Lauro Cabral

February 7, 2018

CECS 229

**HW1**

*Section 4.1*

5) Show that if *a | b* and *b | a*, where *a* and *b* are integers, then *a* = *b* or *a* = *-b*.

a|b ⬄ b = ak0 b|a ⬄ a =bk1

we can combine these two equations and we get:

b = (bk0)k2

b = bk , *k* = 1 because b = b(1)

since *k = 1*, we get:

b = a(1) a=b(1)

b = a a=b

8) Prove or disprove that if *a | bc*, where a, b, and c are positive integers and a ≠ 0 , then ++*a | b* or *a | c*.

a |bc ⬄ a | b , a |c

b = ak0 c = ak1

we can combine these two equations and we get:

c = (k1  or b =(k0

Since *a* is not part of the equation it proves that *a* CANNOT divide into *b* or *c*

13) Suppose that *a* and *b* are integers, *a* ≡ 4(mod 13), and *b* ≡ 9(mod 13). Find the integer *c* +++with 0≤c≤12 such that

1. *c* ≡ 9*a*(mod13)

9a = 9(4mod13) = 36 mod 13 = 10

9a(mod13) = 10 mod 13 = 10

c = 10

Prove:

1. 10≡36 mod 13 ⬄ 10 mod 13 = 36 mod 13 (equivalence)
2. 10 - 36 = -26

- 26/13 = -2

1. *c* ≡ 11*b*(mod13)

11b = 11(9mod13) = 99 mod 13 = 8

11b(mod13) = 8 mod 13 = 8

c = 8

Prove:

1. 8≡99 mod 13 ⬄ 8 mod 13 = 99 mod 13(equivalence)
2. 8 - 99 = -91

-91/13 = -7

1. *c* ≡ *a*+*b*(mod13)

a + b =(4mod13) + (9mod13) = 13 mod 13 = 0

a+b(mod13) = 0 mod 13 = 0

c = 0

Prove:

1. 0≡13 mod 13 ⬄ 0 mod 13 = 13 mod 13 (equivalence)
2. 13 – 0 = 13 / 13 = 1
3. *c* ≡ 2*a*+3*b*(mod13)

2a + 3b = 2(4mod13)+3(9mod13) = 8mod13 + 27mod13 = 35mod13 = 9

2a + 3b(mod13) = 9 mod 13 = 9

c = 9

Prove:

1. 9≡35 mod 13 ⬄ 9 mod 13 = 35 mod 13(equivalence)
2. 35 – 9 = 26 / 13 = 2
3. *c* ≡ a2+*b2*(mod13)

a2+b2 = (4mod13)2+(9mod13)2 = 16 + 81 = 97

a2+b2(mod 13) = 97 mod 13 = 6

c = 6

Prove:

1. 97≡6 mod 13 ⬄ 97 mod 13 = 6 mod 13(equivalence)
2. 97 – 6 = 91 / 13 = 7
3. *c* ≡ a3-*b3*(mod13)

a3-b3 = (4mod13)3-(9mod13)3 = 64 - 729 = -665

a3-b3 (mod 13) = -665 mod 13 = 11

c = 11

Prove:

1. -665≡11 mod 13 ⬄ -665 mod 13 = 11 mod 13(equivalence)
2. 11 – (-665) = 676 / 13 = 52

# 17) Show that if *n* and *k* are positive integers , then [⌈ ⌉=⌊ ⌋+1](https://math.stackexchange.com/questions/421509/show-that-if-n-and-k-are-positive-integers-then-lceil-fracnk-rceil)

Since n=kq+r , where 0 ≤ r < k

**Case 1: r ≠ 0** **(remainder is not 0)**

n =kq + r ⬄ = q +

is an improper fraction (not 0 because we stated not zero) in this case because,

0 ≤ r < k ⬄ 0≤ < 1

Therefore,

= q +⬄ ⌈ ⌉ = q+1, ‘+1’ because we round up

Then,

n-1 =kq + r – 1 ⬄ = q +

is an improper fraction (not 0 because we stated not zero) in this case because,

0 ≤ r-1 < k ⬄ 0≤ < 1

Therefore,

⌊ ⌋ = q , no r because we stated r < 1

If we add ⌊ ⌋

⌊ ⌋⌊ ⌋ = ⌈ ⌉

**Case 2: r = 0 (remainder is 0)**

n =kq + r , where r = 0 -> n = kq

Therefore ,

n = kq ⬄ = q ⬄⌈ ⌉ =q

Then,

n-1 = kq-1 ⬄ = q -

Since , - adds with q( q is an integer) make q less then original value by fractions not whole numbers. Ie. If q = 60 , then it would be 59 ≤ q < 60.

So,

⌊ ⌋

If we add ⌊ ⌋

⌊ ⌋ + 1

=> ⌊ ⌋ + 1

=> ⌊ ⌋ + 1 ⌈ ⌉

# 18) Show that if *a* is an integer and *d* is an integer greater then 1, then the quotient and +++remainder obtained when *a* is divided by *d* are ⌊ ⌋ and *a*-*d­*⌊ ⌋, respectively.

# \* Note: 0≤ r < d , such that a = dq+r \*

# a = dq + r

# => = q + , since 0≤ < 1

# =>⌊ ⌋ = q + (some decimal) ⬄ ⌊ ⌋ = q ✓

# => a = d(⌊ ⌋) + r

# => r = a - d(⌊ ⌋)✓

# 29) Decide whether each of these integers is congruent to 5 modulo 17.

# 80

# 80-5 = 75 / 17 ≈ 4.411, no

# 103

# 103 – 5 = 98/17 ≈ 5.764, no

# -29

# -29 – 5 = -34 / 17 = -2, yes

# d) -122

# -122 – 5 = -127 / 17 ≈ -7.470 , no

# 34) Show that if *a*≡b(mod m) and *c*≡*d*(mod *m)*, where *a* , *b* , *c* , *d* , and *m* are integers with +++m ≥ 2 , then *a­* – *c* ≡ *b* – *d*(mod m).

