# Assignment 2

By Raghav Sakhuja
2021274

## Part A

Header File:byte_stream.hh

ByteStream class declaration: This defines the ByteStream class, which represents an in-order byte stream.

Private members:

- std::deque<char> buffer: A deque to store the byte data.
- const size_t capacity: The maximum capacity of the stream.
- size_t bytesWritten: Tracks the total bytes written.
- size_t bytesRead: Tracks the total bytes read.
- bool InputEnded: Indicates whether the input has ended.
- bool _error: A flag indicating if the stream has suffered an error.

Constructor: ByteStream(const size_t capacity): Initializes the ByteStream object with the specified capacity.

Source File byte_stream.cc:

- Constructor (ByteStream::ByteStream(const size_t capa)): This is the constructor of the ByteStream class. It takes a single parameter, capa, which is the initial capacity of the byte stream. It initialises several member variables: capacity, _error, InputEnded, bytesWritten, and bytesRead.

- size_t ByteStream::write(const string &data): This method is used to write data into the byte stream. It takes a string data as input and attempts to write its characters into the buffer. It respects the capacity constraint and returns the number of bytes successfully written.

- size_t ByteStream::remaining_capacity() const: This method calculates and returns the remaining capacity in the byte stream's buffer.

- void ByteStream::end_input(): This method sets the InputEnded flag to true, indicating that no more input will be added to the byte stream.

- string ByteStream::peek_output(const size_t len) const: This method allows you to examine (peek) a specified number of bytes from the output side of the buffer without removing them. If len is more than the size of data, it returns the max amount of bytes possible.

- void ByteStream::pop_output(const size_t len): This method removes a specified number of bytes from the output side of the buffer.If len is more than the size of data, it raises an error.

- std::string ByteStream::read(const size_t len): This method reads (copies and then pops) the next len bytes from the stream and returns them as a string.If len is more than the size of data, it raises an error.

- bool ByteStream::input_ended() const: This method returns true if the input has ended (as set by end_input()), otherwise false.

- size_t ByteStream::buffer_size() const: This method returns the current size of the buffer.

- bool ByteStream::buffer_empty() const: This method returns true if the buffer is empty; otherwise false.

- bool ByteStream::eof() const: This method returns true if the input has ended and the buffer is empty (end of file); otherwise false.

- size_t ByteStream::bytes_written() const: This method returns the total number of bytes written to the byte stream.

- size_t ByteStream::bytes_read() const: This method returns the total number of bytes read from the byte stream.

Following is result of the test cases provided to us: