z'+'A'-1))- 'A';
        }//end of if
        else if ((c>='a')&&(c<='z'))
        {
            c=((c-trans+'a')%('z'+'a'-1))- 'a';//
        }//end of if

        /*Se escribe el caracter en el codificado*/
        fprintf(dat2,"%c",c);
    }
}

```



```

        /*Se lee un nuevo caracter*/
        fscanf(dat1,"%c",&c);
    }//end of while

    fclose(dat1);
    fclose(dat2);/*Se cierran ambos ficheros*/
}//end of else
}

```

void Escitala (charcadena[])

Definición en la línea 381 del archivo menu.c.

```

{

    int i, j, k=0,x=0,y=0;

    printf("Nombre del fichero donde guardar codificado: ");
    scanf("%s", &nombre2);
    dat2=fopen(nombre2,"wt");

    printf("Numero de Caras: ");
    scanf("%d",&x);
    printf("Largo del cilindro: ");
    scanf("%d",&y);
    char escitala[y][x];

    for(i=0; i<y; i++){
        for(j=0; j<x; j++){
            escitala[i][j] = cadena[k];
            k++;
        }
    }

    for(i=0; i<x; i++){
        for(j=0; j<y; j++){
            fprintf(dat2,"%c",escitala[j][i]);
        }
    }
    fclose(dat2);
}

```

void Escitala2 (charcodpa[])

Definición en la línea 412 del archivo menu.c.

```

{

    int i, j, k=0,x=0,y=0;
    FILE *dat2;

    printf("Nombre del fichero donde guardar codificado: ");
    scanf("%s", &nombre2);

    dat2=fopen(nombre2,"wt");/*Se abre en modo escritura*/

    printf("1.-Circunferencia: ");
    scanf("%d",&x);
    printf("2.-Largo: ");
    scanf("%d",&y);
    char escitala2[y][x];

    for(i=0; i<y; i++)
    {
        for(j=0; j<x; j++)
        {

```


[illegible]

```

        printf("\n");

        //end of while
        while(opcion >3);

    //end of while

//end if 2
else if(menu == 3){

    system("cls");
    system("COLOR 0E");

    opcion = 0;
    while(opcion < 3){

        if(opcion == 1){

            ascii();

        }//end of if opcion == 1

        else if(opcion == 2){

            ascci2();

        }//end of if opcion == 2

        do{

            printf("\n \n");
            printf("
            printf("
            printf("
            printf("
            printf("
            printf("
            printf("
            printf("
            printf("
            printf("\n      Que opcion eliges?: ");
            scanf("%d", &opcion);
            printf("\n");

        }//end of do
        while(opcion >3);

    }//end of while

//end if 3
else if(menu == 4){

    system("cls");
    system("COLOR 0B");

    opcion = 0;
    while(opcion < 3){

        if(opcion == 1){

            morse();

        }//end of if opcion == 1

        else if(opcion == 2){

            morse2();

        }//end of if opcion == 2

        do{

```


[illegible]

```
void morse ()
```

Definición en la línea 522 del archivo menu.c.

```

    {
system("cls");

printf("Nombre del fichero a codificar a morse: ");
scanf("%s", &nombre1);
printf("Nombre del fichero donde guardar codificado: ");
scanf("%s", &nombre2);

```

```

if (fopen(nombre1,"r")==NULL)
    printf("\nEl fichero no existe.\n\n");
else{

    dat1=fopen(nombre1,"r");/*Se abre en modo lectura*/
    dat2=fopen(nombre2,"w");/*Se abre en modo escritura*/
    fscanf(dat1,"%c",&c);

    //printf("La frase escrita en codigo morse es: ");
    int i;
    while (!feof(dat1)){

        switch (c){

            case 65: case 97:
                fprintf(dat2,".-",c);
                break;

            case 66: case 98:
                fprintf(dat2,"-...",c);
                break;

            case 67: case 99:
                fprintf(dat2,"-.-",c);
                break;

            case 68: case 100:
                fprintf(dat2,"-..",c);
                break;

            case 69: case 101:
                fprintf(dat2,".",c);
                break;

            case 70: case 102:
                fprintf(dat2,"...-",c);
                break;

            case 71: case 103:
                fprintf(dat2,"--",c);
                break;

            case 72: case 104:
                fprintf(dat2,"....");
                break;

            case 73: case 105:
                fprintf(dat2,"..",c);
                break;

            case 74: case 106:
                fprintf(dat2,".---",c);
                break;

            case 75: case 107:
                fprintf(dat2,"-.-",c);
                break;

            case 76: case 108:
                fprintf(dat2,"-..",c);
                break;

            case 77: case 109:
                fprintf(dat2,"---",c);
                break;

            case 78: case 110:
                fprintf(dat2,"-.",c);
                break;

```

