Instituto Politécnico Nacional.

Escuela Superior de Cómputo

ESCOM

Profesor: Joel Omar Juárez Gambino

Alumno:

Torres Abonce Luis Miguel

Grupo:3CM3

Materia: Análisis y diseño de algoritmos

Tema: Codificación de mensaje

Fecha: 29 de Noviembre 2021

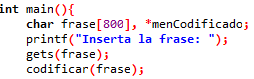
Introducción.

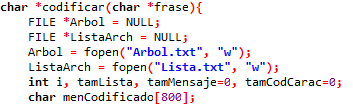
El problema a resolver consiste en realizar un programa que codifique y decodifique (cada una en una función) un mensaje dado por el usuario, esto utilizando el algoritmo de Huffman además de utilizar la estrategia ávida.

El algoritmo de Huffman consiste en crear un árbol binario, todo esto con el fin de disminuir el tamaño de los datos.

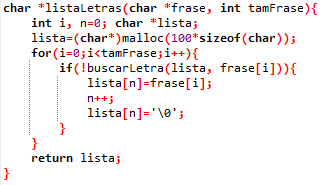
Este código lo realice en el lenguaje c++ ya que lo he ocupado bastante, además utilizó archivos para guardar el árbol y la lista de las letras que se utilizan en el mensaje a codificar.

Desarrollo.

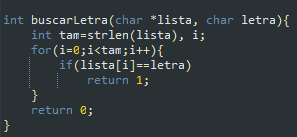
En el main se crea una cadena de caracteres para guardar el mensaje y se pregunta el mensaje a codificar. Se llama la función codificar y se manda el mensaje.

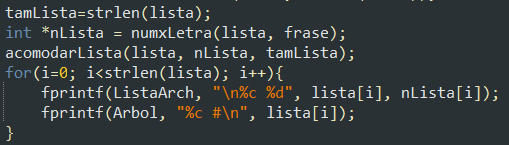
En la función codificar se recibe el apuntador al mensaje y se declaran las variables que abrirán y cierran el archivo que se utilizaran, además de declarar más variables a utilizar.

 Se utiliza un bucle for que recorre toda la cadena y si encuentra un espacio lo sustituye por el signo $ y se manda llamar la función lista letras y se envían la variable frase con su respectivo tamaño.

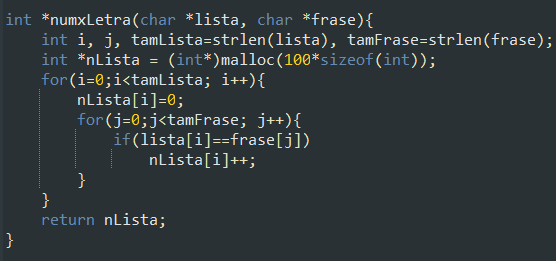
En esta función se le asigna la memoria a la variable lista de tamaño 100\*tamaño de un char

Y se utiliza un bucle que recorre toda la frase y si se cumple la función buscar letra entonces la letra que se esta en la frase se copia a la lista se le suma 1 y en ese lugar de la lista se le iguala “\0” que significa que es el fin del archivo. Se retorna la lista de las palabras.

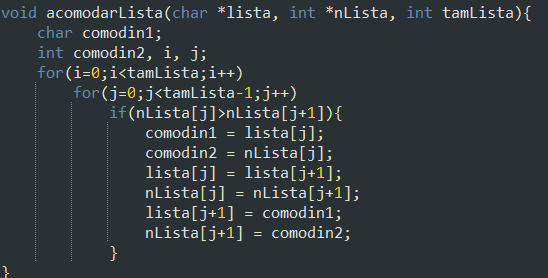
En esta función se utiliza un bucle que recorre la lista recibida y si la letra es igual a la letra que se recibió significa que en la lista ya tiene esa letra y se regresa un 1 (TRUE), de lo contrario significa que esta letra no se encontraba en la lista de palabras por lo que se regresa un 0 (FALSE).



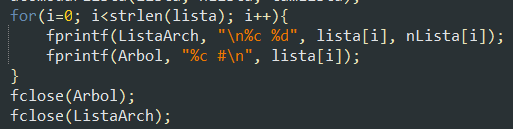
Posteriormente ya que sale de la función lista de letras, ya tenemos en la variable la lista de estas letras sin repetir. Posteriormente se declara el apuntador nlista que será igual al entero que regrese la función numxLetra y se mandan las respetivas variables a utilizar. Se manda llamar la función acomodar lista y se envían las variables que requiere en este caso la lista, nlista, y el tamaño de esta.



