# CSE13S Winter 2021 Assignment 4: Hamming Codes Design Document

## **PRELAB**

1. Calculate Hamming codes for  $0000_2$ –1111 $_2$  using the generator matrix. Show your work. Hint: Convert your codes to hex.

Decimal	Binary	Hamming Code
0	0000	0000 0000
1	0001	1000 0111
2	0010	0100 1011
3	0011	1100 1100
4	0100	0010 1101
5	0101	1010 1010
6	0110	0110 0110
7	0111	1110 0001
8	1000	0001 1110
9	1001	1001 1001
10	1010	0101 0101
11	1011	1101 0010
12	1100	0011 0011
13	1101	1011 0100
14	1110	0111 1000
15	1111	1111 1111

- 2. Decode the following codes. If it contains an error, explain how you can correct it; however, some errors cannot be corrected.
  - (a) 1110 0011<sub>2</sub>
  - (b) 1101 1000<sub>2</sub>
  - 2a.  $0001_2$  or 1 in decimal. There was an error in the 2nd element as e arrow = {1011}.
  - 2b. There is more than one error as e arrow = 0101 and cannot be corrected with my current understanding.
- 3. Complete the rest of the look-up table shown below.

0 0

14

... ...

15 HAM\_ERR

Decimal	Binary	Look-Up
0	0000	0
1	0001	4
2	0010	5
3	0011	HAM_ERR
4	0100	6
5	0101	HAM_ERR
6	0110	HAM_ERR
7	0111	3
8	1000	7
9	1001	HAM_ERR
10	1010	HAM_ERR
11	1011	2
12	1100	HAM_ERR
13	1101	1
14	1110	0
15	1111	HAM_ERR

## WORK (Correction for Work #2)

Ci	2/4/21 C8E 135		
	Prelab a	restinas	
	1. Calculat	e Hammina Co.	des company
	index. ma	itrix. Show	dc) for 00002-11112 with generator
	Decimal	Binary	Hamming lode
Production of the Control of the Con	0	0000	
		0001	1000 0000
-	2	0010	and the same of th
	3	0011	1100 1100
	4	0100	0.01.0
The state of the s	5	0101	1010 1010
=	6	0110	0110 0110
	7	0111	1110 000 1
	8	1000	00011110
	9	1001	10011001
	10	1016	01010101
	11	1011	1101 0010
	12	1100	0011 0011
	13	1101	1011 0100
	14	(110	0111 1000
	15	1111	
	2a.) 01	112, 7,	there was
	element (	e = 0010	there was an error in the 71
3			
7	2'b.) There	is more to	nan one
			nan one error (e = 1010).

William Sanksu 814121 CSE135

	CSE135			).
	Prelab Questions	and the control of th	es umanadorina artista como e umano cambios. Por disposição como esta esta esta esta esta esta esta esta	
Substitution of September 1971 Septe	3.) Decimal	Look-Up	gen i nin til i til til til steller skill de kladenskrikensen i kladisekkrikensen skildisekkrikensen.	
CALL TOWNS OF THE SAME OF THE SAME OF THE SAME	and the second s		So The Took School of the School of September 497 David School	WALLES BOOK - THE COMMISSION OF
officer from the first constant \$1.00 has \$40 minutes \$1.00	O	0	A company of the second	- No. and Add to Company of the Comp
	1	4	and the second transfer of the second transfe	and the second s
	2	5	the second secon	
	3	HAM - ERR		
	4	6	a lamas a	and the second s
	5	HAM - ERR		
	6	HAM-ERR		
	7	3	8 6	
	8	7		- Commence
	9	HAM-ERR	*	
	(0	HAM - ERR	KI	
	11	L HAM-ERR		-
	12	FIANV-EKK		Company of the Association Statement
	13			The second second second second
			1	The control was displaced
	10	HAM EICH		personal and the second
			- A Company of Market	Japanyson glove in Japan Sone is visited in Person Pediatric #1
-	8		entered to see 8	
				and the second s
20002				Security of the Control of the Contr
North Assessment Co.			general desired and the second	
				-
			and the same of th	
		6 HAM-ERR	The second of th	

	WITHOUM Santosa 2/4/21 CSE135
2	Work Q#1
	Deco Convert to Hamming Code
	Let 6 = (1 0 0 0 0 1 1 1 1 ) 0 1 0 0 1 0 1 1 0 1 0 0 0 0 1 1 1 0 0
	0.) 0000
	{0000} · 6 °/. 2 = ₹
•	{00000000}/.2=
	{000000 ₹ =
	1.)0001
	~ 2 1 0 0 0 3 · 6 % 2 = 己
	₹ 10000 111 \$ % X=
	\$1000 611 1 <b>3</b> =

