

Este código se ha realizado mediante el lenguaje de programación C++. Donde podemos codificar y decodificar textos binarios a ASCII o viceversa.

Para ello hemos utilizado librerías como: `stdio.h` , `malloc.h` ,`string.h`, `stdlib.h` e `iostream` para el funcionamiento de los códigos.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <iostream>
#include <malloc.h>
```

```
using namespace std;
```

- `#include <stdio.h>`: contiene las funciones (macros) y tipos para manipular dato de entrada y de salida.
- `#include <string.h>`: para facilitar el uso de las cadenas de caracteres, ya que elimina dificultades que presentan los `char`.
- `#include <stdlib.h>`: contiene las funciones (macros) y tipos para uso general.
- `#include <iostream>`: contiene algoritmos estándar.
- `#include <malloc.h>`: necesaria para poder utilizar la función `malloc` que adjudica un espacio cuyo tamaño es especificado.
- `using namespace std;` : da acceso al espacio de nombres.

También se han creado funciones, para poder pasar de binario a ASCII , de ASCII a binario , para coger la longitud de caracteres y uno en la que podamos incluir ficheros (archivos) en la que nos permita al usuario pueda escribir y leer el documento para luego pasarlo a la impresora.

```
char* cogerTrozo(char frase[], int inicio, int fin);
char bin2ascii(char bin[9]);
char* convertir(int numero);
void callBinario();
```

En la función principal, está dividido en dos partes, en la que se le junta en solo uno mediante un menú para que el usuario pueda decidir lo que quiere realizar, para ello utilizamos un bucle `switch` en la que se encuentra dos opciones : en la opción A pasa de binario a ASCII y la opción pasa de ASCII a binario

```
void callBinario()
{
    fileprint(NULL, "\n-----NUEVA TRADUCCION BINARIO-----\n", NULL);

    printf("\n-----\na. binario a texto\n");
```

Caso A de binario a ASCII, pide al usuario la longitud que necesita, luego el usuario tiene que introducir la frase en binario en la que se divide el input en porciones que representa cada binario, una vez obtenido se pasa de binario a decimal y posteriormente a ASCII

```
printf("b. texto a binario\n-----\nIngresa opcion: ");
char a;
getchar();
scanf_s("%c", &a, 1);

switch (a)
{
case 'a':
{
    char* num;
    int resu = 0, len;
    printf("Introduce la longitud de la frase\n");
    scanf_s("%d", &len);
    num = (char*)malloc(7 * len + 1);
    printf("Introduce una frase formados por 0 o 1:\n");
    getchar();
    gets_s(num, 7 * len + 1);
    fileprint(num, NULL, NULL);
    fileprint(NULL, "\n-----SE TRADUCE A-----\n", NULL);
    for (int i = 0, k = 8; num[i] != '\\0' && num[k] != '\\0'; i += 7, k += 7)
    {
        char* trozo = cogerTrozo(num, i, k);
        resu = bin2ascii(trozo);
        printf("%c", resu);
    }
}
break;
```

Caso B de ASCII a binario, pide al usuario que introduzca una frase en ASCII donde se pasaría el char a int con la equivalencia directa de estas y posteriormente usaríamos un bucle para pasar de decimal a binario .

```

7         case 'b':
3             char numero;
3             char* numBinario;
3             char* ASCII;
1             int i = 0, len;
2             printf("Introduce la longitud de la frase\n");
3             scanf_s("%d", &len);
1             ASCII = (char*)malloc(len + 1);
5             printf("Escribe un texto u oracion con el lenguaje Ascii\n");
5             if (len + 1 != 0)
7             {
3                 getchar();
3                 gets_s(ASCII, len + 1);
3
1                 fileprint(ASCII, NULL, NULL);
2                 fileprint(NULL, "\n-----SE TRADUCE A-----\n", NULL);
3                 numBinario = (char*)calloc(9, sizeof(char));
1
5                 while (ASCII[i] != '\0')
5                 {
7                     numBinario = convertir((int)ASCII[i]);
3                     puts(numBinario);
3                     i++;
3                 }
1                 free(numBinario);
2                 break;
3             }
1
5         }
5     }

```

