Name Anna Kurchenko

HW3: Standard bases and conversions (CS220-02)

1) Convert the following decimal numbers using the iterative integer division (quotient-remainder) technique from lecture. Don't forget to print the final result with its base identified.

(a) 345 to base 2

Q	R
172	1
86	0
43	0
21	1
10	1
5	0
2	1
1	0
0	1

101011001 (base 2)

(b) 654 to base 6

Q	R
109	0
18	1
3	0
0	3

3010 (base 6)

2) Show the conversion from 0xBA5E to binary using the common base factor technique from class.

16 = 2^4 <-> 2^1

0xBA5E : B= 1011, A= 1010, 5 = 0101, E=1110
put all together

1011101001011110

3) Show the conversion from the octal value 50267 to binary using the common base factor technique from class.

2^3 <-> 2^1
5 = 101
0 = 000
2 = 010
6 = 110
7 = 111
put together: 101000010110111

4) Show the conversion from 0xC731 to base 4 using the common base factor technique from class.

16 = 2^4 <-> 2^1 <-> 2^2

0xC731 : C= 1100, 7= 0111, 3 = 0011, 1 = 0001
put all together = 1100011100110001 split into groups of 2: 11 00 01 11 00 11 00 01
-> convert each group to base 4
11 -> 3 , 00 -> 0 , 01 -> 1, 00 -> 0, 11-> 3, 00 -> 0, 01-> 1
= 3010301 (base 4)

5) Show the conversion from 2021013_4 to hexadecimal using the common base factor technique from class.

4 = 2^2 <-> 2^1 <-> 2^4 = 16

2021013: 2= 10, 0= 00, 2= 10, 1=01, 0=00, 1=01, 3=11
put all together = 10001001000111 split into groups of 4: 0010 0010 0100 0111
-> convert each group to base 16
0010= 2, 0010= 2, 0100= 4, 0111= 7
= 2247 (base 16)

6) Show the conversion from 0xD6B4A to octal using the common base factor technique from class.

16 = 2^4 <-> 2^1 <-> 2^3 = 8

0xD6B4A : D= 1101, 6= 0110, B = 1011, 4= 0100, A = 1010
put all together = 11010110101101001010 split into groups of 3: 011 010 110 101 101 001 010
-> convert each group to base 8
011 -> 3, 010 -> 2, 110-> 6, 101-> 5 , 101-> 5 , 001-> 1, 010-> 2
= 3265512 (base 8)

7) Show the conversion from 311203_4 to octal using the common base factor technique from class.

4 = 2^2 <-> 2^1 <-> 2^3 = 8

311203: 3=11, 1=01, 1=01, 2= 10, 0= 00, 3=11
put all together = 110101100011 split into groups of 3: 110 101 100 011
-> convert each group to base 8
110= 6, 101= 5, 100=4, 011= 3
= 6543 (base 8)

8) Show the conversion from the octal value 753721 to hexadecimal using the common base factor technique from class.

8 = 2^3 <-> 2^1 <-> 2^4 = 16

753721: 7= 111, 5= 101, 3= 011, 7=111, 2=010, 1=001
put all together = 111101011111010001 split into groups of 4: 0011 1101 0111 1101 0001
-> convert each group to base 16
0011=3, 1101= 13=D, 0111=7, 1101=13=D, 0001 =1
= 3D7D1 (base 16)

9) Show the conversion from 4170348_9 to base 3 using the common base factor technique from class.

9 = 3^2 <-> 3^1 = 3
41703489: 4= 11 , 1 = 01, 7= 21, 0 = 00, 3= 10, 4= 11, 8= 22
combine:
11012100101122 (base 3)

10) Show the conversion from 0x57AE to binary using the common base factor technique from class.

16 = 2^4 <-> 2^1

0x57AE : 5= 0101, 7=0111, A= 1010, E=1110
put all together

0101011110101110

11) Show the conversion from $1101010110100111101 1_2$ to octal using the common base factor technique from class. The number is rewritten below to prevent copy mistakes.

1101010110100111011

group by 3: 011 010 101 101 001 111 011 (added 0 to front for 3 digits per group)
convert each group to octal:
011=3, 010=2, 101=5, 101=5, 001=1, 111=7, 011=3
put all together: 3255173 (base 8)

12) What is the **character** (*not hex or decimal code… eg, 'a' is a character*) in the variable once this line of code completes?

```
char letter = 'f' - 'a' + 'A';
```

f = 102 , a = 97 , A = 65
102 - 97 + 41 = 70 = 'F'

13) What is the **character** (*not hex or decimal code… eg, 'a' is a character*) in the variable once this line of code completes?

```
char letter = 'D' - 'A' + 'a';
```

D = 68, A = 65, a = 97
68-65 + 97 = 100
100 = 'd'

14) Show the conversion to the BCD bitstring equivalent of the decimal number 174638?

1= 0001, 7 = 0111, 4 = 0100, 6= 0110, 3 = 0011, 8 = 1111
combine:
000101110100011000111111 (BCD)

15) Show the conversion to the decimal equivalent of the BCD bitstring 10101011000001001_{BCD}?

$\mathbf{10101011000001001_{BCD}}$

split: 0001 0101 0110 0000 1001 , add 0's to beginning:
0001=1, 0101=5, 0110=6, 0000=0, 1001= 9
combine:
15609 (base 10)

16) Show the conversion to the BCD bitstring equivalent of the number 103647_8? Careful, this one requires more than one conversion step.

103647(base 8) -> decimal:
each digit * corresponding coeff: 7*2^0 + 4*2^1 + 6*2^2 + 3*2^3 + 0 + 1*2^5
7+8 + 24 + 24 + 32 = 95
9 = 1001 (binary)
5 = 0101 (binary)
combine:
10010101 (BCD)