Assignment 5

Design Document

Purpose

Create a program that will encode a given message using Hamming(8, 4) codes into encoded data, and create a corresponding program to decode the encoded data back into a message.

Layout/Structure and Pseudocode for a Clear Description/Explanation of How Each Part of the Program Should Work:

encode.c

```
parse command-line arguments
identify infile and outfile
create generator matrix
for every character (byte) read from infile …
        separate the upper and lower nibbles
        generate ham code for upper and lower nibble
        write lower nibble to outfile
        write upper nibble to outfile
free memory and close files
```

decode.c

```
parse command-line arguments
identify infile and outfile
enable verbose printing if specified
create H transposed matrix
for every character (byte) read from infile …
        get a second byte from infile
        decode first byte to get lower nibble
        decode second byte to get upper nibble
        combine upper and lower nibble
        write resulting character to outfile
print statistics
free memory and close files
```

ham_encode

```
Convert message to 4x1 bit matrix
Multiply msg as bm with G to get code as bit matrix
Convert code as bm to encoded data
Free the memory
Return the encoded data
```

ham_decode

```
Create lookup table
Convert code to 8x1 matrix
Multiply code bit matrix and H to get error syndrome bit matrix
Convert resulting error syndrome bit matrix to data
Free the memory
Get value of lookup table at error syndrome
If HAM_OK then put lower nibble of encoded data into message and return HAM_OK
If HAM_ERR then there is nothing to be done so simply return HAM_ERR
If any other value is fetched then correct the bit at that value and return HAM_CORRECT
```