*a*≡b(mod m) <=> a=b+m , b = a+m

*c*≡d(mod m) <=> c = d+m , d = c+m

*a*−*c* = b-d**✓**

*b*−*d* = a-c**✓**

# 40) Prove that if *n* is an odd positive integer, then n2≡ 1(mod 8).

# n = 2k + 1, where k > 0

# n2 = (2k + 1)2= 4k2+ 4k + 1 = 4k(k+1) + 1

# if k = 0,

# n2 = 1

# 1≡1(mod 8)

# If k > 0,

# n2 = 4k(k+1)+1

# Sinc, k(k+1) is always even we can say that is k(k+1) = 2k

# n2 = 4(2k) +1

# n2 = 8k + 1

# Hence, n2 ≡1(mod 8)

*Section 4.2*

5. Convert the octal expansion of each of these integers to a binary expansion.

a) (572)8

101111010

b) (1604)8

1110000100

c) (423)8

100010011

d) (2417)8

10100001111

7. Convert the hex expansion of each of these integers to a binary expansion.

a) (80E)16

100000001110

b) (135AB)16

10011010110101011

c) (ABBA)16

1010101110111010

d) (DEFACED)16

1101111011111010110011101101

14. Show that the binary expansions of a positive integer can be obtained from +++its hexadecimal expansion by translating each hexadecimal digit into a +++block of four binary digits.

If *hi*is a hexadecimal digit then we know that

*hi* = 20*bi0* + 21*bi1* + 22*bi2* + 23*bi3*

Therefore,

*(h3h2h1h0)* = (20*b00* + 21*b01* + 22*b02* + 23*b03*) + (20*b10* + 21*b11* + 22*b12* + 23*b13*) + +======== (20*b20* + 21*b21* + 22*b22* + 23*b23*) + (20*b30* + 21*b31* + 22*b32* + 23*b33*) +

19. Give a procedure for converting from the octal expansion of an integer to +++its hexadecimal expansion using binary notation as an intermediate step.

First, we convert octal expansion to binary which can be done in 9 steps:

**Step 1:** Let the given number have n number of digits++++++++++++++++++++ **Step 2:** Multiply each digit of the number with 8n-1, when the digit is in the nth ++++++position from the right end of the number. If the number has decimal part ++++++the multiply each digit in the decimal part by when the digit is in the mth ++++++position from the decimal point. +++ **Step 3:** Add all terms after multiplication **Step 4:** The obtained value is the equivalent decimal number **Step 5:** Consider the decimal number, divide it by 2 **Step 6:** Note the remainder **Step 7:** Continue the above two steps for the quotient till the quotient is zero **Step 8:** Write the remainders in the reverse order **Step 9:** The obtained number is the equivalent binary number for the given octal number.

ie.

Convert 418 to a binary number.

418 = (4 \* 81) + (1 \* 80)

= 4 \* 8 + 1 \* 1

= 32+1  
  
= 33(Decimal number)

Now convert this decimal number to a binary number.

2 | 33  
+++2 | 16 -- 1  
+++2 | 8   -- 0  
+++2 | 4  --0  
++++2 | 2  -- 0  
      1 -- 0  
  
The binary number is 1000012  
  
418 = 1000012

Then we convert binary to hexadecimal which can be done in 5 steps:

1. Working from right to left, split the binary number into groups of 4 digits. If the left-most grouping has less than 4 digits, make up the difference with zeros.
2. 0010 | 0001 // Binary of number 33
3. Populate the next row with "8 4 2 1" under each grouping. The 8-4-2-1 represents the binary place values for each of the four positions, the combination of which total up to 15 (the highest digit that can be used in a base 16 number)
   * + 1. 0010 | 0001 // Binary of number 33
       2. 8421 | 8421
4. Multiply each digit in Row 1 by the corresponding place value in Row 2 and place the result in Row 3.
   * + 1. 0010 | 0001 // Binary of number 33
       2. 8421 | 8421
       3. 0020 | 0001
5. Add the products in Row 3 for each group of 4 and place the total in Row 4.
   * + 1. 0010 | 0001 // Binary of number 33
       2. 8421 | 8421
       3. 0020 | 0001
       4. 2 | 1

1. Change any values in Row D that are greater than 9 into the hexadecimal letter they are represented by.
   * + 1. 0010 | 0001 // Binary of number 33
       2. 8421 | 8421
       3. 0020 | 0001
       4. 2 | 1
       5. 21h

22. Find the sum and the product of each of these pairs of numbers. Express your +++answers as a base 3 expansion.

a) (112)3 , (210)3

1. 112
2. + 210
3. ------
4. 1022
5. 112
6. x 210
7. -------
8. 000
9. 112
10. 1001
11. -------
12. 1001220

b) (2112)3 , (12021)3

1. 1 11
2. 2112
3. + 12021
4. -------
5. 21210
6. 2112
7. x 12021
8. ----------
9. 2112
10. 12001
11. 00000
12. 12001
13. 2112
14. -------
15. 111020122

c) (20001)3 , (1111)3

2. 20001
3. + 1111
4. -------
5. 21112

c) (20001)3 , (1111)

1. 20001
2. x 1111
3. -----------
4. 20001
5. 20001
6. 20001
7. 20001
8. -------
9. 22221111

d) (120021)3 , (2002)3

1. 11
2. 120021
3. + 2002
4. --------
5. 122100
6. 120021
7. x 1111
8. -----------
9. 1010112
10. 0000000
11. 0000000
12. 1010112
13. -----------
14. 1011122112