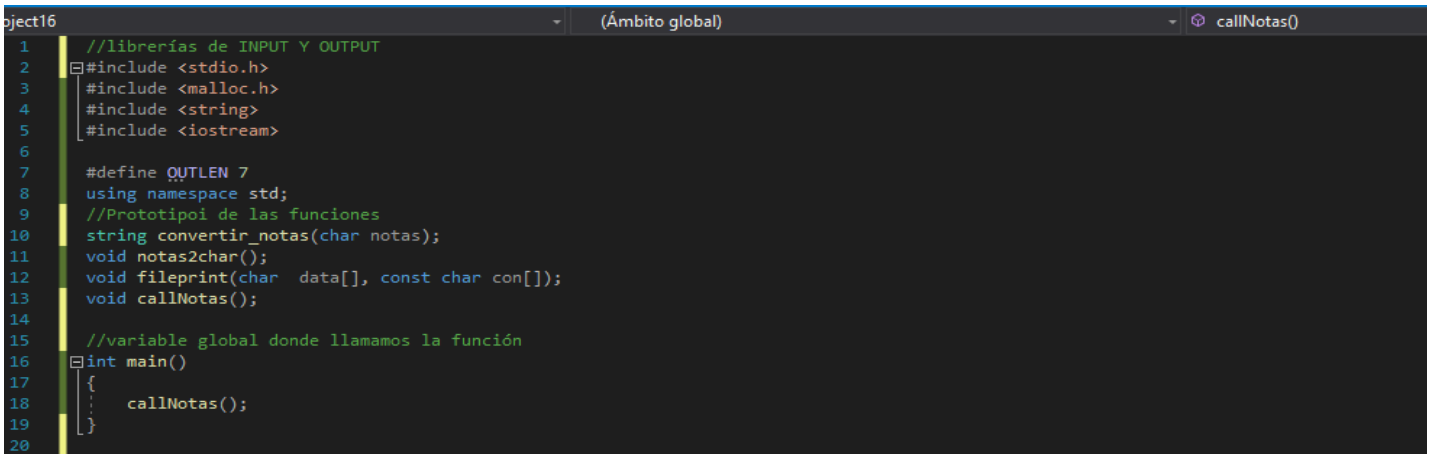
## CÓDIGO NOTAS MUSICALES

Para la reproducción de notas musicales, se crea un algoritmo tal que traduce e imprime caracteres considerados como notas a código binario  y viceversa. Responde tanto a consola como a ficheros.
Si se decide pasar de notas a binario recorremos el input cuyos caracteres equivaldrán al valor de 1 en el conjunto de siete cifras. Y pasa traducir de binario a notas, se identifica la posición del 1, la extraemos con el operador módulo y sustituimos por la nota correspondiente.

