IT 205

Computer Communication and Networking – Lab

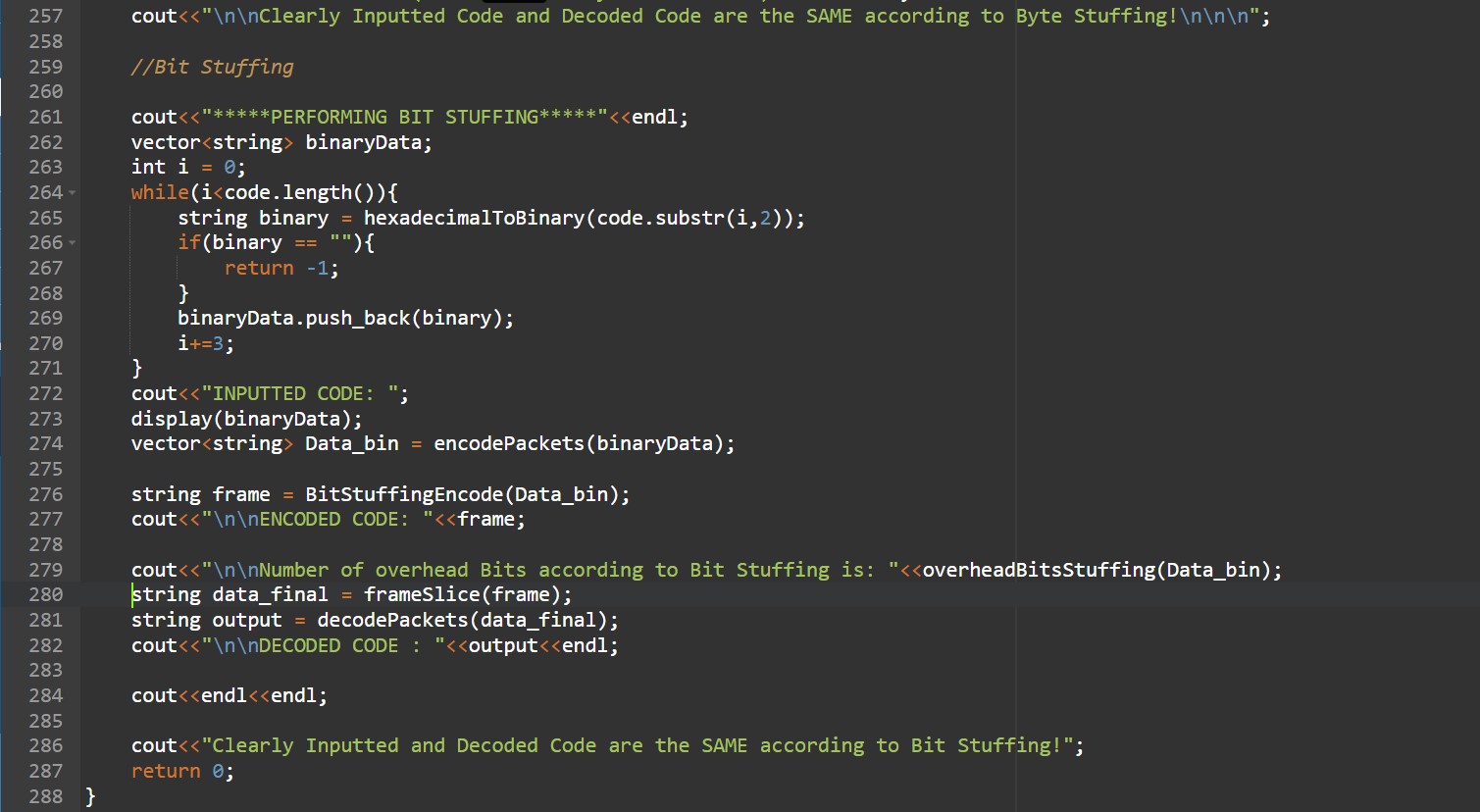