William SanAsa

214121 CSEBS

Work Q#1 2.) 0010

3. 70011

{0=03·6%2=2 €0100 10113 %x= Z \$0100 10113 =

至11003・6%な=己 § 1100 11 00 3 12= € 1100 1100 3 =

4. )0100

5.) 0101

3 0010 11013 1.2= ₹0010 11013 =

€00103·6 %. 2= 2 \$10103·6 %. X= 2 £1010 12123 %2= 2101010103 =

6.70110

7.) 0111

€01103·6%, 2= 2 €11103·6%, 2= 2 € 0110 21123 % 2= €1110 <del>200</del>13%2= 3011001103=

3 1110 00013 =

8.) 1000

9.)1001

₹00013 · 6 7. 2 = 2 \$ 0001 11103 1.2= 3 000 1 1110 3 =

€10013.67,2= = { 1001 12213 / 2= 3 1001 10013=

Į.		100
	William Santosa	
	2/4/21	
	CSEIBS	
	Work Q#1	
	10.7 1010	11.) 1011
	{01013·6 1/. 2 = c	€11013.6 % 2= 0
h	その101 21213 % 2=	£ 1101 12323 % 2 =
	20101 01013 =	₹1101 00103 =
-		
	12.) 1100	13.) 1101
	5 4 4 1 2 7	· · · · · ·
	200113 . 6 1/. 2 = 2	£10113·6 7.2 = €
	20011 22113 1.2=	€1011 13223 % 2 =
	{0011 00113 =	21011 01003 =
	14.) 1110	15.) 1111
	201113 67.2=2	511112
	20111 32223 4.2=	311113.67.2=2
	20111 10003 =	£1111 3333 3 % 2 =
	20(11 10003 =	{
- '>		

-	William Sands 2/4/2   CSE 135
	Work QA2
	(0111)
	Let H = ) 1 10 1
	3 1110
~	1000
	( 0 1 0 D
· · · · · · · · · · · · · · · · · · ·	0010
	0001
	2a.) 21110 00113
	21110 00113 · H % \( \tau = \overline{2} \) 2 \( \tau \) 2 \( \tau \) 3 \( \tau \) \( \tau = \overline{2} \) 2 \( \tau \) 10 \( \tau \) 3 \( \tau \) = \( \overline{2} \)
	There is an error in the 7th row as 2 matches
	It. Flip 7th element.
	2 1110 000 13 . H 1/. 2 = e7 ] Double cheek
	2 104 13 / 4 = 6
	₹00003 = 2
	The original number is 7, or Olling, since
	m = 21110 3

William Santosa 214/21 CSE 135 Work QA2 2b.) 21101 1000 3 2110 € 10003 · H 1/2 = e 至3232311入=已 210103 = e The ta matrix e doesn't match any rows, there are there is more lerror in He ?.

	William Santosu 214/21
	C)£135
	Work Q#3
	3. Fill out look up table
	9 0 0
	2 1 4
and the department of the depa	2
	8 15 HAM-ERR
	4
A design of the second of the	5 (0111)
	6 H = 1011)
World	1101
	8 1110
	1000
0	10) (0100
	0010
	0001
1.2.2.	
	For 0 - 15, just check if the binary marches
	one of the rows in H. If it matchely the lookup valve is
And the second s	
	He row # - 1, else, it's a HAM_ERR.
	0.) 0 7.)3 (4)0
	1.) 4 8.) 7 16.) HAMERA
	2.) 6 9.) HAM-ERR
	3-) FIAM-ERR 10.) HAMERR
	4.7 6
	5.7 HAM-ERR 12.7 HAM-ERR
	6.) HAM-ERR 13.) 1

11	1/4
	William Santosa
	114/21
9	CSE135
	Work #2 (orrected
	Ja.7 {1110 00113
	21100 01113 · H 1/. 2 · E
	{1 2 3 3 ₹ 7. 1 = 己
	{10113 - €}
	(150 101)
	There is an error in the and element as == 1011.
	1000 01113 11110 - 0
	{ 0 2 2 3 1. 2 = e Double cheek
	₹0 x x 1 x / x = ₹
	200003-6
	m = 210003 Original Number = 00012
	Decimal = 1
	2b.) \$ 1101 10003
	→ → →
	₹0001 10113 · H ·/. 2 = e
	2 21213 1. 2·世
	₹01013 = e
	There is at least 1 error in the Hamming lode as == {010}}
	1100 10 00 00 00 00 00 00 00 00 00 00 00
- m	

#### **DESCRIPTION**

There are many different ways to correct errors (known as noise) in data and one such way is with a Hamming Code. A Hamming Code is a simple encoding and decoding technique that allows receivers and senders to transmit extra bits of data in order to detect and correct these errors. In this lab, we implement our own Hamming Code within C using the following files.