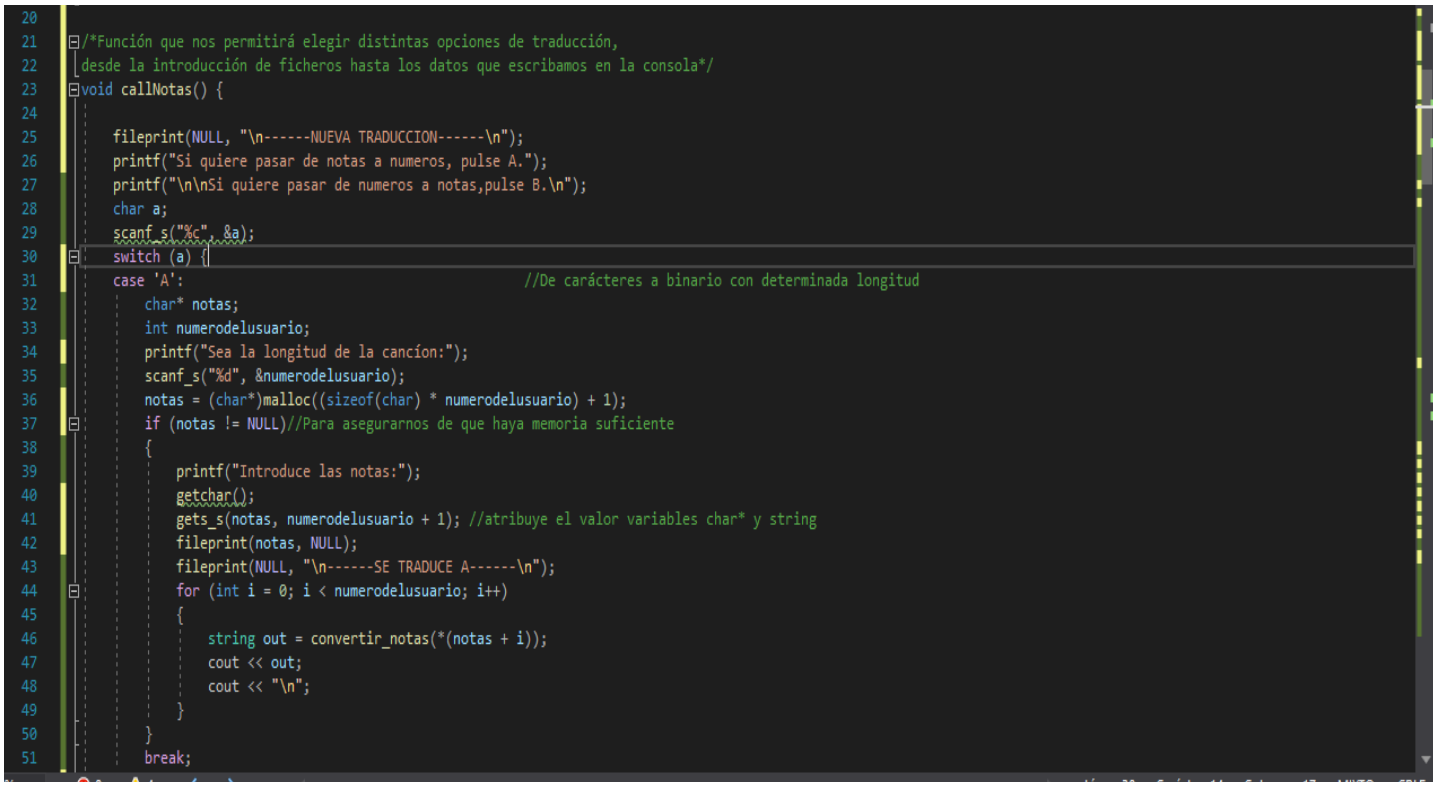
## BASE

Se colocan las librerías de input y output necesarias (para variable string,declaraciones de funciones estándar,etc).
El código está compuesto principalmente por cuatro funciones. El primero es “callNotas()”, que la identificamos dentro de la variable global (int main) y contendrá al resto de funciones(por eso es de tipo void).

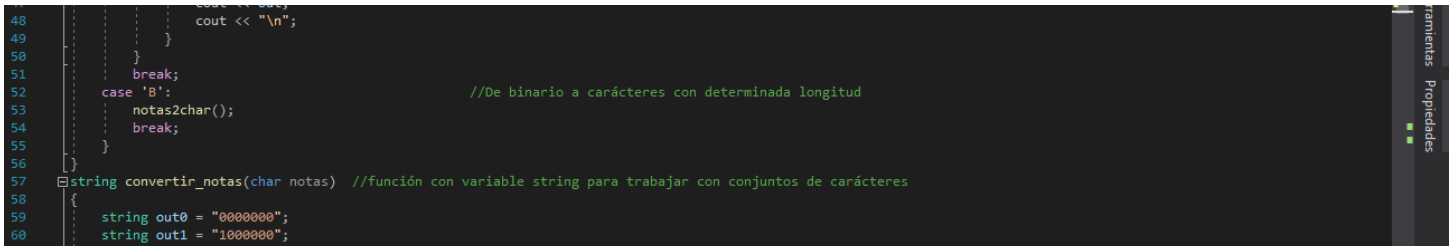


## FUNCIÓN CALL NOTAS

El cuerpo de la función “callNotas” está compuesto por la llamada función “fileprint” para ficheros (en caso de que se introduzcan).Elegimos la longitud de la canción(numero de notas) y utilizamos  condicional “switch-case” para elegir entre dos opciones. La primera es pasar de notas(de variable char) a binario(en el caso A).  Y la segunda es pasar pasar de binario a notas (en el caso B).Todo ello se consigue con la introducción de funciones desarrolladas.