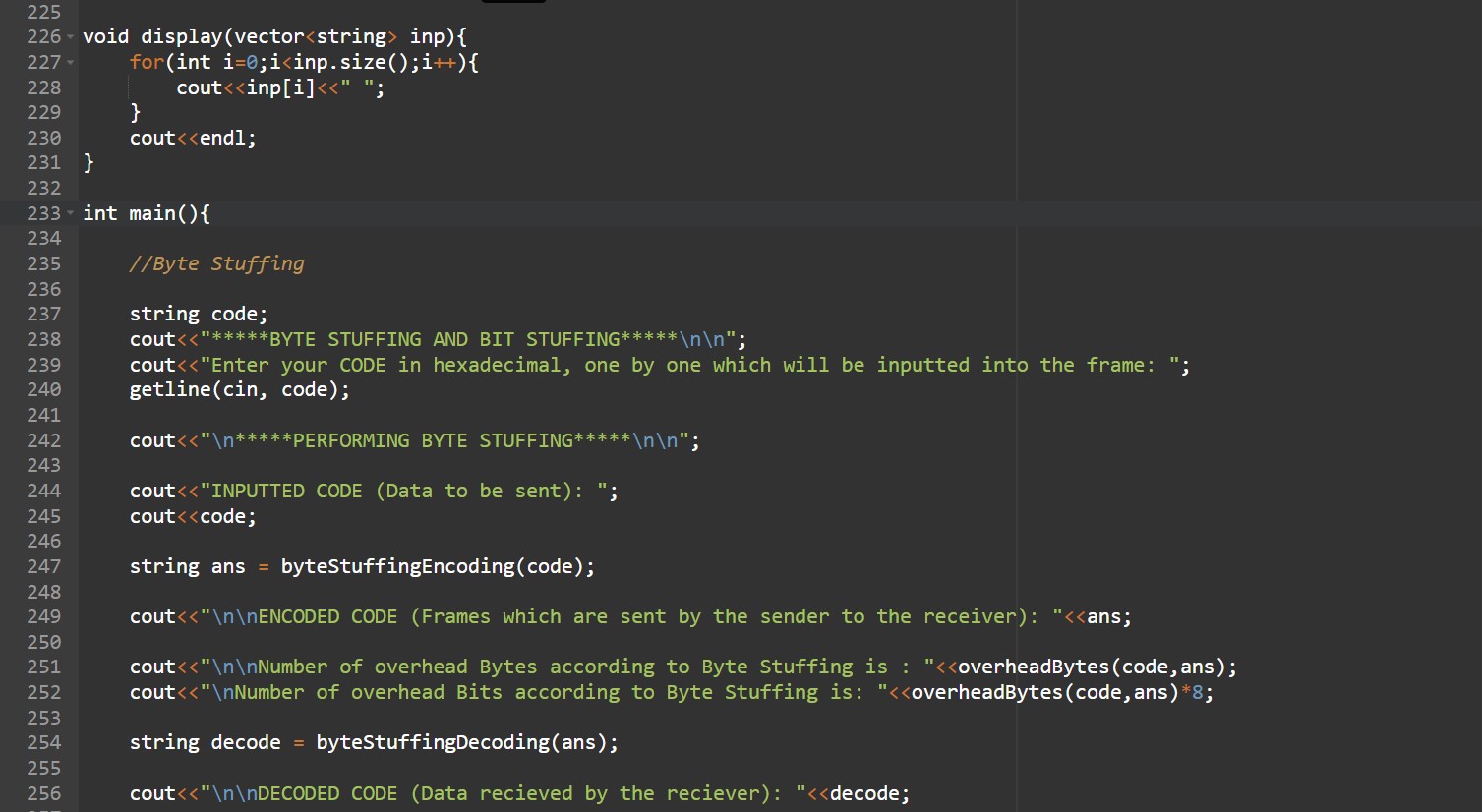
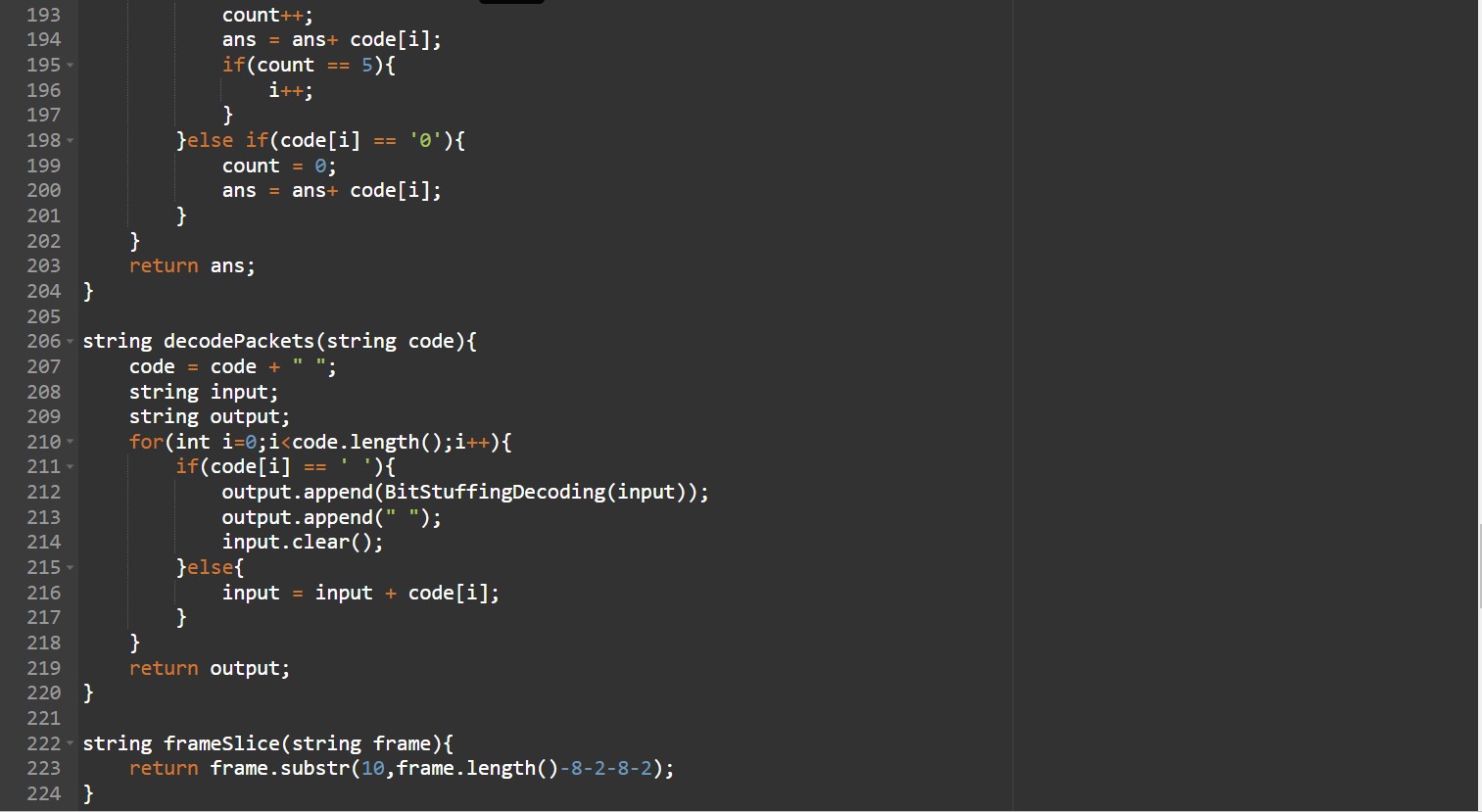
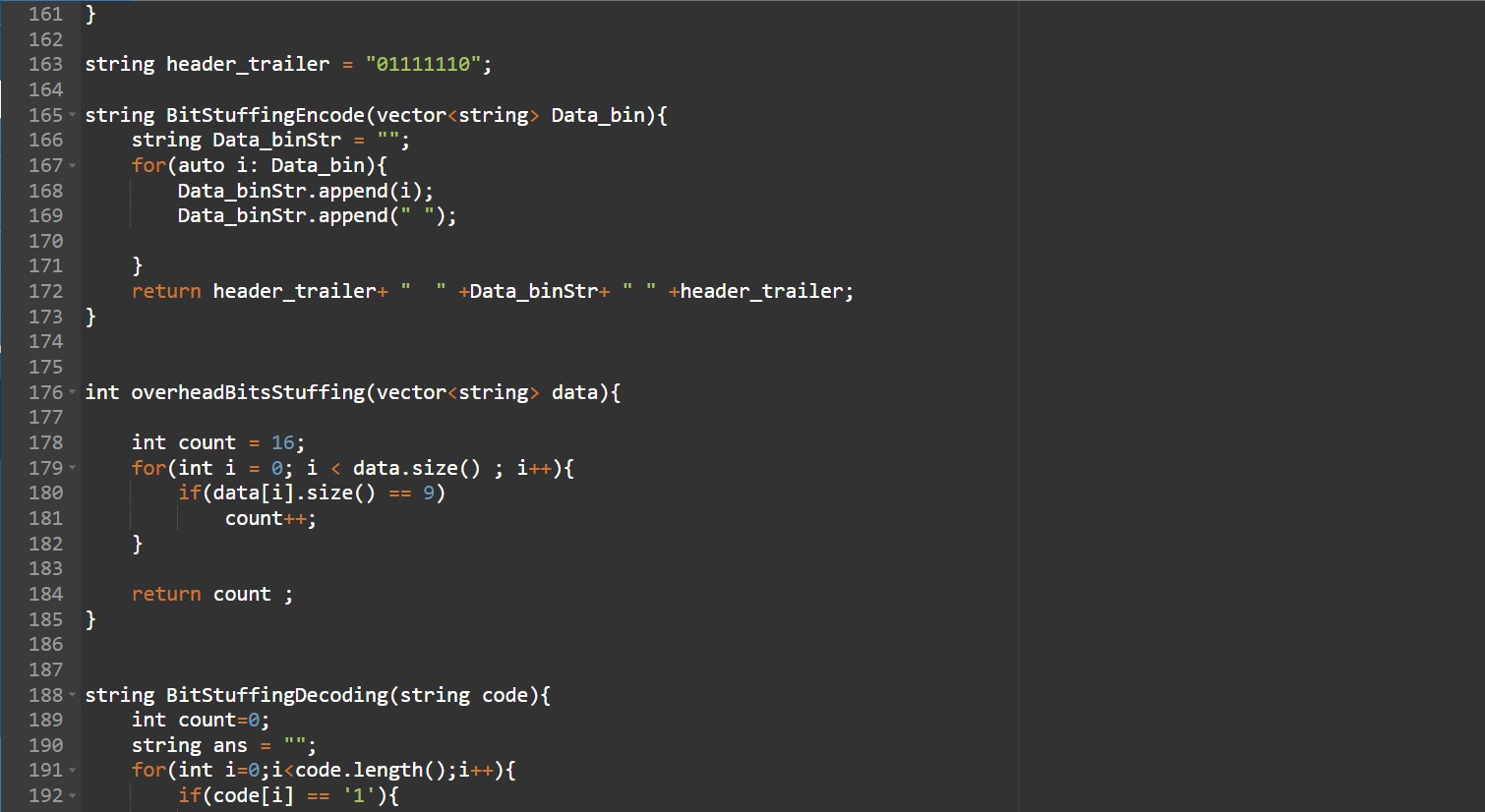
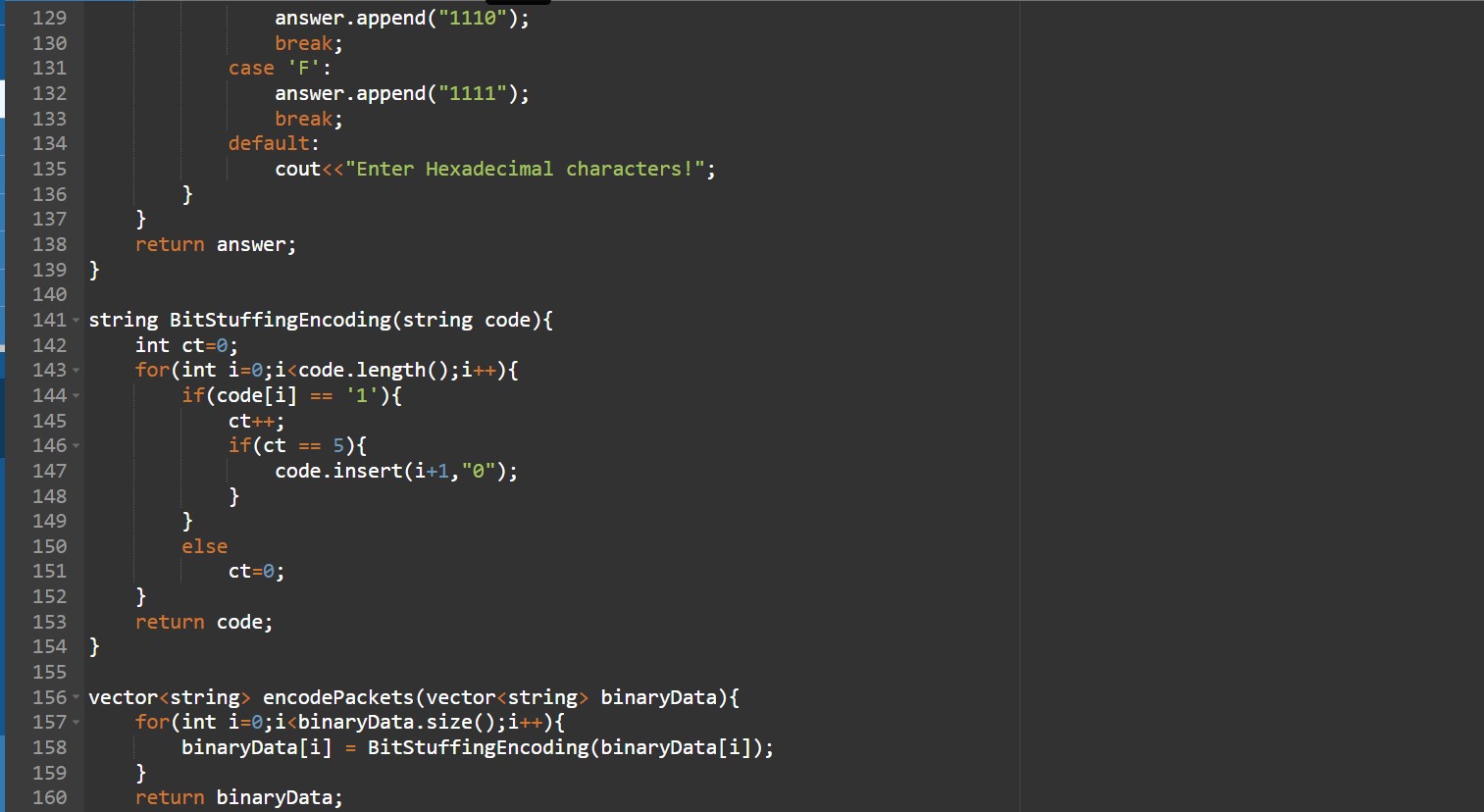
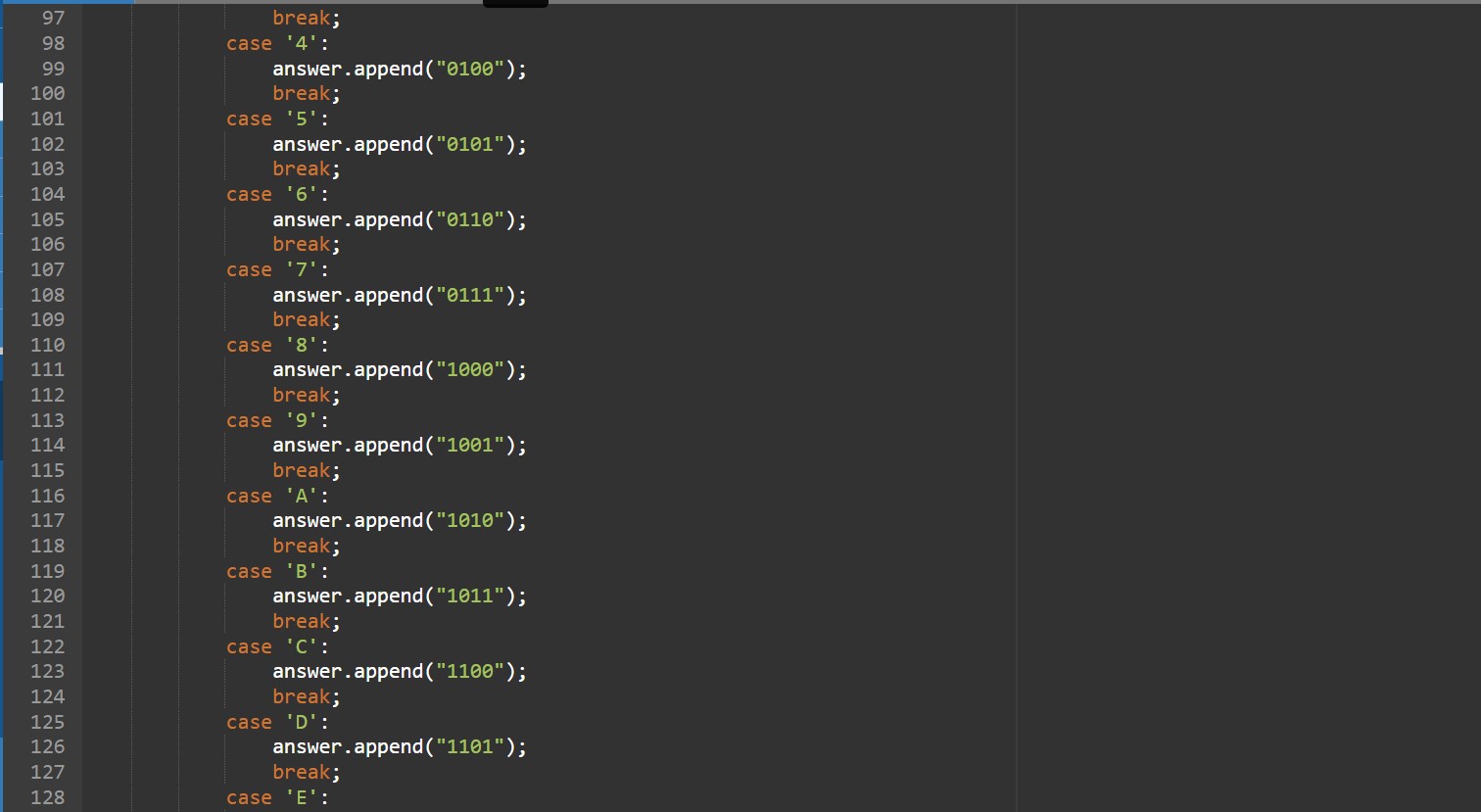
