Homework 5: Number Theory

Due date: Wednesday November 5th at 11:59 PM

If you work with others (and you should!), remember to follow the collaboration policy outlined in the syllabus.
In general, you are graded on both the clarity and accuracy of your work. Your solution should be clear enough that someone in the class who had not seen the problem before would understand it.
We sometimes describe approximately how long our explanations are. These are intended to help you understand approximately how much detail we are expecting. You are allowed to have longer explanations, but explanations significantly longer than necessary may receive deductions.

Be sure to read the grading guidelines on the assignments page for more information on what we're looking for.

In order to assist with the transition from formal proofs to English proofs, we've published a style guide on the website containing some tips. This guide contains references to proof materials that we haven't taught yet, so don't worry if some of these terms are unfamiliar.

For this and all future homework assignments, proofs must be English proofs unless we explicitly say otherwise.

1. Backwards Proofs [6 points]

A common error now that we're doing a lot of algebra is to write a "backwards" or "U-shaped" proof. For a proof to be valid, we must start from facts we know (either givens, or accepted facts, or supposing hypotheses to prove implications), and derive from them the statement we desire.
We cannot start from the equation to be shown and simplify it to something "obviously true."

Read the reading about backwards proofs. Then do these problems:

(a) Complete the Backwards Proof Practice assignment on Gradescope. This part will feel like a concept check (explanations appear when correct, problem is graded automatically, etc.) but counts in the homework category for grades. [4 points]

(b) Revisit the first proof in the Gradescope assignment (trying to prove $\frac{x^2+x-6}{x-2} = x+3$). Help correct your friend's first two steps of the proof if it was wrong, or finish it up for them if it was right! [2 points]

2. Let's Settle Our Differences [8 points]

After hours of trick-or-treating, you and your three roommates have amassed quite the pile of candy. You've separated the candy into three sets: A, B, and C. Roommate one (we'll call them X) says that they want to eat the following set of candy: $(A \cup C) \setminus (B \cup C)$. Your second roommate (we'll call them Y) says they want to eat $(A \cap \overline{B} \cap \overline{C})$. Your last roommate (who already took 311) suggests they can't both get what they want, because X's set is a subset of Y's set. However, you know Y doesn't have the patience to read a full inference proof.

Your job: Suppose A, B, and C are sets. Prove in English $(A \cup C) \setminus (B \cup C) \subseteq (A \cap \overline{B} \cap \overline{C})$ **with an English proof**. In writing your proof, be sure to only apply the definition of one set operation in a single step; since this isn't an inference proof, you can combine multiple logical equivalences together in a single step (if you want to use them), but remember that your reader should easily follow your logic.

3. I'm almost (un)done with this problem... [22 points]

In normal arithmetic, $a+12+(-12) = a$ for every integer a. So we say that -12 "undoes" 12. In modular arithmetic, a similar statement might be that $a + 12 + 2 \equiv a \pmod{14}$, so 2 "undoes 12 for $\pmod{14}$ addition." More generally,

given an integer n, we say that an integer b "undoes 12 for $\pmod n$ addition" if and only if for all integers a, $a + 12 + b \equiv a \pmod n$.

In this problem, you will show that for every integer n (where $n > 12$), there exists some integer b, where $1 \leq b \leq n$, which undoes 12 for $\pmod n$ addition.

(a) Write the statement "for every integer n (where $n > 12$), there exists some integer b, where $1 \leq b \leq n$, which undoes 12 for $\pmod n$ addition." in predicate logic. You should use the predicate "$\texttt{Undoes12}(b, n)$ to say "b undoes 12 in $\pmod n$ arithmetic". [2 points]

(b) You should (hopefully) have a statement that starts with $\forall n \exists b$. If not, make sure to double check before proceeding! Recall that since the $\exists$ come second, the value of b is allowed to depend on n. Give a formula for the b (in range $1 \leq b \leq n$) for which $\texttt{Undoes12}(b, n)$ evaluates to true. The formula will depend on n. *Hint: this formula should be very simple.* [2 points]

(c) Now do the actual proof. You'll start the proof by introducing an arbitrary variable (you're proving a $\forall$) then you'll be doing an exists proof (tell us what value of b you want and argue that it makes $\texttt{Undoes12}()$ evaluate to true). Be sure you don't do a backwards proof! **For this part, you may not use facts about modular equivalences**, you may only use the definitions of divides and equivalence mod n and algebra (applied to equations or individual numbers, not to equivalences). [8 points]
Hint: Don't forget that the definition of $\texttt{Undoes12}$ has another $\forall$ quantifier inside of it!