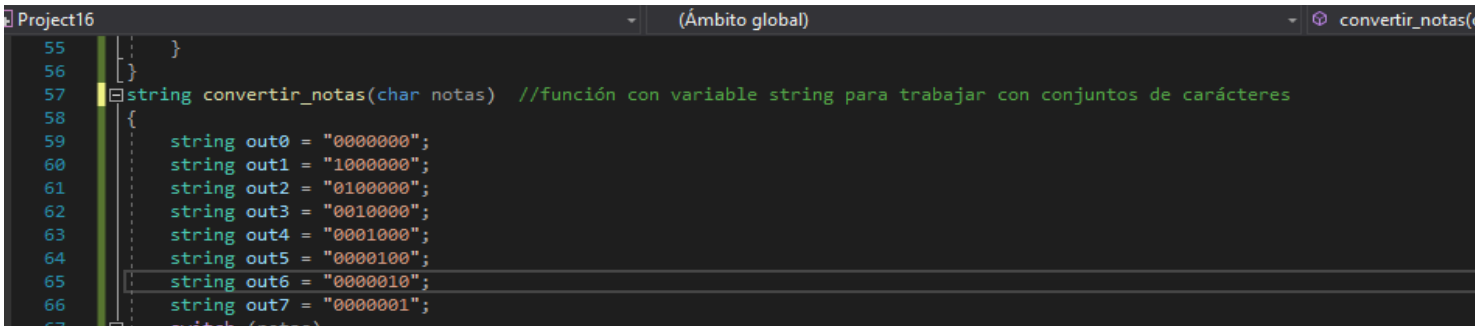
```
/*Función que nos permitirá elegir distintas opciones de traducción,
desde la introducción de ficheros hasta los datos que escribamos en la consola*/
void callNotas() {

    fileprint(NULL, "\n------NUEVA TRADUCCION------\n");
    printf("Si quiere pasar de notas a numeros, pulse A.");
    printf("\n\nSi quiere pasar de numeros a notas,pulse B.\n");
    char a;
    scanf_s("%c", &a);
    switch (a) {
    case 'A':                                       //De carácteres a binario con determinada longitud
        char* notas;
        int numerodelusuario;
        printf("Sea la longitud de la cancíon:");
        scanf_s("%d", &numerodelusuario);
        notas = (char*)malloc((sizeof(char) * numerodelusuario) + 1);
        if (notas != NULL)//Para asegurarnos de que haya memoria suficiente
        {
            printf("Introduce las notas:");
            getchar();
            gets_s(notas, numerodelusuario + 1); //atribuye el valor variables char* y string
            fileprint(notas, NULL);
            fileprint(NULL, "\n------SE TRADUCE A------\n");
            for (int i = 0; i < numerodelusuario; i++)
            {
                string out = convertir_notas(*(notas + i));
                cout << out;
                cout << "\n";
            }
        }
        break;
```

