Part A

1. Consider the following four pieces of code. Do they all have the same effect, and if not, why not? [6]

Code Fragment 1 Code Fragment 2

```
for i=1:1000        for i=1000:-1:1
    a(i)=i^2;           a(i)=i^2;
end                 end
```

Code Fragment 3 Code Fragment 4

```
a=[1:1000];         a=[1:1000];
a=a*a;              a=a.*a;
```

A. *1 and 2 have almost the same effects,* `a` *is an array of squares of integers. In 1* `i` *ends up as 1000, while in 2 it end up as 1. 4 is very much the same, but doesn't change* `i` *at all. 3 is an error, as one can't multiply a* 1×1000 *vector by itself.*

From the point of view of efficiency, which code fragments are bad, and why? [4]

A. *1 has the growing array problem, so is quadratic in the length of the array, while 2 and 4 are linear. Hence 1 is definitely the worst (of those that work). I would need to experiment betwen 2 and 4: 4 uses MatLab builtins, but would seem to make two arrays, each of length 1000.*

What other piece of code might you suggest to get the same effect? [2]

A. `a=[1:1000].*[1:1000];`*. As in 4, but MatLab will probably not bother creating the intermediate arrays as not named.*

Consider Code Fragment 5 from `List.m` — what is its O complexity, in both time and space, for deleting k objects from a list of length n? How does this compare with the array-based equivalent? [4]

Code Fragment 5

```
function l2=delk(l,k) % deletes the first k elements
    if (k==0)
        l2=l;
    else
        l2=l.tail.delk(k-1);
    end
end
```

A. *time:* $O(k)$ *and space* $O(1)$ *as nothing is created. The array based equivalents are* $O(n-k)$ *for both time and space (buidling the new structure).*

2. Define the terms *ordered (binary) tree*, *AVL tree* and *(ordered binary) heap*. Your answers should *define* the AVL condition and the heap condition. [8]

A.

- *An ordered binary tree is one where every node in the left subtree has a value $<$ (or $\leq$) the root, every node is the right sub-tree has a value $>$ (or $\geq$) the root, and the left and right sub-trees, if non-empty, satisfy the same condition recursively.*
- *An AVL tree is one satisfying the* AVL condition*: at each node, the absolute value of the balance, the difference in height betwene the left and right subtrees, is at most one.*
- *A heap is a binary tree satisfying the heap condition: the value at every node is $\leq$ the value at the left and right children (and therefore $\leq$ the values in all descendants).*

Explain how an array of items can be converted into a heap, without needing any extra storage for pointers etc. [12]

A. *Regard the children of* `a(i)` *as* `a(2*i)` *and* `a(2*i+1)` *. Then we make this satisfy the heap condition by the "sift up" process: every time the heap condition is locally violated, we swap* `a(i)` *with the greater of* `a(2*i)` *and* `a(2*i+1)`*. [bookwork, but not in the book!]*

Part B

3. In this question $\left(\dfrac{a}{p}\right)$ denotes the Legendre symbol. If $b = p_1 \times p_2 \times \ldots \times p_n$, where each p_i is a prime (possibly not all different), and $\mathrm{hcf}(a,b) = 1$, then $\left[\dfrac{a}{b}\right]$ denotes the Jacobi symbol, which is defined by

$$\left[\frac{a}{b}\right] = \left(\frac{a}{p_1}\right) \times \left(\frac{a}{p_2}\right) \times \cdots \times \left(\frac{a}{p_n}\right).$$

You should notice in particular that $\left[\dfrac{a}{p}\right] = \left(\dfrac{a}{p}\right)$ if p is prime, and that 1109 is prime.

(a) Define the Legendre symbol $\left(\dfrac{a}{p}\right)$, explaining carefully what the allowable values of a and p are. [2]

A. $\left(\dfrac{a}{p}\right)$ *is defined for* p *an odd prime and* $a \in \mathbb{Z}$ *with* p *not dividing* a. *It is* 1 *if* a *is a square mod* p *and* -1 *if not.*

(b) Show that 11 is a square mod 7 but 7 is not a square mod 11. Write down the rule obeyed by Legendre symbols of which this is a case. [3]

A. $11 = 4$ *is a square mod* 7. *The non-zero squares mod* 11 *are* 1, 4, 9, $16 = 5$ *and* $25 = 3$, *which do not include* 7. *This is a case of quadratic reciprocity,* $\left(\dfrac{p}{q}\right) = \left(\dfrac{q}{p}\right)$ *unless* $p = q = -1$ *mod* 4.

(c) Using any rules for Legendre symbols that you know, calculate the Legendre symbol $\left(\dfrac{611}{1109}\right)$. You may also use the fact that $611 = 13 \times 47$. [7]

A. $\left(\dfrac{611}{1109}\right) = \left(\dfrac{13}{1109}\right)\left(\dfrac{47}{1109}\right) = \left(\dfrac{1109}{13}\right)\left(\dfrac{1109}{47}\right)$ *since* $1109 = 1$ *mod* 4. *But* $1109 = 4$ *mod* 13 *so* $\left(\dfrac{1109}{13}\right) = 1$, *and* $1109 = 28$ *mod* 47, *so* $\left(\dfrac{1109}{47}\right) = \left(\dfrac{28}{47}\right) = \left(\dfrac{4}{47}\right)\left(\dfrac{7}{47}\right) = \left(\dfrac{7}{47}\right)$. *Using QR again, that's* $-\left(\dfrac{47}{7}\right) = -\left(\dfrac{5}{7}\right) = 1$ *(either use QR yet again or notice that the squares mod* 7 *are* 1, 4 *and* 2*) so the answer is* 1.

Question 3 continues on next page ...

(d) Show, by induction on n or otherwise, that $\left[\dfrac{-1}{b}\right] = 1$ if $b = 1 \bmod 4$ and -1 if $b = -1 \bmod 4$. You may assume that $\left(\dfrac{-1}{p}\right) = 1$ if $p = 1 \bmod 4$ and -1 if $p = -1 \bmod 4$. [7]

A. *For* $n = 1$ *there is nothing to prove. Otherwise, put* $b' = p_1 \ldots p_{n-1}$. *Then* $\left[\dfrac{a}{b}\right] = \left[\dfrac{a}{b'}\right]\left[\dfrac{a}{p_n}\right]$ *so* $\left[\dfrac{-1}{b}\right] = 1$ *iff* b' *(by induction) and* p_n *are both* 1 *or both* -1 *mod* 4, *i.e. if* $b = 1$ *mod* 4.

(e) In fact the Jacobi symbol obeys the same rules as the Legendre symbol, including quadratic reciprocity as long as a and b are odd. Using these rules, calculate $\left(\dfrac{611}{1109}\right)$ again. You may *not* use the fact that $611 = 13 \times 47$, or do any other factorising except for dividing by 2. [4]

A. $\left(\dfrac{611}{1109}\right) = \left[\dfrac{611}{1109}\right] = \left[\dfrac{1109}{611}\right] = \left[\dfrac{498}{611}\right]$. *But this is* $\left[\dfrac{2}{611}\right]\left[\dfrac{249}{611}\right]$ *and* $\left[\dfrac{2}{611}\right] = -1$ *because* $611 = 3$ *mod* 8. *Also* $\left[\dfrac{249}{611}\right] = \left[\dfrac{611}{249}\right] = \left[\dfrac{113}{249}\right] = \left[\dfrac{249}{113}\right] = \left[\dfrac{23}{113}\right]$, *and that is* $\left[\dfrac{113}{23}\right] = \left[\dfrac{-2}{23}\right] = \left[\dfrac{-1}{23}\right]\left[\dfrac{2}{23}\right] = (-1)\times 1 = -1$. *So the answer is* $(-1)\times(-1) = 1$.

4. (a) Explain briefly how to use Diffie-Hellman to send a message from Alice to Bob. You may assume that the message comes from a subgroup G of $\mathbb{F}^*$ for some finite field $\mathbb{F}$, and that the encryption method is exponentiation. [4]

A. *Alice chooses an integer a coprime to $N = |G|$ and uses Euclid's algorithm to compute an inverse a' to a mod N. Bob does the same with b and b'. Then Alice chooses a message $m \in G$ and sends m^a to Bob. He replies with $(m^a)^b = m^{ab}$, and she computes $(m^{ab})^{a'} = m^b$. She sends that back to Bob, who computes $(m^b)^{b'} = m$.*

(b) How would you modify your use of the system so as to create a shared secret between Alice and Bob, rather than sending a message? [2]

A. *They choose $m \in G$ at random, but publicly. Then Alice sends m^a to Bob, and Bob sends m^b to Alice. Now Alice can compute $(m^b)^a = m^{ab}$ and Bob can compute $(m^a)^b = m^{ab}$, so their shared secret is m^{ab}.*

(c) Now suppose that $\mathbb{F} = \mathbb{F}_{83}$ and $G = (\mathbb{F}_{83}^*)^2$. Suppose that Alice's encryption key is $a = 7$ and Bob's encryption key is $b = 13$. Alice wants to send the message $m = 17$. Illustrate your answer to part (a) by carrying out all the computations she and Bob must do, including full details. [13]

A. *Here $N = 41$. So $a' = 6$ because $7 \times 6 = 42 = 1$ mod 41 and $b' = 19$ (probably by Euclid's algorithm, but I observe that $3 \times 13 = -2$ so $b' = -3 \times 2^{-1} = -3 \times 21 = -63 = -22 = 19$ mod 41). Now Alice computes 17^7, by calculating $17^2 = 289 = 40$ and $17^4 = 1600 = -60 = 23$ so $17^7 = 17^4 \times 17^2 \times 17 = 40 \times 23 \times 17 = 920 \times 17 = 7 \times 17 = 119 = 36$; Bob computes 36^{13} by computing $36^2 = 1296 = 51 = -32$, $13^4 = 32^2 = 1024 = 28$ and $13^8 = 28^2 = -46 = 37$ so $36^{13} = 37 \times 28 \times 36 = 4 \times 28 = 112 = 29$; Alice computes 29^6 by computing $29^2 = 841 = 11$ and $29^4 = 11^2 = 121 = 38$ so $29^6 = 11 \times 38 = 418 = 3$, and finally Bob computes 3^{19} by $3^2 = 9$, $3^4 = 9^2 = 81 = -2$, $3^8 = (-2)^2 = 4$, $3^{16} = 4^2 = 16$, so $3^{19} = 16 \times 9 \times 3 = 144 \times 3 = -22 \times 3 = -66 = 17$.*

(d) Suppose that Alice and Bob are using a system as in part (b) only for creating shared secrets, never for sending messages. Bob notices that in this case he never actually does any decryption, so he thinks it is harmless to tell Fred his decryption key as long as he keeps b secret. Is he right? Explain why, or show how Eve could recover the secret if she knew how to decrypt messages sent by Bob. [2]

A. *This is disastrous, because now Eve knows b' and she computes an inverse to b' mod N, which is b. Then when she sees Alice send m^a to Bob, she simply computes m^{ab}, just as Bob did.*

(e) Alice and Bob have created a shared secret, but Alice has told Fred what it is. Realising what she has done, she contacts Bob and they set up a new shared secret using the same system. Is this all right? Explain why, or show how Eve could recover the new secret. [2]

A. *This is all right. Eve now knows m, m^a, m^b and m^{ab} but she can't find Alice's key from that: if she could, so could Bob, because he knows all that too. So they can safely start again with a different m' – they may keep a and b.*

5. (a) Say what is meant by the *Fourier matrix* F_m. What is the inverse matrix F_m^{-1}? Justify your answer. [4]

A. *Let* $\zeta = e^{2\pi i/m}$ *be a primitive mth root of unity. Then* $(F_m)_{ij} = \zeta^{ij}$*, where* $0 \leq i, j < m$*. The inverse matrix is* G_m *where* $(G_m)_{ij} = \frac{1}{m}\zeta^{-ij}$*. The reason is that*

$$(F_m G_m)_{ik} = \frac{1}{m}\sum_{j=0}^{m-1} \zeta^{ij}\zeta^{-jk}$$

which is 1 *if* $i = k$ *because all the terms in the sum are* 1*, and is* 0 *if* $i \neq k$ *because* $\left(\sum_{j=0}^{m-1} \zeta^{(i-k)j}\right)(1 - \zeta^{i-k}) = 1 - \zeta^{(i-k)m} = 0$.

(b) Explain how to compute $F_m\,{}^t\mathbf{c}$ in time $O(m \log m)$, where $\mathbf{c} = (c_0, \ldots, c_{m-1})$ and ${}^t\mathbf{c}$ is the corresponding column vector. You may assume that m is a power of 2 if you wish. You should explain the process in your own words, and explain why it is quick, but you are *not* required to provide a proof that the algorithm does have complexity $O(m \log m)$. [6]

A. *Assume that* $m = 2n$ *is even and put* $\mathbf{y} = F_m\,{}^t\mathbf{c}$*. Then*

$$y_i = \sum_{j=0}^{2n-1} (F_{2n})_{ij} c_j = \sum_{j=0}^{2n-1} \zeta^{ij} c_j.$$

Splitting this into even and odd parts gives

$$y_i = \sum_{j=0}^{n-1} \zeta^{2ij} c_{2j} + \zeta^i \sum_{j=0}^{2n-1} \zeta^{2ij} c_{2j+1}.$$

Writing $\mathbf{c}^+$ *for the even subvector* $(c_0, c_2, \ldots)$ *and* $\mathbf{c}^-$ *for the odd subvector, and reading the lower indices mod n we get*

$$y_i = \sum_{j=0}^{n-1} \zeta^{ij} c_j^+ + \zeta^i \sum_{j=0}^{2n-1} \zeta^{2ij} c_j^- = (F_n\,{}^t\mathbf{c}_j^+)_i + \zeta^i (F_n\,{}^t\mathbf{c}_j^-)_i.$$

This allows us to do two F_n *calculations instead of one* F_{2n} *calculation, which means that we can do an* F_{2^r} *calculation by doing about r calculations for* F_2*, which is very quick.*

Question 5 continues on next page . . .

(c) Illustrate your answer to part (b) by computing $(F_8\,{}^t\mathbf{c})_5$, where $\mathbf{c} = (1, 0, 1, -1, 0, i, i, -1) \in \mathbb{C}^8$. [Do not forget that the indexing starts from 0, not 1.] [9]

A. *The algorithm tells us that* $x = (F_8\,{}^t\mathbf{c})_5 = (F_4\,{}^t(1, 1, 0, i))_5 + \zeta^5(F_4\,{}^t(0, -1, i, -1))_5$. *The* 5 *on the bottom is to be read as* 1 *mod* 4 *and* $\zeta^5 = -e^{\pi i/4}$. *So we do the same thing again and get*

$$x = (F_2\,{}^t(1, 0))_1 + i(F_2\,{}^t(1, i))_1 - e^{\pi i/4}\big((F_2\,{}^t(0, 1))_1 + i(F_2\,{}^t(-1, -1))_1\big).$$

Using $F_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ *this works out as* $x = 1+i(i-1)-e^{\pi i/4}(-i+0) = 2+i-ie^{\pi i/4}$

(d) Write down a diagonal matrix Δ_4 and a matrix P with only one non-zero entry in each row and column, such that

$$F_8 = \begin{pmatrix} I & \Delta_4 \\ I & -\Delta_4 \end{pmatrix} \begin{pmatrix} F_4 & 0 \\ 0 & F_4 \end{pmatrix} P.$$

[4]

A. Δ_4 *has entries* $1, \zeta, \zeta^2, \zeta^3$ *and* P *has a* 1 *in the* $(j, 2j)$ *places* $(0 \le j < 4)$ *and the* $(j, 2j-7)$ *places* $(j > 3)$.