Module Title: Information Theory
Exam Diet: Mock Paper 2011
Brief notes on answers:

1. (a) There is more than one possible Huffman code, as there are some free choices in constructing and labelling the tree. The code must be valid, with the correct code lengths: $\{2, 2, 3, 3, 3, 4, 4\}$. [1 mark]
   i) Expected length: $(2\times{}^1/_4 + 2\times{}^1/_4) + (3\times{}^1/_8 + 3\times{}^1/_8 + 3\times{}^1/_8) + (4\times{}^1/_{16} + 4\times{}^1/_{16}) = 1 + {}^9/_8 + {}^1/_2 = 2 + {}^5/_8$ bits/symbol. [1 mark]
   (Any other way of arranging the sum, perhaps on the Huffman tree is fine.)
   ii) In this ensemble each symbol is a negative integer power of two; every symbol is represented in a symbol with length equal to its information content. Therefore the expected length is the entropy of the ensemble. Shannon's source coding theorem states that *no* method, including arithmetic coding, can compress better than to the entropy of an ensemble in expectation. [1 mark] (As Huffman coding is simpler, it would be preferred.) An extra [1 mark] for getting the question largely correct.

   (b) i) A simple Huffman code does not use the context of a symbol when encoding it, so only the expected numbers of each symbol matter when computing the expected length of an encoding. The expected numbers of symbols when observing a single symbol is given by the marginal probability distribution: the expected encoding length per symbol is the same as in the previous part. [2 marks]
   ii) the entropy of a block of K symbols from the stream can be less than K times the entropy of a single symbol when there are dependencies in the stream. An arithmetic coder can use the context of previous symbols to exploit this dependency and will, in expectation, compress the stream better than a Huffman code. [2 marks]

   (c) The Huffman algorithm constructs a binary tree by repeatedly merging the two nodes with the lowest probability. The new node obtains the sum of the probabilities from the merged nodes, and the algorithm terminates when there is a single node with probability one.
   If a symbol has probability $p(\mathtt{a}) > 0.5$, all other nodes in the tree (except the root node containing it) must have smaller probability because probabilities are non-negative and sum to one. Therefore, the symbol '`a`' will be merged last.
   The Huffman algorithm labels each branch of the binary tree with a single `0` or `1`. Codes are formed by traversing the tree and concatenating the symbols. As the symbol '`a`' was merged last, there is a single branch from the root to the symbol: the symbol is always assigned a codeword of length 1. [3 marks]

   (d) $I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$. [2 marks]
   In the first definition, the mutual information is the average information content contained by the (input) variable X, minus the average uncertainty that remains after observing the (output) variable Y. Thus the mutual information gives the average information communicated over a channel with input X and output Y. [1 mark]

   (e) Arithmetic coders compress to the information content of an input under its model (+ an overhead that can be neglected for long sequences). One can take

the difference of the information contents for the two models, or recall that this has the neat form of the KL-divergence:
$\mathrm{KL}(P\,||\,Q) = (1/3)\log_2 \frac{1/3}{2/3} + (2/3)\log_2 \frac{2/3}{1/3} = -1/3 + 2/3 = 1/3$ bits/symbol.

(f) The correct information content is 2020 bits (3sf), or 2.02×10^3 bits (3sf). [1 mark]
The following code would work more robustly:

```
max_lnp = max(ln_prob1, ln_prob2);
min_lnp = min(ln_prob1, ln_prob2);
ln_prob = max_lnp + log(1 + exp(min_lnp - max_lnp)) + log(0.5);
inf_content = -ln_prob / log(2.0);
```

(Anything reasonable would be accepted.) [2 marks]

(g) In an $[N, K]$ block code, K source symbols are jointly encoded into an encoded block of length N. [1 mark]
The rate of a code can be defined as the average number of bits sent per output symbol sent when choosing the K source symbols uniformly. When the K symbols are binary the rate is $R = K/N$. [1 mark]
(If the K symbols come from a source alphabet $\mathcal{A}_S$, the rate would be multiplied by $\log_2 |\mathcal{A}_S|$. Some texts make the rate relative to that achievable by choosing inputs of the channel uniformly. Any consistent definition would be accepted.)

(h) The $[7, 4]$ Hamming code can correct any one bit error in a block. It can never correct more than one bit error. [1 mark].
Probability that a block is corrupted: $1 - (1-f)^7 - 7f(1-f)^6$
OR Probability that a block is corrupted: $\sum_{k=2}^{7} \binom{7}{k} f^k (1-f)^{7-k}$ [1 mark]
The error probability is dominated by the first term in the sum:
$\binom{7}{2} f^2(1-f)^5 \approx 21f^2 \approx 0.002$ [1 mark]
(I would mark correct a sloppier approximation, like: $(1-f)^m \approx 1-fm \;\Rightarrow\; 1-(1-0.07)-0.07(1-0.06) = 0.07\times0.06 \approx 0.004$ — although 0.002 is much closer to the right answer.)