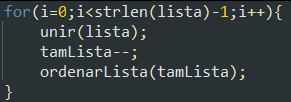
En esta función se reciben las variables apuntador lista y apuntador frase, se declaran variables, el apuntador n lista aparte el tamaño 100\*tamaño de un entero, y se utilizan 2 ciclos anidados en el primer for nLista se iguala a 0 para que se reinicie cada vez que se termine ese for, en el segundo for se verifica si la letra en la lista en el lugar del primer for es igual a la letra de la frase en el lugar del segundo for entonces a nLista se en el lugar del primer for se le suma 1 de esta forma sabemos el número de letras que hay en cada una.

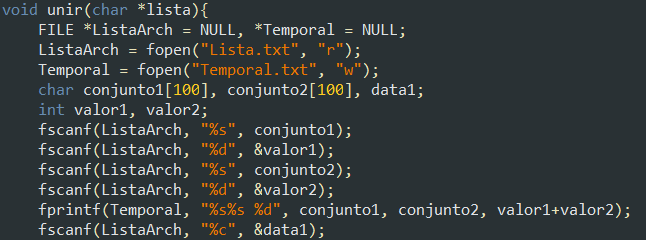


Se declaran las variables a utilizar y se anidan 2 for con las variables i, j respectivamente, y se verifica si el nLista en j es mayor a este mismo en la siguiente posición si esto se cumple se cambia de lugar el de atrás pasa a la siguiente posición y el que estaba siguiente a la anterior mediante el método burbuja.

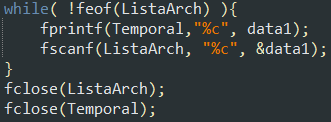


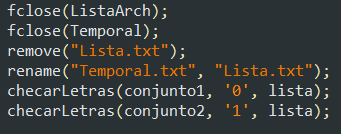
Después de regreso en la función codificación se tiene un ciclo for que escribe en el archivo lista y árbol y se imprimen los caracteres correspondientes y se da un salto de línea, en la lista se imprime el carácter y la cantidad de veces que este se repite y en el árbol la letra y el valor en bits de estos. Y se cierran los archivos.

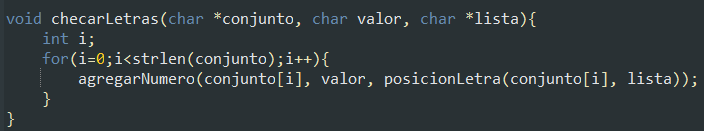
Posteriormente en un ciclo for hasta el tamaño de la lista-1 y se con la función unir se manda la variable lista, se le resta 1 al tamaño de la lista y se llama a la función ordenar lista con el tamaño de esta.



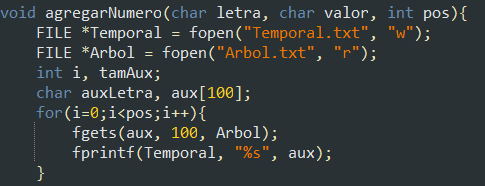
En la función unir se declaran las variables que abrirán los archivos, y se abren estos, después se declaran 2 cadenas(conjunto 1 y 2) de carácter de 100 y un carácter data1, después se escanea del archivo lista la cadena y se guarda en la cadena 1 y de igual forma el número (es el número de veces que se repite el carácter), y de igual forma con el siguiente carácter y número pero esta vez en el conjunto 2 y valor 2 y estos 2 datos ,además de la suma de valor 1 + valor 2, se guardan en el archivo temporal, después en el archivo lista se guarda el carácter en la variable data.

Después en la misma función se utiliza la función feof que regresa verdadero o falso si la variable que se manda es el final del archivo todo esto dentro de un ciclo, se imprime en el archivo temporal carácter por carácter lo que contiene la variable data1 y después se guarda en la variable data 1 el carácter (que se está recorriendo) en el archivo lista. Finalmente se cierran los 2 archivos.

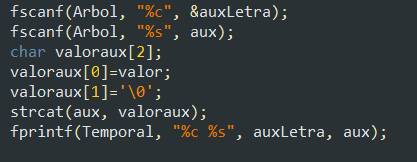
Por último, en esta función se borra el archivo lista y al archivo temporal se le cambia el nombre a lista y se llama a la función checar letras.

