

[illegible]

Una vez declaradas todas las variables se ejecuta un menú, con las opciones del código.

```
Morse.cpp  X
Morse (Ámbito global)

cout << "\n-----\nMenu de Opciones" << endl;
cout << "a. Espa\xA4ol a morse" << endl;
cout << "b. Morse a espa\xA4ol\n-----" << endl;

cout << "\nIngrese una opcion: ";
cin >> opcion;

switch (opcion)
{
case 'a': //espa\A4ol a morse;
    esp2morse(1, m);
    break;
case 'b': //morse a espa\A4ol
    morse2esp(1, m);
    break;
}
```

Para hacer que el usuario pueda elegir entre las opciones, se crea un “switch” con los casos en los que se introduce la función que les pertenece. Pues para distinguir de texto a morse y de morse a texto, se han creado dos funciones; esp2morse y morse2esp.

La primera de español a morse, en la cual, se inicia desde “i” hasta la cantidad de dígitos que haya seleccionado el usuario para la traducción de la cadena y una vez que el “i” llegue a 0, pasa al “j” que inicia hasta llegar al 37 (número de variables posibles en char). Hace esto, de tal modo que una vez que llega a un espacio que representará a un único caracter en ASCII, lo selecciona para compararlo entre todos los posibles de morse para encontrar su correspondiente y coger la posición equivalente en la lista ASCII.

```
Morse.cpp  X
Morse (Ámbito global) fileprint(char data[], const char con[], char c)

void esp2morse(char l[], char m[][7])
{
    char* cadena;
    int len = 0;
    printf("\nIndique la longitud de la cadena:");
    scanf_s("%d", &len);
    if (len != 0)
    {
        cadena = (char*)malloc(sizeof(int) * (len + 1)); //operador para asignar el tamaño exacto
        printf("Escribe un texto u oracion que quieras traducir a codigo morse:\n");
        getchar();
        gets_s(cadena, len + 1);
        fileprint(cadena, NULL, NULL);
        fileprint(NULL, "\n-----SE TRADUCE A-----\n", NULL);

        for (int i = 0; *(cadena + i) != '\0'; i++) //inicia desde i hasta la cantidad de dígitos que tenga la cadena
        {
            for (int j = 0; j < 37; j++) //una vez que el i llegue a 0, pasa al j que inicia hasta llegar al 37 (número de variables posibles en char)
            {
                if (*(cadena + i) == l[j]) //compara la cadena introducida con su correspondiente en traducción
                {
                    for (int k = 0; m[j][k] != '\0'; k++)
                    {
                        printf("%c", m[j][k]);
                        fileprint(&m[j][k], NULL, NULL);
                    }
                    //una vez que la encuentra, lo imprime
                    printf(" ");
                    break;
                }
            }
        }
    }
}
```

La segunda de morse a español, en esta función, en primer lugar cuenta cuántos caracteres hay entre cada espacio. Las señales de morse no tienen longitud fija por ello, hay que capturar todos los caracteres entre espacio según incrementa que juntos representaran 1 señal morse y por lo tanto, una letra o número al ser traducido. Se compara el contenido con el que está ordenado hasta que sean iguales. Es decir, que el algoritmo recorre las líneas según el input, caracter por caracter haciendo así que cada uno corresponda con su equivalente en morse, tanto letras como números.

```

Morse.cpp  x
Morse (Ámbito global) fileprint(char data[], const char con[], char c)

void morse2esp(char l[], char m[][7])
{
    int i = 0;
    char* cadena;
    int len;
    printf("\nIndique la longitud de la cadena:");
    scanf_s("%d", &len);
    cadena = (char*)malloc(sizeof(int) * (len + 1)); //operador para asignar el tamaño exacto
    printf("Escribe un texto u oracion que quieras traducir a codigo espa\xA4ol:\n");
    getchar();
    gets_s(cadena, len + 1);
    int distSpacio = 0;

    fileprint(cadena, NULL, NULL);
    fileprint(NULL, "\n-----SE TRADUCE A-----\n", NULL);
    while (*(cadena + i) != '\0')
    {
        // cuento cuantos caracteres hay entre espacio
        while (*(cadena + i + distSpacio) != ' ' && *(cadena + i + distSpacio) != '\0')
        {
            // si no es espacio continua incrementando, cogerá los símbolos morse que corresponde a 1 letra
            distSpacio++;
        }
        // necesitamos una variable temporal para almacenar el morse// + 1 por el '\0'
        char* temp;
        temp = (char*)calloc(distSpacio, 1);
        if (temp != NULL)
        {
            //printf("El tamaño de tem es %d, dist es %d\n", sizeof(temp), distSpacio);
            //volvemos hacia atras capturando el morse que corresponde a 1 letra
            for (int j = distSpacio; j > 0; j--)
            {
                *(temp + distSpacio - j) = *(cadena + i + distSpacio - j);
            }
            distSpacio++;
        }
        // necesitamos una variable temporal para almacenar el morse// + 1 por el '\0'
        char* temp;
        temp = (char*)calloc(distSpacio, 1);
        if (temp != NULL)
        {
            //printf("El tamaño de tem es %d, dist es %d\n", sizeof(temp), distSpacio);
            //volvemos hacia atras capturando el morse que corresponde a 1 letra
            for (int j = distSpacio; j > 0; j--)
            {
                *(temp + distSpacio - j) = *(cadena + i + distSpacio - j);
                //puts(temp);
            }
            // comparamos el obtenido con el que esta ordenado, m
            for (int x = 0; x < 37; x++)
            {
                if (compare(temp, m[x]) == 1)//son iguales
                {
                    fileprint(NULL, NULL, l[x]);
                    printf("%c", l[x]);
                    break;
                }
            }
            i += distSpacio + 1;
            distSpacio = 0;
            free(temp);
        }
        else
        {
            printf("\nERROR NO SE PUEDE ALOCAR MEMORIA\n");
        }
    }
}

```

¿Cómo utilizar el traductor?