2. (a) $H(Y) = H_2(f) + (1-f)\,H(X)$. (By entropy decomposition.)

(b) The choices should make a uniform distribution over Y. Therefore,
$f = 1/(M+1)$, [1 mark]
and $p_m = 1/M$ for all $m \in \{1, 2, ..., M\}$. [1 mark]

(c) $H(Y\,|\,X) = H_2(f)$
$I(X;Y) = H(Y) - H(Y\,|\,X) = H_2(f) + (1-f)\,H(X) - H_2(f) = (1-f)\,H(X)$
Maximizing, set $\mathcal{P}_X$ to uniform: $C = (1-f)\,\log_2 M$ bits. [3 marks].
That's the expected answer: fraction f of the transmissions are erased. The remaining $(1-f)$ pass through uncorrupted, giving their full information content to the receiver.

(d) The sender uses an arithmetic coder to represent a message close to optimally using the symbols in $\mathcal{A}_X$. These symbols are sent one at a time over the channel. If the receiver gets a '0' the feedback channel is used to ask the sender to resend the symbol before continuing. [2 marks]
The expected number of times each symbol needs to be sent is: $\sum_{k=0}^{\infty}(1+k)\,f^k(1-f) = 1/(1-f)$ Therefore, the rate is $(1-f)\,\log_2 M$. [1 mark]

(e) This channel is an erasure channel. A digital fountain code, or a low-density parity check code, could be used in practice to communicate at rates close to the capacity. [1 mark]

These codes pseudo-randomly select several source packets and bitwise XOR them together. More packets are sent than original source packets so that information about each source packet will still be available even though some packets are lost. [1 mark]

Decoding would ideally done by inferring the most probable source packets given the received packets. This inference problem appears intractable in general but can be solved approximately using the sum-product algorithm (AKA loopy belief propagation). [1 mark].

(f) One could communicate by sending only even-valued symbols (modulo M), i.e., $\{M{\equiv}0,2,4,6,...\}$. (Perhaps use arithmetic coding to compress a message into this representation.) For each received symbol there will only be one possible source symbol and the message can be uncompressed again.

The source alphabet has been reduced to $\lfloor M/2 \rfloor$ in size, so each channel usage communicates $\log_2 \lfloor M/2 \rfloor$ bits. [3 marks]

(g) When $M = 2$, it isn't possible to communicate unambiguously with a single use of the channel. Instead multiple source symbols must be combined into a block of K symbols and the channel used N times in order to send them. Shannon showed that any finite, but possibly tiny, target error rate is achievable for large enough block-lengths N, while maintaining the rate of communication K/N at the capacity of the channel. [2 marks]

(h) The Nth extension of a channel has an input consisting of N input symbols from the original channel, with an alphabet of size $|\mathcal{A}_X|^N$. Similarly the output alphabet is extended to N output symbols. [1 mark]

The capacity of the Nth extensions is simply NC, where C is the capacity of the original channel. [1 mark]

[Apologies, bad question. I'm not sure what I was asking(!).]

If input symbols are chosen independently from $P(X)$, then:
$P(X_1, X_2, X_3, ...X_N) = \prod_{n=1}^{N} P(X_n)$ (1).
However, in a practical code the inputs will be chosen from a discrete set of hard-to-confuse codewords.

Conditioned on the inputs the outputs are drawn independently from:
$P(Y_1, ..., Y_N \,|\, X_1, ...X_N) = \prod_{n=1}^{N} P(Y_n \,|\, X_n)$ (2)
using the conditional distribution from the original channel. The joint distribution is simply the product of (1) and (2). [1 mark]

(i) $\log_2 S$ bits of information are communicated by S non-confusable codewords. (They will be non-confusable for all practical purposes for large enough N.) The rate of communication is $(\log_2 S)/N = C$. That is, $S = 2^{CN}$. [1 mark]

(j) If codewords are selected randomly they could be on top of each other. The maximal block error could be 1. [1 mark]

The maximal block error can be improved by discarding the most confusable half (say) of the codewords. (This is called expurgation.) [1 mark]

The rate of communication is only changed by $(\log_2 0.5)/N$ bits per channel use, which can be made arbitrarily small by increasing N. [1 mark]