PROBLEM 1 ANSWER

The problem is to code a message which is an i.i.d. sequence with values in $\{-1,0,1\}$ into a sequence of zeros and ones as efficiently as possible. The message has $\log_2(3)$ bits per "letter", while the channel (that transmits zeros and ones without error) can transmit only one bit per period. The Shannon bound on the rate of transmission is therefore $\log_2(3)/\log_2(2) = 1.585$ periods per letter of the message.

Students presented two solutions. One is

$$\begin{array}{ccc} -1 & \to & 0 \\ 0 & \to & 10 \\ 1 & \to & 11\,. \end{array}$$

This has an expected number of periods per transmitted character of

$$2 \cdot \frac{2}{3} + \frac{1}{3} = \frac{5}{3} = 1.666667\,.$$

But

$$1.05 * 1.585 = 1.664 < \frac{5}{3}\,.$$

Everyone who put forward this answer nonetheless argued that the letters-per-unit time measure, which is

$$\frac{3}{5} = .6 > .95 \cdot 1/\log_2(3) = .5994$$

satisfies the "within 5% of the Shannon bound" criterion. You got full credit for this, though it is a rather slippery claim.

A safer answer along the same line would code the nine possible two-letter pairs into three or four bit sequences, e.g.

$$\begin{array}{cc} -1,-1 & 000 \\ -1,0 & 001 \\ -1,1 & 010 \\ 0,-1 & 011 \\ 0,0 & 100 \\ 0,1 & 101 \\ 1,-1 & 110 \\ 1,0 & 1110 \\ 1,1 & 1111\,. \end{array}$$

This has expected periods per letter of

$$\frac{1}{2} \cdot \left(\frac{7}{9} \cdot 3 + \frac{2}{9} \cdot 4 = \frac{29}{9} = 1.61111 \right) < 1.05 * 1.585 = 1.664\,,$$

and of course also is well within 5% of the letters per unit time version of the limit also.

Several students also thought of mapping k letters of the original message into a subset of the possible 2^m sequences of zeros and ones, with m chosen as small as possible, to minimize "wasted" sequences of zeros and ones. This approach produces a code that achieves 1.60 periods per letter when 5 letters of the original message are mapped into 8 letters of the telegraph key, so it gets much closer to the theoretical limit. Interestingly, this brute force approach does not get better with larger k until $k = 17$, which gives 1.5882 periods per letter. Note that 5 and 17 are prime numbers. The first value of k that improves on $k = 17$ is 41, also prime. Interesting question: Is every k that improves on all smaller values of k a prime number?