## Homework #1 Hamming Code

# M001

**Author**
Lyndon Engel

**Date**
10/2/2016

**Professor:**
Christer H. Karlsson

**Course:**
CSC 317

**Location:**
McLaury - Room 304

# Program Information

This program can create, parity-check, error correct and decode a message using Hamming(7,4) or Hamming(15,11). It does this by using three Hamming matrices (G, H and R) for Hamming(7,4) and three different hamming matrices (G1511, H1511, and R1511) for Hamming(15,11). The program will first ask the user what Hamming code to be used (or if the user wants to exit). The next question it will ask is whether the user wants to create, parity check (with error correction) or decode a message. The user will then provide either the data or the message, the program will calculate either the message or the data. All messages and data will be entered as a vector of binary digits. Example, four bit data 1011, is a seven. Fifteen bit message, will have eleven data bits and four parity bits and look something like 100100111011110.

# Compiling and Usage

**Compiling Instructions:**

Typing Hamming will start the program. After which a prompt will appear telling you the first basic instructions.Those being H74 to do hamming (7,4) functions or H1511 to do hamming (15,11) functions. There will also be an option to exit.

**Usage:**

```
>>./Hamming
. . .
>>encode 1011
x = <0, 1, 1, 0, 0, 1, 1>
. . .
>>parity 0100011
z = <0, 0, 0>
. . .
>>parity 0100111
z = <1, 0, 1>
x = <0, 1, 1, 0, 0, 1, 1>
. . .
>>decode 0110011
r = <1, 0, 1, 1>
```

# Todo, Bugs, and Modifications

**Bug:**

- No known bugs

**Todo:**

none

**Modifications and Development Timeline:**

| Date | Modification |
|---|---|
| Sep 23, 2016 | General usability |
| Sep 25, 2016 | Hamming (7,4) encode, parity |
| Sep 28, 2016 | Hamming (7,4) decode, and parity error handling |
| Sep 30, 2016 | Hamming (15,11) encode, parity |
| Oct 1, 2016 | finishing up Hamming (15,11) |
| Oct 3, 2016 | error handling |
| Oct 5, 2016 | Documentation |

# functs.cpp File Reference

```
#include <iostream>
#include <vector>
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include "functdef.h"
```

## Functions

| | |
|---|---|
| void | **ham74main** () |
| vector< int > | **encode** (char data[]) |
| void | **parity** (char data[]) |
| vector< int > | **decode** (char data[]) |
| void | **ham15main** () |
| vector< int > | **encode15** (char data[]) |
| void | **parity15** (char data[]) |
| vector< int > | **decode15** (char data[]) |
| int | **ConvertToBinary** (int n) |
| bool | **errorlength** (vector< int > data, int ptype, string command) |
| bool | **numerror** (vector< int > data) |

### vector<int> decode15 ( char data[])

**Author**

Lyndon Engel

**Description:**

This function decodes the users inputed message using hamming (15,11) and then returns a vector with the incoded messages.

**Parameters**

`[in]` **data** - reads in charcter array holding users message

**Returns**

r - vector holding decoded message

Definition at line 511 of file functs.cpp.

### vector<int> decode ( char data[])

**Author**

Lyndon Engel

**Description:**

This function decodes the users inputed message using hamming (7,4) and then returns a vector with the incoded messages.

**Parameters**

`[in]` **data** - reads in charcter array holding users message

**Returns**

r - vector holding decoded message

Definition at line 260 of file functs.cpp.

### int ConvertToBinary ( int n)

**Author**

Lyndon Engel

**Description:**

This function takes in integers and makes sure they are 1's and 0's. It does this by modding the numbers by 2.

**Parameters**

`[in]` **n** - the bit that needs to be turned into 1 or 0.

**Returns**

i - the new number

0 - if error

Definition at line 552 of file functs.cpp.

### vector<int> encode ( char data[])

**Author**

Lyndon Engel

**Description:**

This function encodes the users inputed message using hamming (7,4) and then returns a vector with the incoded messages.

**Parameters**

`[in]` **data** - reads in charcter array holding users message

**Returns**

x - vector holding encoded message

Definition at line 124 of file functs.cpp.

### vector<int> encode15 ( char data[])

**Author**

Lyndon Engel

**Description:**

This function encodes the users inputed message using hamming (15,11) and then returns a vector with the incoded messages.

**Parameters**

[in] **data** - reads in charcter array holding users message

**Returns**

x - vector holding encoded message

Definition at line 386 of file functs.cpp.

### bool errorlength ( vector< int > data, int ptype, string command )

**Author**

Lyndon Engel

**Description:**

This function makes sure the length of the users inputted message is correct for the command they are useing. If it is correct it returns true otherwise it outputs and error message and returns false.

**Parameters**

[in] **data** - reads in vector holding users message

[in] **ptype** - what type of hamming code will be done

[in] **command** - what command the user wants to use

**Returns**

false - if the length is incorrect

true - if the length is correct

Definition at line 580 of file functs.cpp.

### void ham15main ( )

**Author**

Lyndon Engel

**Description:**

This function handles everything to do with the hamming (15,11) opperations. If the user specifys H74 then it will come to this function where they can then designate whether they want to encode, parity check, or decode a message they enter.

Definition at line **301** of file **functs.cpp**.

### void ham74main ( )

**Author**

Lyndon Engel

**Description:**

This function handles everything to do with the hamming (7,4) opperations. If the user specifys H74 then it will come to this function where they can then designate whether they want to encode, parity check, or decode a message they enter.

Definition at line **35** of file **functs.cpp**.

### bool numerror ( vector< int > data)

**Author**

Lyndon Engel

**Description:**

This function checks to make sure that all the numbers in the users message are 1's and 0's. If they arent the function returns false.

**Parameters**

`[in]` **data** - reads in vector holding users message

**Returns**

false - if the message is incorrect

true - if the message is correct

Definition at line **618** of file **functs.cpp**.

### void parity ( char data[])

**Author**

Lyndon Engel

**Description:**

This function parity checks the users already encoded message. If there isnt and error in the message it outputs vector with all 0's however if there is and error the function then it flips the wrong bit to make it correct and then will output the correct encoded message.

**Parameters**

[in] **data** - reads in charcter array holding users message

Definition at line **166** of file **functs.cpp**.

### void parity15 ( char data[])

**Author**

Lyndon Engel

**Description:**

This function parity checks the users already encoded (15,11) message. If there isnt and error in the message it outputs vector with all 0's however if there is and error the function then it flips the wrong bit to make it correct and then will output the correct encoded message.

**Parameters**

[in] **data** - reads in charcter array holding users message

Definition at line **427** of file **functs.cpp**.