Ayush Agarwal
[17114017]

Assignment 7

3rd October 2019

1. Transmit a binary message (from a sender to a receiver) using socket programming in C and report whether the received msg is correct or not; using the following error detection algorithms:

    1. Single Parity Check
    2. Two-dimensional Parity Check
    3. Checksum
    4. Cyclic Redundancy Check (CRC)

Example 1:

Sender -

```
(base) djikstra@helios:~/Academic/CSN361/L7$ ./Q1sender
Choose which algorithm to check:
 1. Single Parity Check
 2.Two-dimensional Parity Check
 3.Checksum
 4.Cyclic Redundancy Check (CRC)
1
Enter Message length:
5
Enter the message to send
10011
Message after adding parity: 100111
1.Add an error
2.Transmit the message
1
Choose how to add an error
 1.Manually add an error
 2.Randomly add an error
1
Enter number of bits to be flipped
1
Enter index to flip
2
message after adding error: 110111
```

Receiver -

```
(base) djikstra@helios:~/Academic/CSN361/L7$ ./Q1receiver
Recieved message: 110111
Error detected: Yes
```

Example 2:

Sender -

```
(base) djikstra@helios:~/Academic/CSN361/L7$ ./Q1sender
Choose which algorithm to check:
 1. Single Parity Check
 2.Two-dimensional Parity Check
 3.Checksum
 4.Cyclic Redundancy Check (CRC)
4
Enter length of Divisor:
4
Enter Divisor:
1001
Enter Message length:
7
Enter the message to send
1010101
Remainder: 110
Message after CRC: 1010101110
1.Add an error
2.Transmit the message
1
Choose how to add an error
 1.Manually add an error
 2.Randomly add an error
2
Enter probability of induced error
0.2
Transmitted message: 0101000001
```

Receiver -

```
(base) djikstra@helios:~/Academic/CSN361/L7$ ./Q1receiver
Recieved message: 110111
Error detected: Yes
```

2. Transmit a binary message (from a sender to a receiver) using socket programming in C. Using Hamming code detect and correct errors in the transmitted message, if any.

Server:

```
(base) djikstra@helios:~/Academic/CSN361/L7$ ./Q2server
The message from the client is 111000101

9
The data word is: Number of redundancy bits 4
1 1 1 0 0 0 1 0 1

 Redundancy bits: 0 0 1 0Assuming single error, the error location is 4
11001
(base) djikstra@helios:~/Academic/CSN361/L7$
```

Client:

```
(base) djikstra@helios:~/Academic/CSN361/L7$ ./Q2client
Enter the size of the data word: 5
1
1
0
0
1
The number of redundancy bits is: 4
The encoded code word is: 1 1 1 0 0 1 1 0 1
Enter the number of errors ( we can correct only one error ): 1
Enter the location of the erroneous bit: 4
Flipping the 4 bit
1 1 1 0 0 0 1 0 1
Client : Sending the code word
Message from server : Read your message
(base) djikstra@helios:~/Academic/CSN361/L7$
```

3. Write a C++ program to compress a message (non-binary, can be anything like a text message or a code like hexadecimal, etc.) using the following data compression algorithm:

    1. Huffman
    2. Shannon-Fano

Huffman Example -

```
(base) djikstra@helios:~/Academic/CSN361/L7$ ./Q3 input.txt
Select Encoding
1. Huffman
2. Shannon-Fano
1
Huffman Codes are :

k:11111
m:11110
p:11101
o:1101
a:11100
r:1011
 :1100
u:1010
e:100
n:0000
c:0001
t:001
s:0111
b:0100
w:0101
l:0110

Original string was :
computer networks lab

Encoded string is :
000111011111101110110100011001011110000001000010101110110111111101111100011011000100
Encoded message length:84
```

Shannon-Fano Example -

```
(base) djikstra@helios:~/Academic/CSN361/L7$ ./Q3 input.txt
Select Encoding
1. Huffman
2. Shannon-Fano
2

16
        0.095238        000
e       0.095238        001
o       0.095238        010
r       0.095238        0110
t       0.095238        0111
a       0.047619        1000
b       0.047619        1001
c       0.047619        1010
k       0.047619        10110
l       0.047619        10111
m       0.047619        1100
n       0.047619        11010
p       0.047619        11011
s       0.047619        1110
u       0.047619        11110
w       0.047619        11111

1010010110011011111100111001011000011010001011111111010011010110111000010111100010001
```