Midterm 1

7:00-9:00pm, 8 October

Notes: There are five questions on this midterm. Answer each question part in the space below it, using the back of the sheet to continue your answer if necessary. If you need more space, use the blank sheet at the end. In both cases, be sure to clearly label your answers! None of the questions requires a very long answer, so avoid writing too much! Unclear or long-winded solutions may be penalized. The approximate credit for each question part is shown in the margin (total 100 points). Points are not necessarily an indication of difficulty!

<table>
<thead>
<tr>
<th>Q1</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q2</td>
<td>20</td>
</tr>
<tr>
<td>Q3</td>
<td>20</td>
</tr>
<tr>
<td>Q4</td>
<td>14</td>
</tr>
<tr>
<td>Q5</td>
<td>15 + 15</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
</tr>
</tbody>
</table>

For official use; please do not write below this line! For official use; please do not write below this line!
1. Propositional Logic [16 pts]

A. (8 pts - 2 pts each) State whether the following equivalences are valid or invalid. There is no need to justify your answers. Guess at your own risk - wrong answers will be awarded negative credit.

I. \( \neg \forall n [(P(n) \land Q(n)) \Rightarrow \neg R(n)] \equiv \exists n [P(n) \land Q(n) \land R(n)] \)

II. \( \forall m \exists n [\forall l (A(m, l) \land B(n, l)) \Rightarrow C(m, n)] \equiv \forall m \exists n [\neg C(m, n) \Rightarrow \exists l (\neg A(m, l) \lor \neg B(n, l))] \)

III. \( \forall m \forall n [P(m) \Rightarrow Q(n)] \equiv \forall n \forall m [Q(n) \Rightarrow P(m)] \)

IV. \( \neg \exists l \forall n [(P(m) \land Q(l)) \lor R(m, n, l)] \equiv \exists l \forall m \exists n [(\neg P(m) \land \neg Q(l)) \lor \neg R(m, n, l)] \)

B. (8 pts - 2 pts each) For nonnegative integers \( x \) and \( y \), let \( P(x, y) \) be the proposition that “\( x + y > xy \)”.

Which of the following statements are true? Give a one line proof or a counterexample.

I. \( \forall x \exists y P(x, y) \)
II. $\exists x \exists y P(x, y)$

III. $\exists x \forall y P(x, y)$

IV. $\forall x \forall y P(x, y)$
2. [Proofs.] [20 pts]

A. (10 pts) Let \( D_n \) be the number of ways to tile a \( 2 \times n \) checkerboard with dominos, where a domino is a \( 1 \times 2 \) piece. Prove that \( D_n \leq 2^n \) for all positive integers \( n \). (Find a recurrence relation for \( D_n \). No need to give a proof. Then inductively prove the upper bound on \( D_n \).)

Note that dominos can only be placed exactly aligned with checkerboard squares, and cannot be placed diagonally.

B. (10 pts) Show that \( \forall \) odd \( a \in \mathbb{N}, a^2 = 1 \mod 8 \).
3. [RSA] [20 pts]

A. (10 pts) $e = 7, p = 7, q = 11$ Find $d$.

Solution: Remember that from the way RSA is setup that $d = e^{-1} \mod (p-1)(q-1)$. So we are looking for $d = 7^{-1} \mod 60$. In order to find this, we use the extended GCD algorithm with inputs 60 and 7.

egcd(60, 7)

egcd(7, 4)

egcd(4, 3)

egcd(3, 1)

egcd(1, 0)

return (1, 1, 0)

return (1, 0, 1)

return (1, 1, 0 - (4 \div 3) \times 1) = (1, 1, -1)

return (1, -1, 1 - (7 \div 4) \times (-1)) = (1, -1, 2)

return (1, 2, -1 - (60 \div 7) \times 2) = (1, 2, -17)

We can read off from here that $d = -17 = 43 \mod 60$.

B. (5 pts) With RSA Amazon can sign a message as follows; For a system with public key $(N, e)$ and secret key $d$, Amazon sends the message $(x, x^d \mod N)$. If Bob gets $(x, y)$, how can he verify that $y = x^d \mod N$? (Bob does not know $d$ and the answer is very brief.)