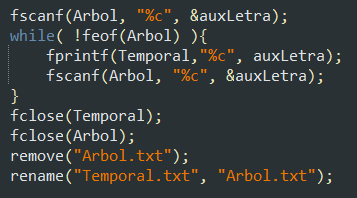


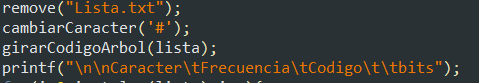
Esta función recibe 2 apuntadores de cadenas de caracteres y un carácter. En esta función solo se utiliza un ciclo for que recorre el arreglo conjunto, después solos se utiliza la función agregar función



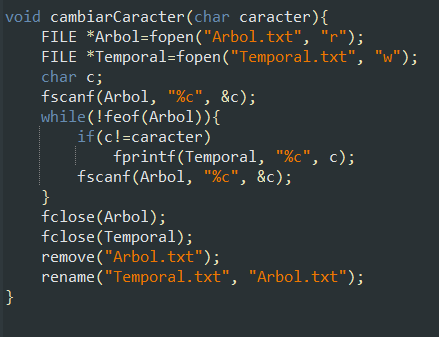
En esta función se declaran las variables que abrirán el archivo y los abren, después en un ciclo for se obtiene del archivo árbol los 100 caracteres y se guardan en auxiliar y después en el archivo temporal se pega los 100 caracteres copiados anteriormente.

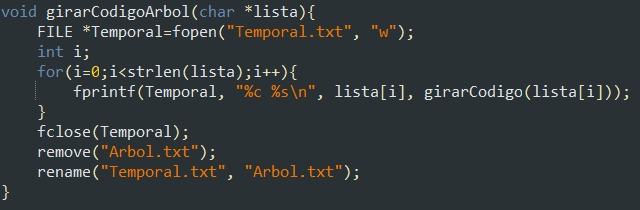
después del archivo árbol se guardan los caracteres en aux letra y la cadena de caracteres en aux, después se declara una variable valoraux de tamaño 2 y en el lugar 0 se guarda la variable valor y en el lugar 1 \0 (final del documento) después se utiliza la función strct para que a aux se le concatene el valoraux y se imprime en el documento temporal aux letra(caracter) y aux (cadena).

Por ultimo se guarda en aux letra el carácter siguiente del archivo árbol. Y mientras no sea el final de la cadena se imprime en el documento temporal aux letra (carácter) y se guarda e esta misma el carácter del archivo árbol básicamente copia el archivo árbol a temporal se cierran los archivos y renombramos al archivo árbol a temporal.

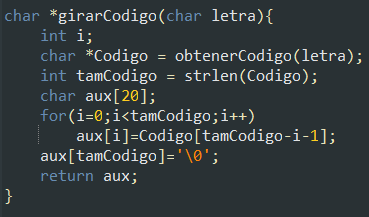


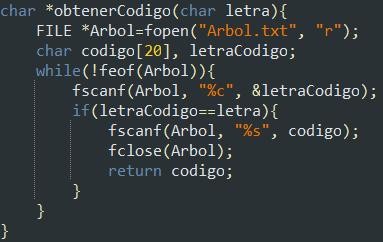
Después de nuevo en la función codificar se borra el archivo lista y se manda ‘#’ a la función cambiar carácter, y se manda lista a la función girar código árbol y se imprime los datos solicitados en la salida (carácter, frecuencia, código y bits).

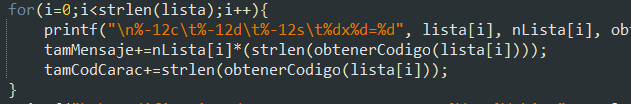
En esta función se abren 2 archivos árbol y temporal y se tiene un carácter c; se guarda en esta variable el primer carácter del archivo árbol y mientras no sea el final del archivo árbol se verificará si c es diferente del carácter recibido en la función entonces en al archivo temporal se imprimirá esta variable (c), después en esta misma variable se guarda el siguiente carácter del archivo árbol, finalmente cerramos los archivos y se borra el archivo árbol y se renombra el archivo temporal por árbol.



En esta función se abre el archivo temporal en una variable con el mismo nombre, después con un ciclo se imprime en este archivo el carácter de la lista en el lugar de i (va recorriendo el arreglo) y la cadena que regresa la función girar código, finalmente se cierra el archivo y se borra el archivo árbol y se le cambia el nombre de este archivo a “árbol”.

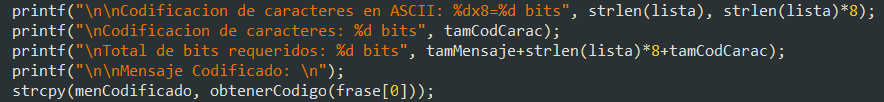
Esta función crea un apuntador código que es lo que regresa la función obtener código, y un entero del tamaño de código, y con un ciclo que recorrerá la cadena código, y el aux en i tendrá lo que tiene la cadena código en su tamaño -1-i y en el último lugar tendrá “\0” (final del archivo) y se regresa aux.

Esta función abre el archivo árbol y mientras el archivo no termine en letra código se guarda el carácter del archivo y si letra código es igual a la letra que recibe la función entonces se guarda en código la cadena del archivo y se cierra el archivo y se retorna esta.