```

char* cogerTrozo(char frase[], int inicio, int fin)
{
    char* trozo;
    int distancia = fin - inicio;

    trozo = (char*)calloc(distancia + 1, 1);
    if (trozo != NULL)
    {
        for (int i = 0; i < distancia; i++)
        {
            *(trozo + i) = frase[inicio + i];
        }
        return trozo;
        fileprint(trozo, NULL, NULL);
    }
}

```

```

char bin2ascii(char bin[9])
{
    int resu = 0, i = 0;
    while (bin[i] != '\0')
    {
        if (bin[i] == '1')
        {
            resu += pow(2, 6 - i);
        }
        i++;
    }
    char c;
    c = resu;
    fprintf(NULL, NULL, c);
    return resu;
}

char* convertir(int numero)
{
    char* out;
    int i = 0;
    out = (char*)calloc(9, sizeof(char*));
    if (out != NULL)
    {
        //inicio a 0s para tener siempre un binario de 8 bits
        for (int i = 0; i < 8; i++)
        {
            *(out + i) = '0';
        }
        //paso de int a binario
        while (numero > 0)
        {
            if (numero % 2 == 0)
            {
                *(out + 7 - i) = '0';
            }
            else
            {
                *(out + 7 - i) = '1';
            }
            numero = numero / 2;
            for (int i = *out - 1; i >= 0; i--);
            i++;
        }
        fprintf(out, NULL, NULL);
    }
}

```

```
        return out;

    }
    else
    {
        printf("No hay memoria suficiente");
    }
}
```

FUNCIONAMIENTO DEL CÓDIGO:

Una vez que el usuario que le de a depurar le saldrá la siguiente pantalla para que elija la opción que necesite:

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

Si el usuario quiere traducir un texto de ASCII a binario o viceversa pulsaría el numero "2", donde le aparecería otra opción en la que tendrá que elegir si quiere traducir de binario a texto(ASCII) o de texto a binario.

```
-----  
Opciones:  
1.Morse  
2.Binario  
3.Notas  
4.Salir  
-----  
Ingresa opcion: 2  
  
-----  
a. binario a texto  
b. texto a binario  
-----  
Ingresa opcion: █
```

Si el usuario quiere traducir de binario a texto tendría que pulsar "a", donde a continuación le pedirá la longitud y la frase que quiera escribir

```
-----  
a. binario a texto  
b. texto a binario  
-----  
Ingresa opcion: a  
Introduce la longitud de la frase  
4  
Introduce una frase formados por 0 o 1:  
1101000110111111011001100001  
hola  
-----
```

Y en caso contrario tendría que pulsar “b”, después escriben la longitud de la frase y a continuación la frase:

```
-----
a. binario a texto
b. texto a binario
-----
Ingresa opcion: b
Introduce la longitud de la frase
10
Escribe un texto u oracion con el lenguaje Ascii
hola mundo
01101000
01101111
01101100
01100001
00100000
01101101
01110101
01101110
01100100
01101111
-----
```

Por último tras realizar la opción que quiere vuelve a aparecer en menú

```
-----
Ingresa opcion: b
Introduce la longitud de la frase
10
Escribe un texto u oracion con el lenguaje Ascii
hola mundo
01101000
01101111
01101100
01100001
00100000
01101101
01110101
01101110
01100100
01101111
-----
Opciones:
1.Morse
2.Binario
3.Notas
4.Salir
-----
Ingresa opcion: _
```

y si ya no quieres hacer nada pulsa “4” para salir.

```
-----  
Opciones:  
1.Morse  
2.Binario  
3.Notas  
4.Salir  
-----  
Ingresa opcion: 4  
  
C:\Users\irene\source\repos\Project9\Debug\Project9.exe (proceso 7060) se cerró con el código 0.  
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->  
Cerrar la consola automáticamente al detenerse la depuración.  
Presione cualquier tecla para cerrar esta ventana. . .
```