Introduction to Computer Systems and Platform Technologies

Week 1 Tutorial and Practical-introduction and Number Systems

These sheets will cover *both* tutorial and practical solutions for this course.

Number Systems

In the week 1 Module material on Canvas this week, you were introduced to different number systems. We shall go through some examples together**. Note that there are Quizzes each week**. This is beneficial practise of each weekly topic.

One question involves conversion between different bases e.g. decimal to binary, decimal to hexadecimal. There are many methods but the more important thing is to show your working in assignments and the exam.

In the exam, there are **no calculators allowed!**

If you show your working, a slight slip will result usually in a minor deduction.

After laying out your working, you can check your solution with:

All Programs-> Accessories-> Calculator (Programmer)

Different Number Bases

Decimal
10 symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,10 = $1x10^1+0x10^0$ = 10_{10}

Binary
2 symbols 0, 1,10 = $1x2^1+0x2^0$ = 2_{10}

Base 5
5 symbols 0, 1, 2, 3, 4,10 = $1x5^1+0x5^0$ = 5_{10}

Octal
8 symbols 0, 1, 2, 3, 4, 5, 6, 7,10 =$1x8^1+0x8^0$ = 8_{10}

Hexadecimal

16 symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F,10= $1x16^1+0x16^0$ = 16_{10}

Method 1

If you use the division method of base conversion e.g. suppose you are 29 next birthday.

29 is a decimal number and can be written 29_{10} where the base is 10.

As an example, let us converse 29_{10} to base 5. The division method involves repeated division of 29_{10} by 5 and noting the remainders:

	Quotient	Remainder
29/5	5	4
5/5	1	0
1/5	0	1

You stop when 0 is obtained for the quotient.

Then you **read up the remainder column** to obtain the answer.
So $29_{10} = 104_5$

As a check $104_5 = 1x5^2 + 0x5^1 + 4x5^0 = 25+0+4 = 29_{10}$

What is 5^0? This is always 1

Note that base 5 has 5 digits 0, 1, 2, 3, 4
Note there is no 5 digit.

Method 2

Another method could have been using the powers of 5 :

$5^2 = 25$	$5^1 = 5$	$5^0=1$
1	0	4

If you consider 29_{10}, it is made up of one $25= 5^2$, zero 5 and 4 ones or units

So again $29_{10} = 104_5$

You can use any mathematical method. This second method is used in “Data Communications” – a later subject.

Consider the following table:

Decimal	Binary	Base 5	Octal-base 8	Hexadecimal
0	**0000**	**0**	**0**	**0**
1	**0001**	**1**	**1**	**1**
2	**0010**	**2**	**2**	**2**
3	**0011**	**3**	**3**	**3**
4	**0100**	**4**	**4**	**4**
5	**0101**	**10**	**5**	**5**
6	**0110**	**11**	**6**	**6**
7	**0111**	**12**	**7**	**7**
8	**1000**	**13**	**10**	**8**
9	**1001**	**14**	**11**	**9**
10	**1010**	**20**	**12**	**A**
11	**1011**	**21**	**13**	**B**
12	**1100**	**22**	**14**	**C**
13	**1101**	**23**	**15**	**D**
14	**1110**	**24**	**16**	**E**
15	**1111**	**30**	**17**	**F**
16	**0001 0000**	**31**	**010 000= 20**	**F+1= 10**
17	**0001 0001**	**32**	**21**	**11**
18	**0001 0010**	**33**	**22**	**12**
19	**0001 0011**	**34**	**23**	**13**
20	**0001 0100**	**40**	**24**	**14**
21	**0001 0101**	**41**	**25**	**15**

The above table is very useful for your calculations.
Note in the binary column you are really doing binary addition:

For example:

Carries		+1	
	0	0	1
+	0	0	1
		1	0

For example:

Carries	+1	+1	
	0	1	1
+	0	0	1
	1	0	0

Note in different “contexts” hexadecimal has different symbols e.g 0x20 is used in “C” programming and $ - sometimes in “Mathematics”:

$0x20 = 20_{16} = \$20 = 32_{10}$

$=2x16^{1}+0x16^{0}$
$=32_{10}$

Tutorial Questions

Number System Conversions

Question 1
Decimal to Other Base Systems (binary/octal/hexadecimal conversions).
Convert the following decimal numbers to binary and hexadecimal:

a. 117_{10}
b. 127_{10}
c. 128_{10}
d. 255_{10}
Solutions:

Binary a. 117_{10}

	Quotient	Remainder
117/2	58	1
58/2	29	0
29/2	14	1
14/2	7	0
7/2	3	1
3/2	1	1
1/2	0(STOP)	1

Read up the Remainder column to get the equivalent binary number.

$117_{10} = 0111\ 0101_2$

Octal equivalent - from the "binary string" cut into groups of 3 bits from the right, the **Least Significant Bit (LSB)**, and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit)**, to fill out the last group of 3 bits.
Then replace each 3-digit group with the equivalent octal digit
So,
$001\ 110\ 101_2 = 165_8$

Hexadecimal equivalent - from the "binary string" cut into groups of 4 bits – called a "nibble" **starting from the right, the LSB (Least Significant Bit)**, and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit)**, to fill out the last group of 4 bits.
Then replace each 4-digit group with the equivalent hexadecimal digit.
So,
$0111\ 0101_2 = 75_{16}$

Binary b. 127_{10}

	Quotient	Remainder
127/2	63	1
63/2	31	1
31/2	15	1
15/2	7	1
7/2	3	1
3/2	1	1
1/2	0(STOP)	1

Read up the Remainder column
$127_{10} = 111\ 1111_2$

Octal equivalent - from the "binary string" cut into groups of 3 bits **starting from the right, the LSB (Least Significant Bit)**, and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit)**, to fill out the last group of 3 bits.
Then replace each 3-digit group with the equivalent octal digit.
So,
$001\ 111\ 111_2 = 177_8$

Hexadecimal equivalent - from the "binary string" cut into groups of 4 bits – called a "nibble" **starting from the right, the LSB (Least Significant Bit)**, and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit)**, to fill out the last group of 4 bits.
Then replace each 4-digit group with the equivalent hexadecimal digit.
So,
$0111\ 1111_2 = 7F_{16}$

Binary c. 128_{10}

	Quotient	Remainder
128/2	64	0
64/2	32	0
32/2	16	0
16/2	8	0
8/2	4	0
4/2	2	0
2/2	1	0
1/2	0(STOP)	1

Read up the Remainder column

$128_{10} = 1000\ 0000_2$

Octal equivalent - from the "binary string" cut into groups of 3 bits **starting from the right, the LSB (Least Significant Bit)**, and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit)**, to fill out the last group of 3 bits.
Then replace each 3-digit group with the equivalent octal digit.
So,
$010\ 000\ 000_2 = 200_8$

Hexadecimal equivalent - from the "binary string" cut into groups of 4 bits – called a "nibble", **starting from the right, the LSB (Least Significant Bit)**, and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit)**, to fill out the last group of 4 bits.
Then replace each 4-digit group with the equivalent hexadecimal digit.
So,
$1000\ 0000_2 = 80_{16}$

Binary d. 255_{10}

	Quotient	Remainder
255/2	127	1
127/2	63	1
63/2	31	1
31/2	15	1
15/2	7	1
7/2	3	1
3/2	1	1
1/2	0(STOP)	1

Read up the Remainder column

$255_{10} = 1111\ 1111_2$

Octal equivalent - from the "binary string" cut into groups of 3 bits **starting from the right, LSB (Least Significant Bit)**, and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit)**, to fill out the last group of 3 bits.
Then replace each 3-digit group with the equivalent octal digit.
So,
$011\ 111\ 111_2 = 377_8$

Hexadecimal equivalent - from the "binary string" cut into groups of 4 bits – called a "nibble" **starting from the right, LSB (Least Significant Bit)**, and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit),** to fill out the last group of 4 bits.
Then replace each 4-digit group with the equivalent hexadecimal digit.
So,
$1111\ 1111_2 = FF_{16}$

Question 2 Other Base Systems (binary, octal, and hexadecimal) to Decimal

a) Convert 1101_2 to decimal
b) Convert 7014_8 to decimal
c) Convert $7DE_{16}$ to decimal

Solutions:

a) $1101_2 = 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 8 + 4 + 0 + 1 = 13_{10}$

b) $7014_8 = 7x8^3 + 0x8^2 + 1x8^1 + 4x8^0 = 7x512 + 0 + 8 + 4 = 3596_{10}$

c) $7DE_{16} = 7x16^2 + 13x16^1 + 14x16^0 = 7x256 + 208 + 14x1 =$
$= 2014_{10}$

Question 3 Binary<-> Octal, Binary<->Hexadecimal

a) Convert 1 1100 1010 1110 1111 1111_2 to octal
b) Convert 1010 1001 0101 1111 1000_2 to hexadecimal
c) Convert 671_8 to binary
d) Convert $DEADFACE_{16}$ to binary

Solutions:

a) The binary number is cut into groups of **three**, **starting from the right bit, the LSB (Least Significant Bit)**. Add extra zeroes to the front bit, the **MSB (Most Significant Bit),** to fill out the last group of three if necessary. Then replace each **3-digit** group with the equivalent **octal** digit.

   111 001 010 111 011 111 $111_2 = 7127377_8$

b) The binary number is cut into groups of **four**, a "nibble", **starting from the right bit , the LSB (Least Significant Bit)**. Add extra zeroes to the front bit , the **MSB (Most Significant Bit),** to fill out the last group of three if necessary. Then replace each **4-digit** group with the equivalent **hexadecimal** digit.

   1010 1001 0101 1111 $1000_2 = A95F8_{16}$

c) Convert each octal digit to a **3-digit** binary number. Combine all the resulting binary groups (of **3 digits** each) into a single binary number.

   $671_8 = 110\ 111\ 001_2$

d) Convert each hexadecimal digit to a 4-**digit** binary number. Combine all the resulting binary groups (of 4 **digits** each) into a single binary number.

$\text{DEADFACE}_8 = 1101\ 1110\ 1010\ 1101\ 1111\ 1010\ 1100\ 1110_2$

Question 4- Other Base Systems (decimal, binary, octal, and hexadecimal) to Non-Decimal

a) Convert 217_{10} to base 7
b) Convert 1101_2 to base 5
c) Convert 7014_8 to base 9
d) Convert $7DE_{16}$ to base 6

Solutions:

a) **Convert 217_{10} to base 7.** The division method involves repeated division of 217_{10} by 7 and noting the remainders:

	Quotient	Remainder
217/7	31	0
31/7	4	3
4/7	0(STOP)	4

You stop when 0 is obtained for the quotient.

Then you read up the remainder column to obtain the answer.

So $217_{10} = 430_7$

As a check $430_7 = 4x7^2 + 3x7^1 + 0x7^0 = 4x49+21+0 = 217_{10}$

What is 7^0? This is always 1

Note that base 7 has 7 digits 0,1,2,3,4,5,6

Note there is no 7 digit.

b) Convert 1101_2 to base 5

Solution:

Step 1: convert 1101_2 to decimal: $1101_2 = 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 8 + 4 +0+1= 13_{10}$

Step 2: convert 13_{10} to base 5:

	Quotient	Remainder
13/5	2	3
2/5	0	2

You stop when 0 is obtained for the quotient.

Then you read up the remainder column to obtain the answer.