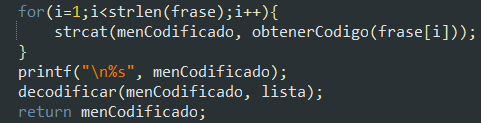


Siguiendo en la función codificar en un ciclo for se imprime en consola todos los datos que se piden de salida

Y se obtiene el tamensaje y el tamaño de codCarac con la función obtener código explicada anteriormente.



Después se imprime en consola los datos que se piden, además del mensaje codificado que está en una función separada.

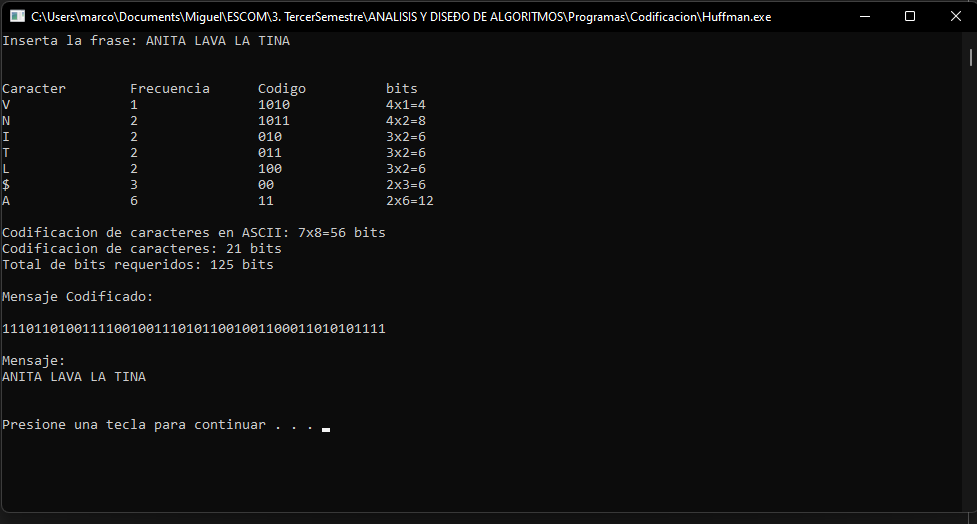


En un ciclo que recorre el mensaje se va concatenando en mencodifacado, y se imprime este mensaje ya decodificado con la función decodificar.

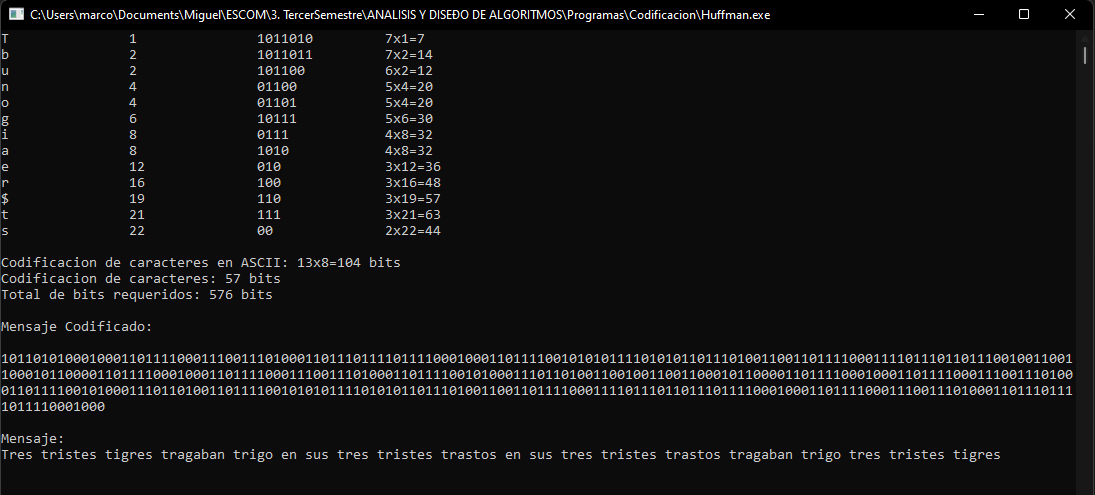


Por ultimo para decodificar el mensaje se declara una cadena de 20, y en el posición 0 tendrá la variable de mensaje codificado en la misma posición y en la siguiente posición tendrá ‘\0’ para saber que es el fin del archivo, después en un ciclo recorrerá el mensaje codificado en conjunción con el carácter c que dependiendo de la situación que se pueda dar y esta se agregara a código en su siguiente posición y en la siguiente posición ‘\0’, además de cambiar el carácter $ en espacio como anteriormente lo hice para obtener el mensaje correcto.

Prueba 1:



Prueba 2:



Prueba 3:

