CO253 - Programming Project (worth 10%)

Objectives
Making you familiar with,

- Arrays
- Loops
- Functions
- Pointers
- Debugging
- Good coding practices

Introduction

For this project, you need to implement your own program to perform a secure, error-free binary communication. You need to implement a cryptographic function which can encrypt messages to binary sequences and also can decrypt binary sequences and obtain messages. In addition, you need to integrate the Hamming code to your binary sequence to detect errors that occur during transmission and correct them (assuming only single-bit errors can happen).

When you are given a message (the plain text), you should first represent it in a bit sequence and then get the corresponding encrypted bit sequence (known as the cipher text) and add error detection/correction bits to the cipher text. The encrypted bit sequence + parity bits at relevant positions will be the final outcome in this case.

When you are given a received bit sequence (encrypted bit sequence + parity bits), you should first check whether there were any errors during bit transmission (if yes, correct it) and then remove all parity bits and obtain the encrypted bit sequence. Then decrypt the bit sequence to obtain the intended message (the plain text) from that. In this case, plain text (message) should be the final output.

To get the relevant binary representation of a message, we should first get the ASCII value of each character (of the plain text) and represent that in binary with 8 bits each. We should append all these binary representations together to form one binary sequence for each plain text.

Eg   Message is “Hi”   (H = 72 and i = 105)   Bit sequence = 0100100001101001

Encryption explanation: To encrypt the obtained binary sequence, it is written down k times, shifted right by 0,1, 2, ..., k-1 bits. Then, each of the columns is XOR-ed together to get the final encoded string.

If k = 3 for the above binary sequence,

```
0 1 0 0 1 0 0 0 0 1 1 0 1 0 0 1        Shift 0
  0 1 0 0 1 0 0 0 0 1 1 0 1 0 0 1      Shift 1
    0 1 0 0 1 0 0 0 0 1 1 0 1 0 0 1    Shift 2
------------------------------------------------
0 1 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 1    XOR-ed string/ the encrypted text
```

Error detection/correction - Hamming code (Use even parity)

https://www.tutorialspoint.com/error-correcting-codes-hamming-codes
https://www.youtube.com/watch?v=kAezzdEGJ8A&t=39s&pbjreload=10

Hamming code: Adding parity bits to the above obtained encrypted bit sequence.

p1 p2 p4 p8 p16

X X 0 X 1 1 1 X 1 1 1 0 0 1 0 X 0 0 1 1 1 1 1 - highlighted X should be 1

1 X 0 X 1 1 1 X 1 1 1 0 0 1 0 X 0 0 1 1 1 1 1 - highlighted X should be 0

1 0 0 X 1 1 1 X 1 1 1 0 0 1 0 X 0 0 1 1 1 1 1 - highlighted X should be 0

1 0 0 0 1 1 1 X 1 1 1 0 0 1 0 X 0 0 1 1 1 1 1 - highlighted X should be 0

1 0 0 0 1 1 1 1 1 1 1 0 0 1 0 X 0 0 1 1 1 1 1 - highlighted X should be 1

Parity bits are selected such that the number of 1s in highlighted bits are becoming an even number (since we are considering even parity). Check the given links to understand how the number of parity bits required is decided and understand further about error correction code.

When dealing with a received bit sequence we should first check whether there was an error during transmission. Check the given video, it clearly explains how the error is detected and how to correct it. You should implement the exact same thing https://www.youtube.com/watch?v=N8Yy0-4YMS4

After detecting and correcting the error, you should remove parity bits and obtain the encrypted bit sequence and follow the given algorithm to decrypt it.

Decryption explanation: Received text (S) after removing parity bits is 0111111001000111111 and k = 3

- The first digit in S=0 so our decrypted binary string is going to start with 0.
- The next digit in s is 1, we already know the shifted value is 0 then we XOR them together we get 1. Therefore the second bit of decrypted bit sequence is 1.
- The 3rd digit of s is 1, We know the first digit of our shift2 string is a 0 and 2nd digit of shift1 string is 1, we XOR them all together, the answer is 0, Therefore the 3rd digit of original bit sequence is 0.
- We should continue with that logic until the end

After decrypting you will obtain the original binary sequence of the message. Then you need to get the decimal representation of the binary and obtain ASCII values. Then get the corresponding characters to obtain the message.