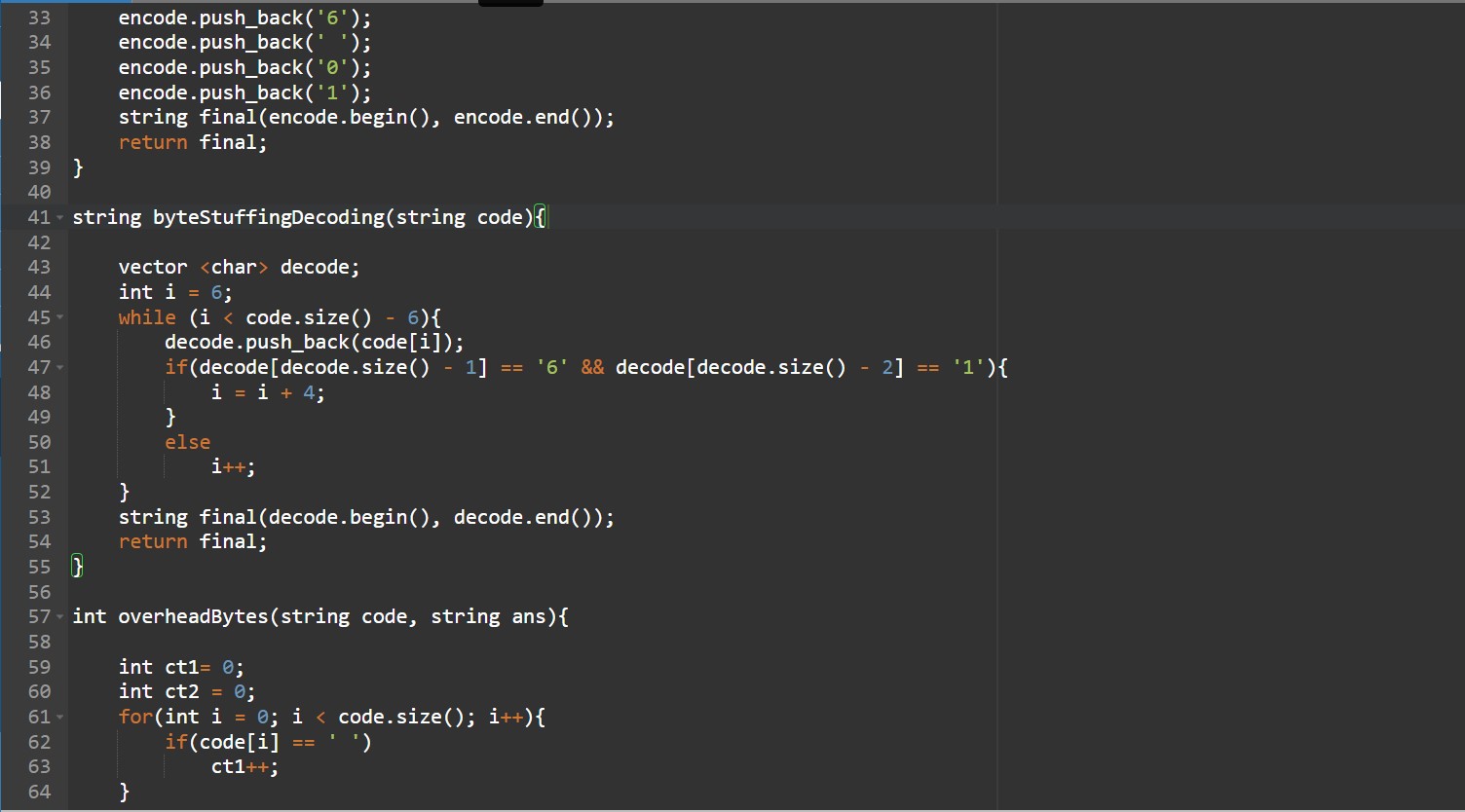
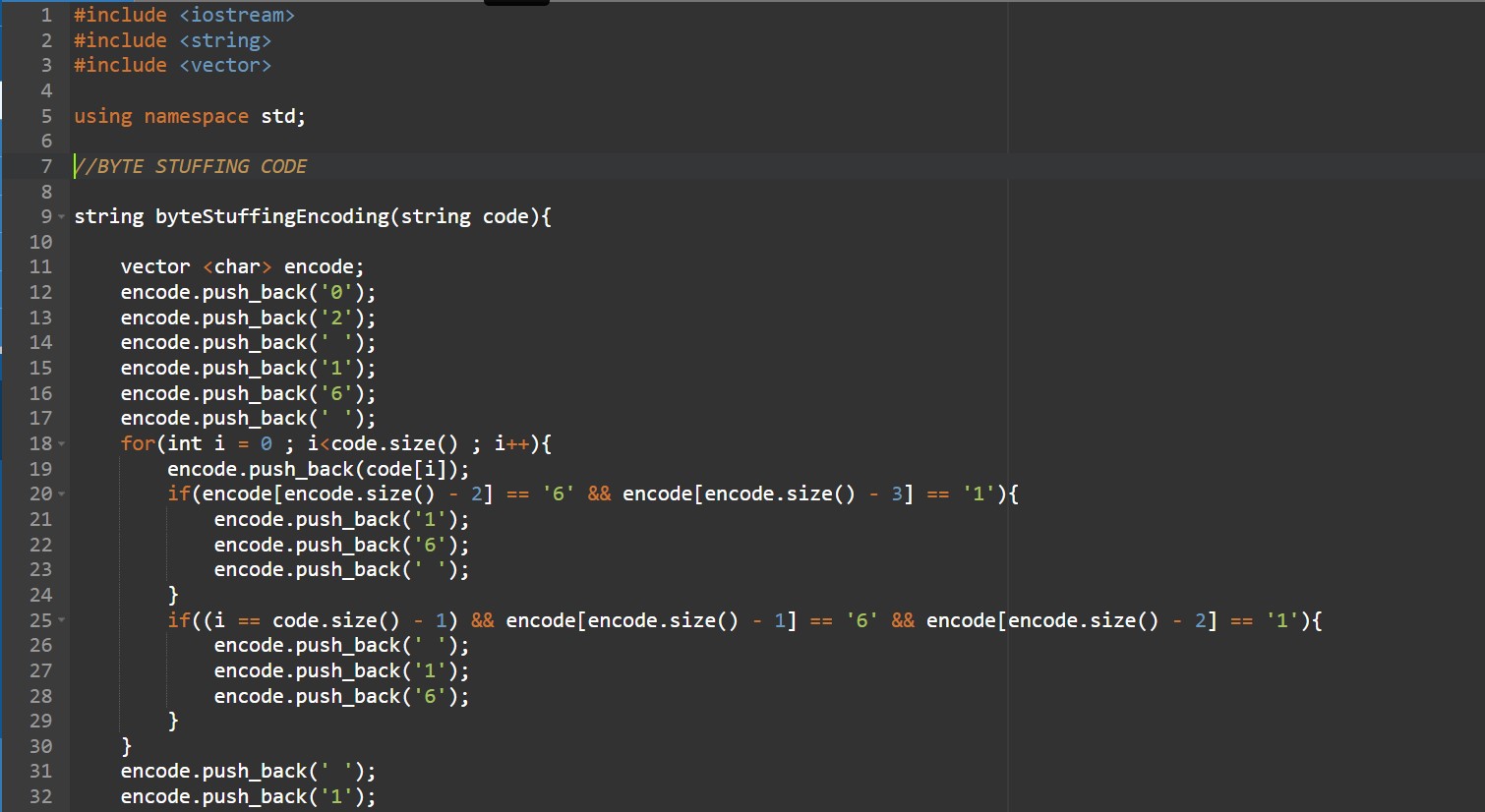
Assignment 4