Now that we've shown there is a way to undo 12, next we're going to try to show there's not a bunch of different ways. In this problem, you'll show that for every integer n (where $n > 12$), for all integers b, b' where both b and b' undo 12 for $\pmod n$ addition, that $b \equiv b' \pmod n$. Note that we've gotten rid of the $1 \leq b \leq n$ requirement in this part!

(d) Write the statement above in predicate logic. Use the predicate $\texttt{Undoes12}(b, n)$ for "b undoes 12 for $\pmod n$ arithmetic." [2 points]

(e) Now write an English proof of the statement. For this part you may use theorems shown in class and on the Number Theory Reference Sheet. [8 points]
You may also use the following fact:

Theorem 1 (Transitivity of Equivalence)**.** *For all integers a, b, c, n with $n > 0$:*
if $a \equiv b \pmod n$ and $b \equiv c \pmod n$ then $a \equiv c \pmod n$.

(f) For similar concepts in modular arithmetic, people will say things like "There is a unique number that undoes 12 $\pmod n$." Ponder why this use of "unique" makes sense, but also why this is a little different from the example of "unique" we saw in class. You do not have to write anything for this part [0 points]

4. The power of InDUCKtion [20 points]

You wake up on day zero to find one duck in your yard. You shoo it off to protect your pet snails, but to your dismay, the next morning (the morning of day one) there are three ducks in your yard, and then on day two, there are five ducks in your yard. Being a theorist, you quickly spot the pattern: on day i, there will be $2i + 1$ ducks in your yard. Also being a theorist, instead of doing anything to prevent further duck invasions, you decide to use the power of induction to prove a formula for the total number of ducks that will have visited your yard by day n. Counting each time a duck visits as a new visit (regardless of whether this duck has visited on previous days), you think that the correct formula is $(n + 1)^2$ total visits by day n.

Use induction on n to show that for all integers $n \geq 0$:

$$\sum_{i=0}^{n} (2i + 1) = (n + 1)^2$$

5. A Proof By Contradiction [11 points]

In this problem, we'll examine the following claim.
Claim: For all integers n, if $n \equiv 3 \pmod 4$, then $n \not\equiv 4 \pmod 6$

Parts (a) and (b) will help you figure out what the skeleton of a proof by contradiction would look like. Part (c) is doing the actual English proof.

(a) Write the predicate logic translation of the Claim using the predicate `Congruent`:
`Congruent`(x, y, z) is true if and only if $x \equiv y \pmod z$
Hint: your answer should use a $\forall$ quantifier!

(b) Find the negation of the claim. Give your answer in predicate logic notation. (Don't forget to include the quantifier!)

(c) Complete a proof by contradiction for the claim: "For all integers n, if $n \equiv 3 \pmod 4$, then $n \not\equiv 4 \pmod 6$"

Hint: You may use without proof that a number cannot be both even and odd.

6. Counterexamples Galore [6 points]

In this problem you will use proof by counterexample to disprove claims in a wider set of problems than what we've seen in class. For each part, provide a counterexample that disproves the given claim. Remember to provide one counter-example, not a class of them.

(a) **Buggy Algorithm**:

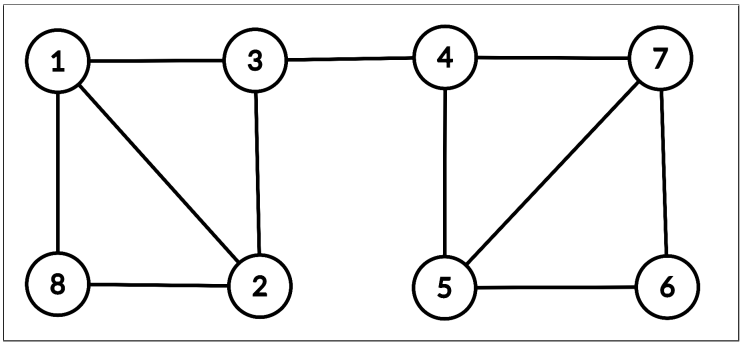
Your friend claims "For every non-empty linked list whose nodes have distinct integer values, the function below will return `true` if the target value exists in the list, and `false` otherwise." Disprove this claim with a counter-example.

