Name: Class:

Task 1

1. The following three bytes are transmitted across a serial interface using odd parity.  
Insert the parity bits for each byte that is transmitted.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 0 | 1 | 1 | 0 | 0 | 1 |  |  | 1 | 1 | 0 | 1 | 1 | 1 | 0 |  |  | 1 | 1 | 0 | 1 | 0 | 0 | 1 |  |

2. The table below shows eight bytes of data that will be sent with a parity block check. An odd parity is to be used.

(a) Add the parity bit for each byte of data along with the parity byte for the whole block.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Parity | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 |
| Byte 1 |  | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| Byte 2 |  | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| Byte 3 |  | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| Byte 4 |  | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| Byte 5 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Byte 6 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Byte 7 |  | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| Byte 8 |  | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| Parity Byte |  |  |  |  |  |  |  |  |

(b) Explain how it is possible to locate an error of one incorrectly transmitted bit in the block.

Task 2

Using a barcode on the back of a book, calculate the check digit using the   
ISBN system.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ISBN** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Weight** | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 3 |  |
| **Multiplication** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Addition** | Add all the numbers | | | | | | | | | | | |  |
| **Remainder** | Find the remainder when divided by 10 | | | | | | | | | | | |  |
| **Subtraction** | Subtract the result from 10 | | | | | | | | | | | |  |

Task 3

The Luhn algorithm was devised as a checksum formula to ensure credit card numbers are valid when manually or automatically entered into a machine.

The steps in the algorithm are as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Credit Card Number | 4 | 3 | 6 | 2 | 6 | 2 | 6 | 8 | 7 | 7 | 4 | 3 | 3 | 1 | 1 | 6 |  |
| Double every other number | **8** |  | **12** |  | **12** |  | **12** |  | **14** |  | **8** |  | **6** |  | **2** |  |  |
| Subtract 9 if number > 9 |  |  | **3** |  | **3** |  | **3** |  | **5** |  |  |  |  |  |  |  |  |
| Find sum of all digits | **8** | **3** | **3** | **2** | **3** | **2** | **3** | **8** | **5** | **7** | **8** | **3** | **6** | **1** | **2** | **6** | **70** |

If the sum of all digits is a number divisible by 10, the number will be accepted. If not, it is rejected assuming an error has been input.

Would the following credit card number be accepted? Show your working.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Credit Card Number | 4 | 4 | 7 | 4 | 8 | 5 | 2 | 4 | 6 | 6 | 7 | 8 | 5 | 4 | 8 | 5 |  |
| Double every other number |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Subtract 9 if number > 9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Find sum of all digits |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Task 4

Complete the diagram below to show the following stages of a transmission using an Automatic Repeat Request (ARQ):

* The first data block sent and received correctly
* The second data block has an error in transmission, but sends correctly the second time
* The third data block is sent correctly, but the acknowledgement is never received
* The third data block is sent correctly when it is re-sent.