```

case 79: case 111:
    fprintf(dat2,"---",c);
break;

case 80: case 112:
    fprintf(dat2,"--.",c);
break;

case 81: case 113:
    fprintf(dat2,"--.",c);
break;

case 82: case 114:
    fprintf(dat2,"-.",c);
break;

case 83: case 115:
    fprintf(dat2,"...",c);
break;

case 84: case 116:
    fprintf(dat2,"-",c);
break;

case 85: case 117:
    fprintf(dat2,"-.",c);
break;

case 86: case 118:
    fprintf(dat2,"...-",c);
break;

case 87: case 119:
    fprintf(dat2,"--",c);
break;

case 88: case 120:
    fprintf(dat2,"-..",c);
break;

case 89: case 121:
    fprintf(dat2,"-.-",c);
break;

case 90: case 122:
    fprintf(dat2,"--..",c);
break;

case 48:
    fprintf(dat2,"-----",c);
break;

case 49:
    fprintf(dat2,".----",c);
break;

case 50:
    fprintf(dat2,"...--",c);
break;

case 51:
    fprintf(dat2,"...--",c);
break;

case 52:
    fprintf(dat2,"....-",c);
break;

case 53:
    fprintf(dat2,".....",c);
break;

```



```

char *Puntero;

// Delimitador para romper la cadena morse en trozos
char Delimitador[5]=" ";

// Guardar los caracteres correspondientes al código morse
// en una matriz y guardar los caracteres correspondientes de la
// traducción en otra matriz asociando así los códigos de índice
// entre las dos matrices
strcpy(Morse[0],".-"); // A
Simbolos[0]='a';
strcpy(Morse[1],"-..."); // B
Simbolos[1]='b';
strcpy(Morse[2],"-.-."); // C
Simbolos[2]='c';
strcpy(Morse[3],"-.."); // D
Simbolos[3]='d';
strcpy(Morse[4],"."); // E
Simbolos[4]='e';
strcpy(Morse[5],"..-"); // F
Simbolos[5]='f';
strcpy(Morse[6],"--."); // G
Simbolos[6]='g';
strcpy(Morse[7],"..."); // H
Simbolos[7]='h';
strcpy(Morse[8],".."); // I
Simbolos[8]='i';
strcpy(Morse[9],".---"); // J
Simbolos[9]='j';
strcpy(Morse[10],".-.-"); // K
Simbolos[10]='k';
strcpy(Morse[11],".-.."); // L
Simbolos[11]='l';
strcpy(Morse[12], "--"); // M
Simbolos[12]='m';
strcpy(Morse[13], "-."); // N
Simbolos[13]='n';
strcpy(Morse[14], "----"); // O
Simbolos[14]='o';
strcpy(Morse[15], ".---."); // P
Simbolos[15]='p';
strcpy(Morse[16], "--.-"); // Q
Simbolos[16]='q';
strcpy(Morse[17], "-.-."); // R
Simbolos[17]='r';
strcpy(Morse[18], "..."); // S
Simbolos[18]='s';
strcpy(Morse[19], "-"); // T
Simbolos[19]='t';
strcpy(Morse[20], "..-"); // U
Simbolos[20]='u';
strcpy(Morse[21], "...-"); // V
Simbolos[21]='v';
strcpy(Morse[22], "-.-"); // W
Simbolos[22]='w';
strcpy(Morse[23], "-.-.-"); // X
Simbolos[23]='x';
strcpy(Morse[24], "-.-.-"); // Y
Simbolos[24]='y';
strcpy(Morse[25], "--.."); // Z
Simbolos[25]='z';
strcpy(Morse[26], "-.-.-"); // Ä
Simbolos[26]='Ä';
strcpy(Morse[27], "----."); // Ö
Simbolos[27]='Ö';
strcpy(Morse[28], "-.-.-"); // Ü
Simbolos[28]='Ü';
strcpy(Morse[29], " "); // Ch
Simbolos[29]=' '; // no se ha especificado el símbolo correcto
strcpy(Morse[30], "-----"); // 0
Simbolos[30]='0';

```