Este traductor tiene un funcionamiento sencillo e intuitivo para poder ser usado por cualquier usuario que desea traducir de morse a texto y/o viceversa.

Una vez compilado el código y seleccionado la opción “Morse”, nos imprime en pantalla;

A screenshot of a Windows command prompt window titled "C:\Users\Claudia\source\repos\Morse\Debug\Morse.exe". The window displays a menu with the following text: "-----", "Opciones:", "1.Morse", "2.Binario", "3.Notas", "4.Salir", "-----", "Ingresa opcion: 1", "-----", "Menu de Opciones", "a. Español a morse", "b. Morse a español", "-----", and "Ingresa una opcion: _". The cursor is positioned at the end of the last line.

```
-----
Opciones:
1.Morse
2.Binario
3.Notas
4.Salir
-----
Ingresa opcion: 1
-----
Menu de Opciones
a. Español a morse
b. Morse a español
-----
Ingresa una opcion: _
```

La opción seleccionada se escribirá con su correspondiente letra en el aparatado que pone “Ingresa una opción”.

Cuando se elige una opción, por ejemplo la “a”, el programa pedirá la longitud de la cadena, en la que el usuario deberá tener en cuenta los espacios empleados y el número de caracteres usados.

A screenshot of the same Windows command prompt window. The menu options are the same as in the previous screenshot, but now the user has entered 'a' for the second option. The prompt "Ingresa una opcion: a" is shown. Below it, a new prompt "Indique la longitud de la cadena:_" is displayed, with the cursor at the end of the line.

```
-----
Opciones:
1.Morse
2.Binario
3.Notas
4.Salir
-----
Ingresa opcion: 1
-----
Menu de Opciones
a. Español a morse
b. Morse a español
-----
Ingresa una opcion: a
Indique la longitud de la cadena:_
```

Por consiguiente, ya se escribirá el texto u oración que de desee traducir.

```
C:\Users\Claudia\source\repos\Morse\Debug\Morse.exe

-----
Opciones:
1.Morse
2.Binario
3.Notas
4.Salir
-----
Ingresa opcion: 1

-----
Menu de Opciones
a. Español a morse
b. Morse a español
-----
Ingresa una opcion: a

Indique la longitud de la cadena:11
Escribe un texto u oracion que quieras traducir a codigo morse:
hola amigo _
```

Para completar la traducción se presiona la tecla “ENTER” del teclado y éste, instantáneamente, traducirá a morse teniendo en cuenta que los espacios equivalen a “/”.

```
C:\Users\Claudia\source\repos\Morse\Debug\Morse.exe

-----
Opciones:
1.Morse
2.Binario
3.Notas
4.Salir
-----
Ingresa opcion: 1

-----
Menu de Opciones
a. Español a morse
b. Morse a español
-----
Ingresa una opcion: a

Indique la longitud de la cadena:11
Escribe un texto u oracion que quieras traducir a codigo morse:
hola amigo
.... --- .-.. - / .- -- .. --. --- /
-----
Opciones:
1.Morse
2.Binario
3.Notas
4.Salir
-----
Ingresa opcion: _
```

Como se ve en el ejemplo, el programa continuará en bucle hasta que se seleccione la opción de “Salir”. Por otro lado, si se seleccionase la opción b (“Morse a español”), el procedimiento del programa sería el mismo, tan sólo variando que el usuario deberá escribir en morse. En este caso, es importante tener en cuenta que los espacios corresponderán con presionar: espacio, barra, espacio, en este orden.

En el ejemplo siguiente se muestra.

```
C:\Users\Claudia\source\repos\Morse\Debug\Morse.exe

-----
Opciones:
1.Morse
2.Binario
3.Notas
4.Salir
-----
Ingresa opcion: 1
-----

Menu de Opciones
a. Español a morse
b. Morse a español
-----

Ingresa una opcion: b

Indique la longitud de la cadena:60
Escribe un texto u oracion que quieras traducir a codigo español:
.- / . / .. / --- / ..-
a e i o u
```

Ejemplo 1. En este ejemplo, el usuario ha querido escribir las letras por separado. Para ello, ha empleado barras con espacios delante y detrás.

```
C:\Users\Claudia\source\repos\Morse\Debug\Morse.exe

Menu de Opciones
a. Español a morse
b. Morse a español
-----

Ingresa una opcion: b

Indique la longitud de la cadena:60
Escribe un texto u oracion que quieras traducir a codigo español:
.- . . --- .-
aeiou
```

Ejemplo 2. En este ejemplo, el usuario ha querido escribir las letras juntas. Por ello, no se emplean barras, sino que se escriben las letras con espacios.