Expected Input to Your Program

- The first-line contains a char [C or P] which indicates whether you are getting
  - **P** sending message (plain text)
  - **C** received bit sequence (encrypted bit seq + parity bits)
- Next line contains 2 or 3 space-separated integers.
  - ★ If you get P previously, you will get 2 integers N and k where N is the length of the message and k is the number of shifts.
  - ★ If you get C previously, you will get 3 integers N, L and k where N is the length of the message(original message in this case output), L is the length of the receiving bit sequence and k is the number of shifts.
- The next line contains a string. It should be considered as the message (plain text) if you got P in the first line, and if you got C then it is the received bit sequence (encrypted bit sequence with parity bits for error correction).

The Expected Output (Final output of milestone 4) of Your Program

If you got a plain text as the input, you have to output the encrypted bit sequence with error correction bits included.
If you got the received bit sequence as the input, then you should perform error detection and correction, then remove parity bits and decrypt the message, obtain the plain text from the decrypted bit sequence and output the message. (If you detect an error you should print “Error detected and corrected!” in the first line and then print the message)

Observe the below examples with all the steps included!

Input:

P
2 3
Hi

Message to be passed is “Hi” (N=2) and K = 3

‘H’ ascii value = 72								‘i’ ascii value = 105							
0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	1

0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	1			Shift 0
	0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	1		Shift 1
		0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	1	Shift 2
0	1	1	1	1	1	1	0	0	1	0	0	0	1	1	1	1	1	Encrypted message

⬇

XOR (1, 1, 0) = 0

0	1	1	1	1	1	1	0	0	1	0	0	0	1	1	1	1	1	Encrypted bit sequence					
P1	P2	0	P4	1	1	1	P8	1	1	1	0	0	1	0	P16	0	0	1	1	1	1	1	
1	0	0	0	1	1	1	0	1	1	1	0	0	1	0	1	0	0	1	1	1	1	1	Output bit sequence

Input:

C

23 3

10001110111001010011111

1	0	0	0	1	1	1	0	1	1	1	0	0	1	0	1	0	0	1	1	1	1	1	Input bit sequence
P1	P2	0	P4	1	1	1	P8	1	1	1	0	0	1	0	P16	0	0	1	1	1	1	1	Check for errors
0	1	1	1	1	1	1	0	0	1	0	0	0	1	1	1	1	1	Remove parity and obtain encrypted bit seq					

0	1	1	1	1	1	1	0	0	1	0	0	0	1	1	1	1	1	Encrypted message
0																		Shift 0
	0																	Shift 1
		0																Shift 2

0	1	1	1	1	1	1	0	0	1	0	0	0	1	1	1	1	1	Encrypted message
0	1																	Shift 0
	0	1																Shift 1
		0	1															Shift 2

⬇

XOR (1, 0) = 1

0	1	1	1	1	1	1	0	0	1	0	0	0	1	1	1	1	1	Encrypted message
0	1	0																Shift 0
	0	1	0															Shift 1
		0	1	0														Shift 2

0	1	1	1	1	1	1	0	0	1	0	0	0	1	1	1	1	1	Encrypted message
0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	1			Shift 0
	0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	1		Shift 1
		0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	1	Shift 2

0	1	0	0	1	0	0	0	0	1	1	0	1	0	0	1
ascii value = 72								ascii value = 105							
H								i							

Problem breakdown

We have divided your project into 3 milestones for you to progress one by one. We are giving deadlines for each milestone. In each milestone, you have to build functions that are useful for the completion of the project, so **each milestone will have different inputs and outputs which will be clearly explained in the hackerrank tests** provided.

Milestone 1: Reading the input and if the input is the message, then it has to be encoded to its equivalent binary representation, and if the input is the bit sequence, the sequence has to be decoded to the corresponding message.

Milestone 2: The encryption and the decryption functionality on the binary sequence.

Milestone 3: (a) Applying hamming code to a given binary sequence which is to be sent and error detection and removing hamming code from a binary sequence which is received. (b) Integration of work done up to now to achieve the final goal as previously explained in the description.

Submission

Your performance in each milestone will be considered to your final mark. We will do plagiarism checks on your answers therefore please make sure you do not give your code to others or take others code, including not from the Internet.

Deadlines

Milestone 1: .20th April 2020, 11.55pm
Milestone 2: .29th April 2020, 11.55pm
Milestone 3: .11th May 2020, 11.55pm