

## 6ª Lista de Exercícios de Teleprocessamento

142. Um pacote de uma camada superior está dividido em 10 quadros, e cada quadro tem 80% de chances de chegar sem danos. Se o protocolo de enlace de dados não fizer qualquer controle de erros, quantas vezes em média a mensagem deverá ser enviada para que o processo inteiro seja concluído?
143. A codificação de caracteres a seguir é usada em um protocolo de enlace de dados:  
A: 01000111; B: 11100011; FLAG: 01111110; ESC: 11100000
- Mostre a seqüência de bits transmitida (em binário) para o quadro de quatro caracteres: A B ESC FLAG quando é utilizado cada um dos métodos de enquadramento a seguir: (a) Contagem de caracteres. (b) *Bytes de flag* com inserção de *bytes*. (c) *Bytes de flag* no início e no fim, com inserção de *bits*.
144. O fragmento de dados a seguir ocorre no meio de um fluxo de dados para o qual é usado o algoritmo de inserção de bytes descrito no texto: A B ESC C ESC FLAG FLAG D. Qual será a saída após a inserção?
145. Um de seus colegas, Scrooge, assinalou que é um desperdício encerrar cada quadro com um *byte de flag* e depois iniciar o próximo quadro com um segundo *byte de flag*. Um único *byte de flag* também poderia servir, e um *byte* economizado é um *byte* ganho. Você concorda?
146. Uma cadeia de *bits*, 011110111110111110, precisa ser transmitido na camada de enlace de dados. Qual é a cadeia realmente transmitida após a inserção de *bits*?
147. Quando o recurso de inserção de *bits* é usado, é possível que a perda, da inserção ou a modificação de um único *bit* provoque um erro não detectado pelo total de verificação? Se não for possível, qual é o motivo? Se for possível, como isso é feito? O comprimento do total de verificação desempenha alguma função nesse caso?
148. Para proporcionar maior confiabilidade que a obtida com um único *bit* de paridade, um esquema de codificação para detecção de erros utiliza um *bit* de paridade para verificar todos os *bits* de numeração ímpar e um segundo *bit* de paridade para todos os *bits* de numeração par. Qual é a distância de Hamming desse código?
149. As mensagens de dezesseis *bits* são transmitidas com o uso de um código de Hamming. Quantos *bits* de verificação são necessários para assegurar que o receptor poderá detectar e corrigir erros de um único *bit*? Mostre o padrão de *bits* transmitido no caso da mensagem 1101001100110101. Suponha que seja usada a paridade par no código de Hamming.
150. Um *byte* de 8 *bits* com valor binário 10101111 deve ser codificado com a utilização de um código de Hamming de paridade par. Qual é o valor binário depois da codificação?

151. Um código de Hamming de 12 *bits* cujo valor hexadecimal é 0xE4F chega a um receptor. Qual era o valor original em hexadecimal? Suponha que não exista mais de 1 *bit* com erro.
152. Uma forma de detectar erros é transmitir dados como um bloco de  $n$  linhas com  $k$  *bits* por linha e acrescentar *bits* de paridade a cada linha e a cada coluna. O canto inferior direito é um *bit* de paridade que verifica sua linha e sua coluna. Esse esquema detectará todos os erros simples (isolados)? E os erros duplos? E os erros triplos?
153. Um bloco de *bits* com  $n$  linhas e  $k$  colunas utiliza *bits* de paridade horizontais e verticais para a detecção de erros. Imagine que exatamente 4 *bits* sejam invertidos devido a erros de transmissão. Derive uma expressão para a probabilidade de que o erro não seja detectado.
154. Qual é o resto obtido pela divisão de  $x^7 + x^5 + 1$  pelo polinômio gerador  $x^3 + 1$ ?
155. Um fluxo de *bits* 10011101 é transmitido com a utilização do método de CRC padrão descrito no texto. O polinômio gerador é  $x^3 + 1$ . Mostre a cadeia de *bits* real transmitido. Suponha que o terceiro *bit* a partir da esquerda seja invertido durante a transmissão. Mostre que esse erro é detectado na extremidade receptora.
156. Os protocolos de enlace de dados quase sempre colocam o CRC em um final, em vez de inseri-lo no cabeçalho. Por quê?
157. Por causa da condição de que um único sinalizador pode ser usado tanto como sinalizador final quanto como sinalizador inicial, um erro de *bit* único pode causar problemas.
- Explique como um erro de *bit* único pode mesclar dois quadros em um.
  - Explique como um erro de *bit* único pode dividir um único quadro em dois quadros.
158. Sugira melhorias para o algoritmo de enchimento de *bits* para superar os problemas de erros de *bit* único descritos no problema anterior.
159. Usando a sequência de *bits*:

original:                    11111111111101111110111110  
 depois do *bit-stuffing*: 1111101111101101111101011111010

mostre o padrão de sinal na linha usando a codificação NRZ-L. Isso sugere um benefício colateral da inserção de *bits* de *bits*?