

```

/*****TRABAJO
INFORMATICA*****/
Alumnos: Juan Andrés García Perozo y Marta García Gallardo
Grupo: A-104

```

Este trabajo se basa en la traducción de caracteres a código morse. La frase es introducida mediante un fichero. Se trabaja la frase mediante funciones y operaciones de cadenas de caracteres. Convierte la frase:

- Si se escribe en minúscula se convierte a mayúscula para trabajar el código morse con solo mayúscula.
- Las tildes son eliminadas de las letras que la lleven y las 'ñ' son sustituidas por 'n'.

```

*****/

```

```

#include<stdio.h>
#include<stdlib.h>
#include<windows.h>
#include<string.h>
#include<iostream>
#include<sstream>
#define _CRT_SECURE_NO_WARNINGS

```

```

void mayus (char cadena[]);/* Funcion que cambia las letras minusculas a
mayusculas para trabajar con
un solo segmento de la tabla ASCII*/

```

```

char Tildes(char); /* Funcion que quita las tildes a las vocales y cambia ñ por
una n y Ñ por N*/

```

```

void morse(char cadm[], int m); /* Funcion que traduce la cadena de caracteres a
morse*/

```

```

int main(void)
{
    int i = 0;
    char *cadi, cadiCod;
    FILE *pf;
    char lectura;//con esta variable leeremos uno a uno el fichero
    int w = 0;//con esta variable contamos cuantos caracteres ocupa el fichero
    para crear luego una cadena de caracteres de ese tamaño

```

```

printf("\t=====CODIGO MORSE=====\\n\\n");
printf("\t\\n");
printf("\t|A .-      | B -...   | C -. -.  | D -..   |\\n");
printf("\t|E .        | F ...   | G --    | H ....  |\\n");
printf("\t|I ..       | J .---   | K -. -  | L .-.   |\\n");
printf("\t|M --       | N -.     | O ---   | P .-.   |\\n");
printf("\t|Q -. -.   | R -.     | S ...   | T -     |\\n");
printf("\t|U ..-     | V ...-   | W ---   | X -.-   |\\n");
printf("\t|Y -.-    | Z ---.   |         |         |\\n");
printf("\t\\n\\n");

```

```

pf = fopen("FRASE.txt", "r");// abrimos un fichero SOLO LECTURA
if (pf == NULL)
{
    printf("Error al abrir el fichero\\n");
    exit(1);//Se termina la ejecución del programa en este punto
}
printf("Fichero abierto:\\n\\n");

```

```

    lectura = getc(pf); //lectura del fichero, el getc pasa por cada caracter
y cuando feof sea !=0 indicara el final del fichero
    while (feof(pf) == 0)
    {
        lectura = getc(pf);
        w++; //incrementa a cada caracter que hay
    }
    cadi = (char*)malloc(w + 1); //se crea cadena de caracteres del mismo
tamaño que la frase del fichero
    cadi[w] = '\0';
    rewind(pf); //se vuelve al inicio del fichero
    lectura = getc(pf);
    while (feof(pf) == 0) // se asigna cada caracter del fichero a nuestra
cadena de caracteres
    {
        for (i = 0; i < w; i++)
        {
            cadi[i] = lectura;
            lectura = getc(pf);
        }
    }
    i = 0; //reset de i
    while (cadi[i] != '\0') //arreglo de tildes y Ñ ñ en cadi mediante la
funcion tildes
    {
        cadiCod = cadi[i];
        cadi[i] = Tildes(cadiCod);
        i++;
    }
    printf("%s\n\n", cadi); //se imprime la cadena sin tildes y sin ñ's
    mayus(cadi);
    morse(cadi, w);

    system("pause");
}

```

```

char Tildes(char cadiCod)
{
    if (cadiCod == 'Á')
        cadiCod = 'A';
    else if (cadiCod == 'á')
        cadiCod = 'a';
    else if (cadiCod == 'É')
        cadiCod = 'E';
    else if (cadiCod == 'é')
        cadiCod = 'e';
    else if (cadiCod == 'Í')
        cadiCod = 'I';
    else if (cadiCod == 'í')
        cadiCod = 'i';
    else if (cadiCod == 'Ó')
        cadiCod = 'o';
    else if (cadiCod == 'ó')
        cadiCod = 'O';
    else if (cadiCod == 'Ú')
        cadiCod = 'U';
    else if (cadiCod == 'ú')
        cadiCod = 'u';
    else if (cadiCod == 'Ñ')
        cadiCod = 'N';
    else if (cadiCod == 'ñ')
        cadiCod = 'n';
}

```

```

        else
            cadiCod = cadiCod;
        return(cadiCod);
    }
    void mayus(char cadena[])
    {
        int i, des = 'a' - 'A';
        for (i = 0; cadena[i] != '\0'; i++)
        {
            if (cadena[i] >= 'a' && cadena[i] <= 'z')
            {
                cadena[i] = cadena[i] - des;
            }
        }
    }
    void morse(char cadm[], int m)
    {
        int i;
        for (i = 0; i < m; i++)
        {
            if (cadm[i] == 'A')
                printf(" .- ");
            if (cadm[i] == 'B')
                printf(" -... ");
            if (cadm[i] == 'C')
                printf(" -.-. ");
            if (cadm[i] == 'D')
                printf(" -.. ");
            if (cadm[i] == 'E')
                printf(" . ");
            if (cadm[i] == 'F')
                printf(" ..-. ");
            if (cadm[i] == 'G')
                printf(" --. ");
            if (cadm[i] == 'H')
                printf(" .... ");
            if (cadm[i] == 'I')
                printf(" .. ");
            if (cadm[i] == 'J')
                printf(" .- - - ");
            if (cadm[i] == 'K')
                printf(" -.- ");
            if (cadm[i] == 'L')
                printf(" .-.. ");
            if (cadm[i] == 'M')
                printf(" -- ");
            if (cadm[i] == 'N')
                printf(" -. ");
            if (cadm[i] == 'O')
                printf(" --- ");
            if (cadm[i] == 'P')
                printf(" .- -. ");
            if (cadm[i] == 'Q')
                printf(" --.- ");
            if (cadm[i] == 'R')
                printf(" .-. ");
            if (cadm[i] == 'S')
                printf(" ... ");
            if (cadm[i] == 'T')
                printf(" - ");
            if (cadm[i] == 'U')
                printf(" ..- ");

```

```
        if (cadm[i] == 'V')
            printf(" ...- ");
        if (cadm[i] == 'W')
            printf(" .- - ");
        if (cadm[i] == 'X')
            printf(" -.- - ");
        if (cadm[i] == 'Y')
            printf(" -.- - ");
        if (cadm[i] == 'Z')
            printf(" -.-. ");
        if (cadm[i] == ',')
            printf("||");
    }
    printf("\n\n");
}
```