

Description:

This program will perform functions for a Hamming[7,4] or Hamming Code [15,11]. It will encode a 4 bit or 11 bit data string with the necessary parity bits, check an existing code for proper parity and correct it if it is incorrect, and decode a 7 bit or 15 bit code to get the original message.

Libraries:

#include <iostream> - used for console input and output.
#include <string> - used for strings which handles input and output.
#include <vector> - used for vectors which function as arrays.
#include <bitset> - used to convert a string to a binary number.

Structure:

The program is separated into various functions. Main is the beginning point which then branches to either the h74 or h1511 function. From there either encode74, encode1511, parity74, parity1511, decode 74, or decode1511 is called and passed a string containing a string representation of a bit string depending on user input. Within each of these functions, a function to multiply matrices is called and the results are outputted to the screen. The process then begins again from main.

Functions:

Int main(int argc, char** argv) – main function that handles initial user input.
Void h74() - handles calling appropriate encode, parity, or decode function.
Void h1511() - handles calling appropriate encode, parity, or decode function.
Void encode74(string last) – multiplies given message by a matrix generator matrix and outputs the results to the screen.
Void parity74(string last) – multiplies given code by a parity check matrix. The resulting syndrome vector is printed to the screen. If the syndrome vector is a 3 bit zero vector, then the code is correct and nothing else is done. If it is not a 3 bit syndrome vector, the the code is corrected and printed to the screen.
Void decode74(string last) – multiplies the given code by a decoding matrix and prints the original message to the screen.
Void encode1511(string last) – same as encode74 just with an 11 bit message.
Void parity1511(string last) – same as parity74, but with a 15 bit code.
Void decode1511(string last) – same as decode74, but with a 15 bit code.
Void matrix_mult(vector<vector<int>> farr, vector<vector<int>> sarr, vector<vector<int>> &oarr) – multiplies two matrices and stores the resulting matrix in oarr. The matrices must be appropriate sizes (ex. (4x3) times a (3x4)) to be multiplied.

Compilation and Usage:

A Makefile has been created for this program. Typing make at the command line will compile the program and create the executable named “Hamming”. “./Hamming” will run the program. If manual compilation is necessary, using the following command will accomplish the same results minus the object files:

```
g++ -std=c++11 Hamming.cpp functions.cpp -o Hamming
```

Testing:

The program was tested thoroughly and every function is working and returns appropriate results. The following test cases were used:

H74:

encode 1011: returns (0,1,1,0,0,1,1)
parity 0110011: returns (0,0,0)
parity 0110111: returns (1,0,1) then (0,1,1,0,0,1,1)
decode 0110011: returns (1,0,1,1)

H1511:

encode 01010101010: returns (0,1,0,0,1,0,1,1,0,1,0,1,0,1,0)
parity 010101010101010: returns (0,0,0,0)
parity 010101010101011: returns (1,1,1,1) then (0,1,0,1,0,1,0,1,0,1,0,1,0,1,0)
decode 010010110101010: returns (0,1,0,1,0,1,0,1,0,1,0)

Invalid Input:

blank line: re-prompts
random letters: re-prompts
encode abc: tells user can only be 0s or 1s and re-prompts
decode 11: tells user that bit string is not the appropriate size

Submission:

Hamming.cpp
functions.cpp
Hamming.h
Makefile
prog1.pdf