CSE 20, Discrete Mathematics, Fall 1995
Functions, the Pigeon-hole Principle and Partial Orders
Homework 4, Due Wednesday November 29

Full Proofs (15pts. each):

Palindromes Call a binary string $w = w_1..w_k \in \{0,1\}^k$ a *palindrome* if $w_1..w_k = w_k..w_1$. Prove that there are exactly 2^k palindromes of length $2k$. (Examples: 01110, 1001, 01010, 011110 are all palindromes (because they stay the same if you reverse them; 0111 ,1010, 00001, 0010 are not palindromes. Over the Roman alphabet, madaminedenimadam and ablewasiereisawelba are famous palidromes.) (Hint: there are exactly 2^k words in $\{0,1\}^k$. Give a one-to-one, onto correspondence between the words of length k and the palindromes of length $2k$.)

Sum Prove that for any 3 integers, there must be two whose sum is even.

Let p be a prime number $p \geq 3$. Say that $0 \leq a \leq p-1$ is a quardratic residue modulo p if and only if there is an integer x so that $x^2 mod p = a$. How many quardratic residues modulo p are there? Prove your answer. (Hint: use the problem about squaring mod a prime from the last homework.)

Decimal Prove that for any number N, there is a number divisible by N whose decimal representation contains only 3's and 0's. (Hint: look at $3 mod N$, $33 mod N$, $333 mod N$, etc.)

Program Say that a program P has l lines. Give an informal proof that if P goes for more than l steps without changing the value of a variable, then P is in an infinite loop, i.e., P will never halt. Assume that P never increases the value of any of its variables, all of which are input variables. Describe how to modify P to get an equivalent program P' that always eventually halts on every input; if P halts, P' should have the same output, if P does not halt, P' should output an error message.

(Proofs for this problem can be slightly informal, but still be precise.)

Extra Credit:

1. Consider the following pseudo-program. Write a much simpler program that has the same Input- Output behaviour. Prove that the two programs compute the same function.

   Program: Confuse
   Input: X
   0. LET Y =0;
   1. IF X MOD 3 =2 THEN GOTO 4;
   2. LET Y = Y +1;
   3. IF X MOD 2 =1 THEN GOTO 5;

4. LET X = 2*X +4;
5. IF X MOD 3 =1 THEN GOTO 7;
6. LET Y =Y+1;
7. LET X = X +1;
8. IF X MOD 2 =1 THEN GOTO 4;
9. OUTPUT Y;
10. HALT.

(3 points extra credit)

2. You open a new deck of cards, which come sorted (lowest to highest in each suit, suits ordered clubs, diamonds, hearts, spades.) You repeat the following:

   - Step 1: Repeat 7 times. Take the top card off the deck and put it on the bottom of the deck.
   - Step 2: Divide the deck into two equal size halves, the top 26 cards and the bottom 26 cards. Make a third pile by repeatedly placing the top cards from the old piles onto the new pile, always starting with the pile from the top half of the deck. Do this until all the cards are in the new pile.
   - Step 3: Flip through the cards from the top, until you find a club. Mark this position. Flip through the cards from the bottom until you find a club. Switch the two clubs, leaving all other cards in the same order they were in.

   I claim that if you repeat steps 1-3 above indefinitely, the cards would eventually be in sorted order again. However, I've never done it and in fact have no idea how long it would take, and it is conceivable that you would die of old age or at least boredom before it happens. (So you probably don't want to just do it and keep track of what happens.) How do I know? (5 points extra credit.)