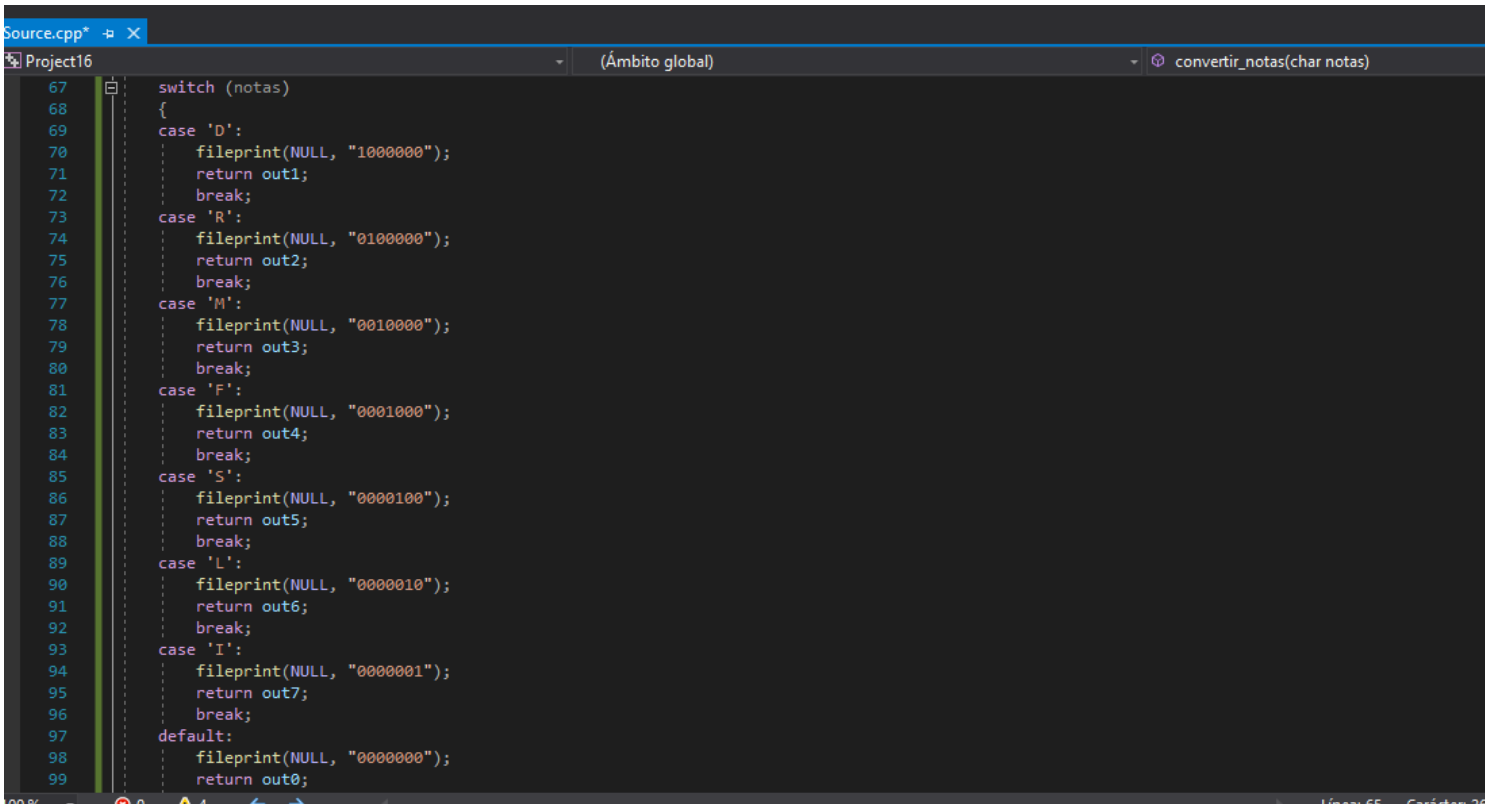


```
                cout << "\n";
            }
        }
        break;
    case 'B':                                       //De binario a carácteres con determinada longitud
        notas2char();
        break;
    }
}
string convertir_notas(char notas)  //función con variable string para trabajar con conjuntos de carácteres
{
    string out0 = "0000000";
    string out1 = "1000000";
```

## FUNCIÓN CONVERTIR NOTAS

El cuerpo de la función “convertir_notas” está compuesto por variables de tipo string para trabajar con conjuntos de caracteres(conjunto de cifras en binario). Estas variables son asignadas a cada nota(do=D,re=R,etc.) identificada por la condicional switch, cuyo resultado es ser impresa



## FUNCIÓN NOTAS2CHAR

El principal objetivo de la función “notas2char” es identificar la posición del 1 en cada binario, la extraemos y sustituimos por la nota correspondida según el valor que nos devuelva.

urce.cpp*

Project16 | (Ámbito global) | convertir_notas(char notas)

```
97      default:
98          fileprint(NULL, "0000000");
99          return out0;
100         break;
101     }
102 }
103 void notas2char()//función con void que identifica cada nota según la posición de la nota 1
104 {
105     int len = 0, i = 0;
106     char* frase;
107     printf("\nIntroduce la longitud de tu canción\n");
108     scanf_s("%d", &len);
109     frase = (char*)malloc((len * 7) + 1);
110     printf("\nIntroduce una frase formada por 0s y 1s\n");
111     if (frase != NULL)
112     {
113         getchar();
114         gets_s(frase, ((len * 7) + 1));
115         fileprint(frase, NULL);
116         fileprint(NULL, "\n------SE TRADUCE A------\n");
117     }
118     while (*(frase + i) != '\0')
119     {
120         if (*(frase + i) == '1')
121         {
122             switch (i % 7)
123             {
124             default:
125                 break;
126             case 0:
127                 printf("DO ");
128                 fileprint(NULL, "DO ");
129                 break;
```

