1. Consider a generator polynomial $C = x^3 + 1$, when the data is 101100101, what will be the final messge transferred after CRC encoding?

Answer:
Since all CRC calculations are done in modulo 2 arithmetic without carries or borrows, both adition and substraction are identical to XOR.

The detailed calculation is as follows:

```
         101001100
1001√101100101 000
         1001
           1000
           1001
                1101
                1001
                  1000
                  1001
                       100
```

The remainder of the above calculation is 100, so the final messge transferred should be 101100101100.

2. Based on the previous problem, how can the receiver detect an error? For example, if the message has been changed to 101110101, how will the receiver detect it?

Answer:
Because both the sender and receiver have agreed on the same generator polynomial C, the receiver can calculate the remainder according to C. After the message has become 101110101, now the remainder is 110, different from the attached CRC remainder 100. So that the receiver can tell there is something wrong with this transfer.

3. Why CRC is used? Give some examples where CRC can be used.

Answer:
CRC is used to detect errors during data transfer. By carefully selecting polynomial generators, CRC can detect most of the errors. Thus, it provides a reliable channel between the sender and receiver. CRC can be used in different layers of the network stack, for example, data link layer and network layer.

4. Compare CRC-12, CRC-16, CRC-CCITT and CRC-32. What is common among them? Why is it helpful?

Answer:
The common thing among them is that (x+1) is a factor. In this sense, all of these polynomials can detect all errors which contain an odd number of inverted bits.