By-

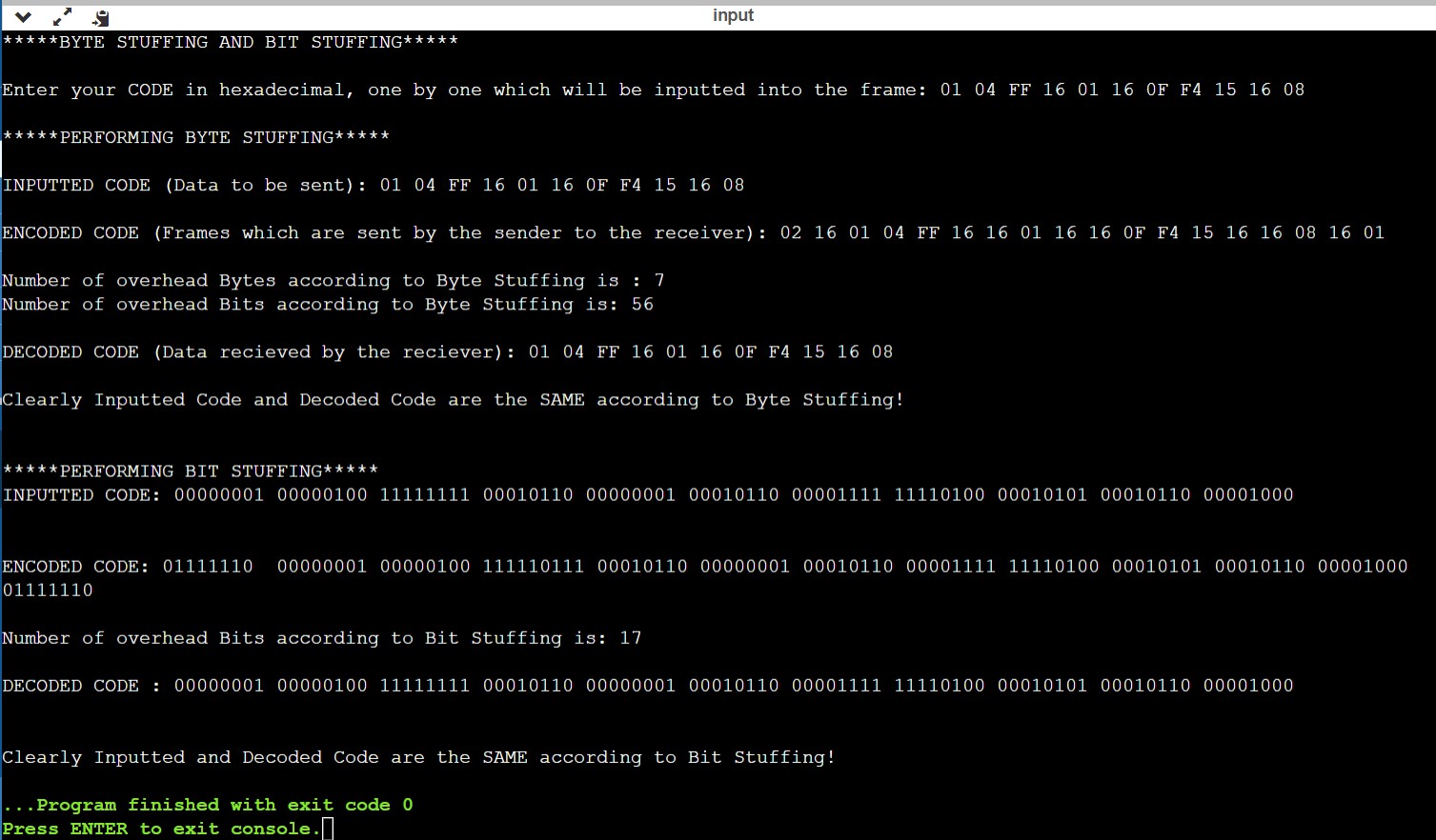
Sachin Prasanna

211IT058

Source Code Screenshots:



Output Screenshot:



Observations:

Implemented Byte Stuffing and Bit Stuffing by writing a program in C++.

Bit Stuffing and Byte Stuffing let the receiver know about the size of the frame by using different techniques. (i.e. Adding extra Bit for Bit Stuffing and extra Byte for Byte Stuffing)

Given Input in hexadecimal: 01 04 FF 16 01 16 0F F4 15 16 08

**Byte Stuffing**

Since the given DLE is 16, whenever 16 is encountered in the input, an extra 16 is stuffed into the input. This helps when the receiver decodes the code and knows to remove one 16 when two simultaneous 16s are encountered.

In the given Input, there are 3 16s, so another 3 16s will be stuffed into the input. The STX, DLE, DLE, ETX are 1 byte each and 4 bytes in total and they are also stuffed into the input. Since each 16 is a byte and 8 bits, the total number of stuffed bytes or Overhead bytes = 4+3 = 7. Correspondingly, the number of stuffed bits will be 7\*8 = 56, which is shown in the output. (1 byte = 8 bits)

Also, STX and DLE are stuffed into the starting of the input and DLE and ETX are stuffed into the ending of the input. Finally, this frame is sent to the receiver.

Upon receiving, the receiver removes 16 if there are 2 16s together. This gets back the original sent data.

This is the process of Byte Stuffing.

**Bit Stuffing**

Since Bit Stuffing works on Bits, the given hexadecimal input is converted into Bits.

The header and trailer will be same in Bit Stuffing unlike Byte Stuffing, and the given sequence for the same is 01111110.

As we traverse through each byte, if any byte has 5 1s in a row, then we add an extra 0 to it to differentiate from the starting and ending sequence.

The header and trailer are 16 bits each, so by default the overhead Bits will be 16 or more. Then, since only one byte has 5 1s in a row, the number of Overhead Bits = 16+1 = 17.

Also, the sequence is stuffed into the starting and ending of the input and sent to the receiver for decoding.

Upon receiving, if the receiver notices 5 1s in a row, apart from starting and ending, it removes the extra 0 after it, as it knows it has been added extra by the sender. The original data is thus retrieved.

This is the process of Bit Stuffing.

THANK YOU