For your counterexample, provide a LinkedList and a target value. Briefly explain why the code fails on your example.

```
public boolean contains(ListNode head, int target) {
    ListNode curr = head;
    while(curr.next != null) {
        if (curr.val == target) {
            return true;
        }
        curr = curr.next;
    }
    return false;
}
```

(b) *You may have seen graphs like the one below in an introductory programming course. In case you haven't: we call the circles in the graph "vertices" and the lines connecting two vertices "edges". No further understanding of graphs is required to complete this problem.*

Husky Edge Coloring: Your friend claims "There exists no way to color the vertices of the graph below using colors purple, gold, and pink such that no two adjacent vertices (vertices connected by an edge) share the same color." For your counterexample, provide a coloring of the vertices of the graph. For this part, you do not have to explain your counter-example.



7. Feedback [1 point]

Answer these questions on the separate gradescope box for this question.

Please keep track of how much time you spend on this homework and answer the following questions. This can help us calibrate future assignments and future iterations of the course, and can help you identify which areas are most challenging for you.

- How many hours did you spend working on this assignment (excluding any extra credit questions, if applicable)? Report your estimate to the nearest hour.
- Which problem did you spend the most time on?
- Any other feedback for us?

8. Extra Credit: Exponentially increasing fun [0 points]

You will submit this question to the separate gradescope box for "homework 5 extra credit."

Since $a \equiv a \ \% \ n \pmod{n}$, we know that we can reduce the base of an exponent in $\pmod{n}$ arithmetic. That is:

$$a^k \equiv (a \ \% \ n)^k \pmod{n}.$$

But the same is **not** true of the exponent! That is, we cannot say that $a^k \equiv a^{(k \ \% \ n)} \pmod{n}$. Consider, for instance, that $2^{10} \equiv 1 \pmod{3}$ but $2^{(10 \ \% \ 3)} \equiv 2 \pmod{3}$. The correct way to simplify exponents is quite a bit more subtle. In this problem you'll prove it in steps.

For these proofs you may use any theorem on the number theory reference sheet, even the ones we haven't proven yet in class.

(a) Let $R = \{t \in \mathbb{Z} : 1 \leq t \leq n-1 \wedge \gcd(t, n) = 1\}$. Define the set $aR = \{(ax) \ \% \ n : x \in R\}$. Prove that $aR = R$ for every integer $a > 0$ with $\gcd(a, n) = 1$.

(b) Consider the product of all elements in R (taken $\pmod{n}$) and consider the product of all the elements in aR (again, taken $\pmod{n}$). By comparing these two expressions, conclude that for all $a \in R$ we have $a^{\varphi(n)} \equiv 1 \pmod{n}$ where $\varphi(n) = |R|$.

(c) Use the previous part to show that for any $b \geq 0$ and $a \in R$ we have $a^b \equiv a^{(b \,\%\, \varphi(n))} \pmod{n}$.

(d) Now suppose that $y = x^e \,\%\, n$ for some x with $\gcd(x, n) = 1$ and e some integer ≥ 0 such that $\gcd(e, \varphi(n)) = 1$. Let $d = e^{-1} \,\%\, \varphi(n)$. Prove that $y^d \equiv x \pmod{n}$.

(e) Prove the following two facts about φ: First, if p is prime then $\varphi(p) = p - 1$. Second, for any positive integers a and b with $\gcd(a, b) = 1$, we have $\varphi(ab) = \varphi(a)\varphi(b)$.

These facts together are the basis for the most-widely used "public key encryption system." One chooses $n = pq$ for large primes p and q, and a value of e. The numbers n and e are made public to anyone who wants to send a message securely. To send a message x, the sender computes $y = x^e \,\%\, n$ and sends y (the "encrypted text"). To decrypt, one computes $y^d \,\%\, n$ (note that the recipient must be the one who chose p, q so they can calculate d). The security of the system relies on it being hard to compute d from just e and m.