Step 1: Group the binary numbers into sections

We have been given a binary sequence:

```
10110011 10101011 01011010 11010101
```

Each group here is 8 bits, so we can consider these as four groups of 8-bit binary numbers.

Step 2: Calculate the Sender Checksum

The sender checksum is calculated by adding these binary values together, applying 1's complement if necessary.

1. Convert each group of 8-bit binary numbers to decimal (or simply add in binary):
   - `10110011` (179 in decimal)
   - `10101011` (171 in decimal)
   - `01011010` (90 in decimal)
   - `11010101` (213 in decimal)
2. Add the binary numbers:
   To keep it simple, add these values in binary. If there's any carry beyond 8 bits, wrap it around by adding it back to the lower bits.
3. Take the 1's complement of the sum to get the sender checksum.

Let's calculate that step-by-step.

Step 3: Calculate the Receiver Checksum

The receiver checksum is calculated similarly:

1. Add the received binary sequence and the sender's checksum.
2. Take the 1's complement of this sum.

If the result is all `0` s, the transmission is error-free; otherwise, an error has occurred.

Step 4: Verify Data Transmission

- If the receiver checksum results in all zeros, it indicates that the data was correctly transmitted without any errors. If there's any other result, an error has occurred.

I'll perform these calculations for you.

The results of the checksum calculations are as follows:

1. **Sender Checksum**: 112 (in decimal), or `01110000` in binary.
2. **Receiver Checksum**: 0, which confirms that the transmission is error-free.

Explanation for Correct Data Transmission:

Since the receiver checksum results in all zeros, this indicates that the data was transmitted correctly without errors. If any bits had changed during transmission, the checksum would not result in zero, revealing an error.