The answer that we were looking for is very simple. If Bob receives $(x, y)$ and knows $(N, e)$, one way that he could verify that $y = x^d$ is simply to encrypt the message $y$. If $y = x^d$, we get:

$E(x^d) = (x^d) \mod N = x^{de} \mod N = x^{1 + k(p-1)(q-1)} \mod N = x \mod N$

This was shown all with properties that we learned from RSA. So, if $E(y) = x$, then we know that the message $(x, y)$ was sent by Amazon.

C. (5 pts) Use the fact that $a^{p-1} = 1 \mod p$ for prime $p$ and $a$ relatively prime to $p$ to prove that $a^{(p-1)(q-1)} = 1 \mod pq$ for primes $p$ and $q$ and $a$ relatively prime to $p$ and $q$. 
4. [Stable Marriage] [14 pts]

A. (8 pts) Consider an instance of the Stable Marriage problem in which the men are \{1, 2, 3, 4\}, the women are \{A, B, C, D\}, and the preference lists are

<table>
<thead>
<tr>
<th>Men (1-4)</th>
<th>Women (A-D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: A B D C</td>
<td>A: 2 3 4 1</td>
</tr>
<tr>
<td>2: C B A D</td>
<td>B: 1 4 2 3</td>
</tr>
<tr>
<td>3: D C B A</td>
<td>C: 1 4 2 3</td>
</tr>
<tr>
<td>4: D C A B</td>
<td>D: 1 3 2 4</td>
</tr>
</tbody>
</table>

Use the traditional marriage algorithm to find the male-optimal pairing.

B. (3 pts) Given \(n\) men and \(n\) women, what is the minimum number of stable pairings that must exist for any set of preferences? Justify your answer by describing an instance.

C. (3 pts) We saw in the homework that it was possible for a pairing to be stable even if there was a pair \((M, W)\) such that \(M\) was \(W\)'s least favorite man and \(W\) was \(M\)'s least favorite woman. What is the maximum number of couples with this property (each member is paired with their least favored partner) can there be in any stable pairing? Justify your answer.
5. **[Codes]** [30 pts]

A. (15 pts) Your friend sends you a message in the alphabet R = 0, F = 1, A = 2, U = 3, and N = 4 using the polynomial scheme discussed in class. Assume that a polynomial $P(\cdot)$ over $GF(q)$ is used, for the smallest value of $q$ that will accommodate the given alphabet. The message size is 3. Four packets are sent where packet $i$ (starting from 0) corresponded to $P(i)$. You receive the following packets.

- F
- U
- clearly corrupted
- N

Assuming the three decipherable packets arrive uncorrupted, what is the value in the corrupted packet? Justify your answer.
B. Say another message is sent using five packets and you receive packets F, U, N, U, and R, one of which is wrong.

I. (7 pts) The original message is either “FUN” or “RUN”. Which is it? Why? (Hint: try one.)

Since we already know what the encoding polynomial looks like if the original message was FUN, let’s start by assuming that the original message was FUN. This means that \( P(x) = 2x^2 + 1 \), and we know that there is only one corrupted packet. Since the original message was FUN, this means that the first three packets were sent through uncorrupted. The fourth packet is corrupted - it should have been \( P(3) = 4 = N \), but we received a U instead. However, the fifth packet is also corrupted - it should have been \( P(4) = 3 = U \), but we received an R instead. Therefore, the original message could not have been FUN, and thus the original message must have been RUN.

II. (4 pts) Recall that in the Berlekamp-Welch algorithm, one can set up a set of linear equations and use the solution to reconstruct the original polynomial. How many unknowns and equations do you have in the Berlekamp-Welch system for this situation?

Recall that in Berlekamp-Welch we are trying to solve for the coefficients of \( Q(x) \) and \( E(x) \).

\( Q(x) \) is an \( n - 1 + k \) degree polynomial - in this case it would be a \( 3 - 1 + 1 = 3 \) degree polynomial, and thus it has \( 4 \) unknown coefficients.

\( E(x) \) is a degree \( k \) polynomial, but we already know that its highest order coefficient is \( 1 \). In this case it is a degree \( 1 \) polynomial, and thus it has \( 1 \) unknown coefficient. Thus there are \( 5 \) unknowns and equations total.

III. (4 pts) Write out the equations that correspond to the first two received characters: i.e., \( R(0) \) and \( R(1) \). Denote the coefficients of \( Q(x) \) using \( a_i \) and the coefficients of \( E(x) \) by \( b_i \).