#### Files:

- generator.c
  - Contains my implementation of the Hamming Code generator
- decoder.c
  - Contains my implementation of the Hamming Code decoder
- bm.h
  - Contains the BitMatrix ADT interface
- bm.c
  - Contains my implementation of the BitMatrix ADT interface
- hamming.h
  - File provided in resources repository, contains the Hamming Code module
- hamming.c
  - Contains my implementation of Hamming Code module
- error.c
  - File provided in resources repository
- Makefile
  - Runs program (All, gen, dec, err, clean, format)
- README.md
  - Information about building, running, and options of the program
- DESIGN.pdf
  - Describes purpose, covers the layout, clear description of program parts, pseudo code, and contains the pre lab questions.

#### TOP LEVEL DESIGN / PSEUDOCODE

typedef enum ham rc { HAM ERR = -1, HAM OK = 0, HAM ERR OK = 1 } ham rc;

ham\_rc ham\_init(void){

Create G and H BitMatrix

G has 4 rows 8 columns

H has 8 rows 4 columns

For loop to set the G and H matrix

Return HAM ERR if fails

Return 0 if no errors

Return HAM\_ERR\_OK if 1 error and is fixed

```
}
void ham_destroy(void){
       Free G and H columns
       Free G and H rows
       Free G and H
       Return nothing
}
ham_rc ham_encode(uint8_t data, uint8_t *code){
       BitMatrix C initialize 1 row 8 columns
       Check if data and code is valid
       Multiply data by code using for loop
       Check for HAM_OK/ERR/ERR_OK
       Return HAM OK if successful generate and valid
       Return HAM_ERR if not valid and not fixable
       Return HAM_ERR_OK if not valid but fixable
}
ham_rc ham_decode(uint8_t code, uint8_t *data){
       BitMatrix C initialize 1 row 4 columns
       Check if data and code is valid
       Multiply data by code using for loop
       Check for HAM_OK/ERR/ERR_OK
       Return HAM_OK if successful generate and valid
       Return HAM_ERR if not valid and not fixable
       Return HAM_ERR_OK if not valid but fixable
}
struct BitMatrix BitMatrix{
       uint32_t rows;
       uint32 t cols;
       uint8_t ** mat;
}
BitMatrix *bm_create(uint32_t rows, uint32_t cols){ // SIMILAR TO UNIVERSE.C
       Create m pointer to (BitMatrix *) calloc(1, sizeof(BitMatrix))
       Set rows and cols
       Make a mat point toward (uint8_t **) calloc(rows, sizeof(uint8_t**)
       For loop to make all rows contain cols amount of column
       For loop to set all elements to 0
       Return m
}
```

```
void bm_delete(BitMatrix **m){ // SIMILAR TO UNIVERSE.C
       Free the columns first
       Free rows
       Free BitMatrix m
       Return nothing
}
uint32_t bm_rows(BitMatrix *m){
       Return m pointer to rows
}
uint32_t bm_cols(BitMatrix *m){
       Return m pointer to cols
}
void bm_set_bit(BitMatrix *m, uint32_t r, uint32_t c){
       M pointer to mat[r][c/8] set 1
       Return nothing
}
void bm_clr_bit(BitMatrix *m, uint32_t r, uint32_t c){
       M pointer to mat[r][c/8] set 0
       Return nothing
}
uint8_t bm_get_bit(BitMatrix *m, uint32_t r, uint32_t c){
       Return m pointer mat[r][c/8]
}
void bm_print(BitMatrix *m){ // DO THIS FIRST
       For loop to access every element and print out the bitmatrix
       Return nothing
NOTE: Will definitely be altered (Listed below)
```

### **DESIGN PROCESS / MODIFICATIONS**

- Using pointers for shifting bits in the BitMatrix stuff
- Using pointers for shifting bits in the hamming code
  - Checking with the Q#1 and Q#3 prelab question using shift bits
- Generator.c uses one input, decoder.c needs two inputs
  - o fgetc() twice, double check within while loop and make a label + goto to get out
- Encode and decode needs to be double checked
  - o Don't need to flip the code bit
- Decode needs if this then change this bit, else return error or return ok

- Makefile needs format
- Counters to keep track of iterations