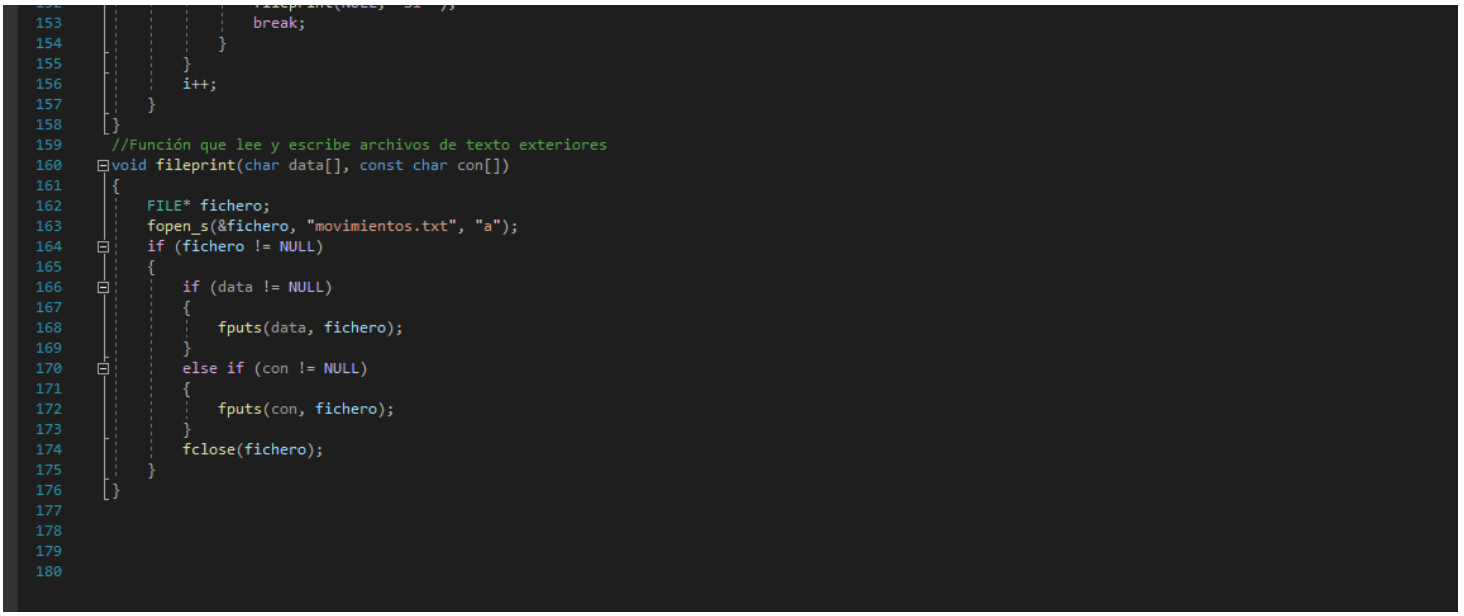
0 | 4 | Línea: 69 | Carácter: 11

Project16 | (Ámbito global) | convertir_notas(char notas)

```
127                 printf("DO ");
128                 fileprint(NULL, "DO ");
129                 break;
130             case 1:
131                 printf("re ");
132                 fileprint(NULL, "RE ");
133                 break;
134             case 2:
135                 printf("mi ");
136                 fileprint(NULL, "MI ");
137                 break;
138             case 3:
139                 printf("fa ");
140                 fileprint(NULL, "FA ");
141                 break;
142             case 4:
143                 printf("sol ");
144                 fileprint(NULL, "SOL ");
145                 break;
146             case 5:
147                 printf("la ");
148                 fileprint(NULL, "LA ");
149                 break;
150             case 6:
151                 printf("si ");
152                 fileprint(NULL, "SI ");
153                 break;
154             }
155         }
156         i++;
157     }
158 }
159 //Función que lee y escribe archivos de texto exteriores
```

## FUNCIÓN FILEPRINT

La función fileprint consiste en identificar, abrir, leer los datos y cerrar los ficheros recibidos exteriores.



```
153                 break;
154             }
155         }
156         i++;
157     }
158 }
159  //Función que lee y escribe archivos de texto exteriores
160 void fileprint(char data[], const char con[])
161 {
162     FILE* fichero;
163     fopen_s(&fichero, "movimientos.txt", "a");
164     if (fichero != NULL)
165     {
166         if (data != NULL)
167         {
168             fputs(data, fichero);
169         }
170         else if (con != NULL)
171         {
172             fputs(con, fichero);
173         }
174         fclose(fichero);
175     }
176 }
177
178
179
180
```