## Question - 1

**CO253 Programming Project Milestone 03**

In this milestone, you have to implement functionalities to,

- Add hamming code to a given bit sequence,
- Error detection using a given bit sequence with Hamming code, correcting it and obtaining the original message.

If the input is the message, then (a) encode it to its equivalent binary representation, (b) encrypt it and (c) add parity bits to the bit sequence, and if the input is the bit sequence, you have to (a) detect whether there are errors, (b) correct them, (c) remove parity bits, (d) decrypt the sequence and then (e) decode it to obtain the corresponding message.

You have already implemented functions to encode, encrypt, decode and decrypt. You have to use those 4 functions and implement the expected functionality of the program. (See the project specification document for related links and explanations about Hamming code.)

Since this is the final milestone you can do the work in your own way.

**Inputs**

- First-line contains a char[C or P] which indicates whether you are getting Message/Plain text(if P) or Bit sequence(if C)
- Next line contains 2 or 3 space-separated integers.
    1. 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.
    2. 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.
    1. It should be considered as the Message(Plain text) if you got P in the first line and you need to encode it then encrypt it and add parity bits.
    2. If you got C you are getting the received bit sequence as the input, you should perform error detection and correction, remove parity bits, decrypt the message, obtain the plain text from the decrypted bit sequence and output the message.

**Constraints**

Length of message < 1000
Length of bit sequence < 10000
Number of shifts < 1000

**Output**

- If you got the plain text as the input, you have to output the encrypted bit sequence with parity bits.
- If you got the received bit sequence as the input, you should output the message. (If you detect an error you should print “Error detected and corrected!” in the first line and then print the output message)

**Sample test cases**

Use the provided reference to understand how Hamming code is applied to a bit sequence and how errors are detected and corrected (which is not explained step by step in the example.)

*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 sequen | |

*Output:*

10001110111001010011111

*Input:*
C
2 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 |

XOR (1, 1, 0) = 0 Continue until all bits are found

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | | | | | | | |

*Output:*

Hi

*Input:*
C
2 23 3
10001110111001010011110

| | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 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 | 0 | Check for errors |
| P1 | P2 | 0 | P4 | 1 | 1 | 1 | P8 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | P16 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | Error detected |
| P1 | P2 | 0 | P4 | 1 | 1 | 1 | P8 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | P16 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | Error corrected |
| 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 |

XOR (1, 1, 0) = 0 Continue until all bits are found

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | | | | | | | |

*Output:*

Error detected and corrected!
Hi