```

strcpy(Morse[31],".----"); // 1
Simbolos[31]='1';
strcpy(Morse[32],"..---"); // 2
Simbolos[32]='2';
strcpy(Morse[33],"...--"); // 3
Simbolos[33]='3';
strcpy(Morse[34],".---."); // 4
Simbolos[34]='4';
strcpy(Morse[35],".----."); // 5
Simbolos[35]='5';
strcpy(Morse[36],"-...."); // 6
Simbolos[36]='6';
strcpy(Morse[37], "--..."); // 7
Simbolos[37]='7';
strcpy(Morse[38], "----."); // 8
Simbolos[38]='8';
strcpy(Morse[39], "----."); // 9
Simbolos[39]='9';
strcpy(Morse[40], ".-.-."); // .
Simbolos[40]='.';
strcpy(Morse[41], "--..."); // ,
Simbolos[41]=',';
strcpy(Morse[42], "..--."); // ?
Simbolos[42]='?';
strcpy(Morse[43], ".-.-."); // !
Simbolos[43]='!';
strcpy(Morse[44], "----."); // :
Simbolos[44]=':';
strcpy(Morse[45], ".-.-."); // "
Simbolos[45]='"';
strcpy(Morse[46], "----."); // '
Simbolos[46]='\'';
strcpy(Morse[47], "-..."); // =
Simbolos[47]='=';

// Solicitar la frase en morse
printf("Nombre del fichero a codificar a morse: ");
scanf("%s", &nombre1);
printf("Nombre del fichero donde guardar codificado: ");
scanf("%s", &nombre2);

if (fopen(nombre1,"r")==NULL)
    printf("\nEl fichero no existe.\n\n");
else{

    dat1=fopen(nombre1,"r");/*Se abre en modo lectura*/
    dat2=fopen(nombre2,"w");/*Se abre en modo escritura*/
    fscanf(dat1,"%s",FraseMorse);

    while (!feof(dat1)){

        // Realizar copia de la cadena para trabajar con ella
        strcpy(CopiaFraseMorse,FraseMorse);

        // Descomposición de la cadena en trozos
        contadorfrasenormal = 0;

        // Primera llamada => Primer token
        Puntero = strtok( CopiaFraseMorse, Delimitador);

        // Mientras exista una palabra asociada al puntero recorrer la frase
        while(Puntero!=NULL) {

            // Buscar el carácter correspondiente a la cadena
            // Inicializar los contadores para recorrer la matriz de simbolos
            // y código morse hasta encontrar el carácter correspondiente
            contador=0;
            encontrado=0;

            while(contador<48 && encontrado!=1) {

```

```

        // Si se encuentra el carácter se cambia la var encontrado
        if (strcmp(Puntero,Morse[contador++]==0) encontrado=1;

    }//end of while

    // si se ha encontrado el carácter
    if (encontrado==1) {

        // se guarda el carácter en la variable de frase normal
        FraseNormal[contadorfrasenormal++]=Simbolos[contador-1];

        // se visualiza el símbolo correspondiente al carácter encontrado

    }//end of if

    // Siguientes llamadas de descomposición de la cadena
    Puntero = strtok( NULL, Delimitador );

    }//end of while

    // Finalizar la variable que guarda la frase traducida con el símbolo de
    // fin de cadena
    FraseNormal[contadorfrasenormal]='\0';

    // Visualizar la frase traducida

    /*Se escribe el caracter codificado*/
    fprintf(dat2,"%s",FraseNormal);

    /*Se lee un nuevo caracter*/
    fscanf(dat1,"%s",&FraseMorse);

    fprintf(dat2," ");

    }//end of while

    fclose(dat1);
    fclose(dat2);/*Se cierran ambos ficheros*/

    }//end of else

    }//end of morse2

```

Documentación de las variables

char c

Definición en la línea 6 del archivo menu.c.

FILE* dat1

Definición en la línea 5 del archivo menu.c.

FILE * dat2

Definición en la línea 5 del archivo menu.c.

char l

Definición en la línea 6 del archivo menu.c.

char nombre1[50]

Definición en la línea 6 del archivo menu.c.

char nombre2[50]

Definición en la línea 6 del archivo menu.c.

int trans

Definición en la línea 7 del archivo menu.c.

Índice

INDEX