So $13_{10} = 23_5$

c) **Convert 7014_8 to base 9**

Answer:

Step 1: convert 7014_8 to decimal: $7014_8 = 7x8^3 + 0x8^2+1x8^1+ 4x8^0 = 7x512+0+8+4x1 = 3596_{10}$

Step 2: convert 3596_{10} to base 9: $3596_{10} = 4835_9$

	Quotient	Remainder
3596/9	399	5
399/9	44	3
44/9	4	8
4/9	0(STOP)	4

d) **Convert $7DE_{16}$ to base 6**

Solution:

Step 1: Convert $7DE_{16}$ to decimal: $7DE_{16} = 7x16^2 + 13x16^1+14x16^0= 7x256+208+14x1= 2014_{10}$

Step 2: convert 2014_{10} to base 6: $2014_{10} = 13154_6$

	Quotient	Remainder
2014/6	335	4
335/6	55	5
55/6	9	1
9/6	1	3
1/6	0(STOP)	1

Question 5 Some special conversions

Perform the following conversions:

a. Binary to decimal, octal and hexadecimal:

a) $01111111_2 =$
b) $10000000_2 =$
c) $11111111_2 =$

Solutions:

a) Binary to Decimal:

$0111\ 1111_2 = 0x2^7 + 1x2^6+1x2^5 + 1x2^4 +1x2^3 + 1x2^2+1x2^1+ 1x2^0$ =

$0+64 + 32 +16+8+4+2+1= 127_{10}$

Octal equivalent - from the "binary string" cut into groups of 3 bits from the right bit , the **Least Significant Bit (LSB)** and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit),** to fill out the last group of 3 bits .

Then replace each 3-digit group with the equivalent octal digit

So,

$001\ 111\ 111_2 = 177_8$

Hexadecimal equivalent - from the "binary string" cut into groups of 4 bits – called a "nibble" from the right, the **Least Significant Bit (LSB)** and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit),** to fill out the last group of 4 bits.

Then replace each 4-digit group with the equivalent hexadecimal digit

So,

$0111\ 1111_2 = 7F_{16}$

b) Binary to Decimal:

$1000\ 0000_2 = 1x2^7 + 0x2^6+0x2^5 + 0x2^4 +0x2^3 + 0x2^2+0x2^1+ 0x2^0$ =

$128+0 + 0 +0+0+0+0+0= 128_{10}$

Octal equivalent - from the "binary string" cut into groups of 3 bits from the right bit, the **Least Significant Bit (LSB)** and if necessary add extra zeroes to the front binary digit , the **MSB (Most Significant Bit),** to fill out the last group of 3 bits.

Then replace each 3-digit group with the equivalent octal digit

So,

$010\ 000\ 000_2 = 200_8$

Hexadecimal equivalent - from the "binary string" cut into groups of 4 bits – called a "nibble" from the right, the **Least Significant Bit (LSB)** and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit),** to fill out the last group of 4 bits.

Then replace each 4-digit group with the equivalent hexadecimal digit

So,

$1000\ 0000_2 = 80_{16}$

c) Binary to Decimal:

$1111\ 1111_2 = 1x2^7 + 1x2^6+1x2^5 + 1x2^4 +1x2^3 + 1x2^2+1x2^1+ 1x2^0$ =

$128+64 + 32 +16+8+4+2+1= 255_{10}$

Octal equivalent - from the "binary string" cut into groups of 3 bits from the right, the **Least Significant Bit (LSB)** and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit),** to fill out the last group of 3

bits.
Then replace each 3-digit group with the equivalent octal digit
So,
$011\ 111\ 111_2 = 377_8$

Hexadecimal equivalent - from the "binary string" cut into groups of 4 bits – called a "nibble" from the right, the **Least Significant Bit (LSB)** and if necessary add extra zeroes to the front binary digit, the **MSB (Most Significant Bit),** to fill out the last group of 4 bits.
Then replace each 4-digit group with the equivalent hexadecimal digit
So,

$1111\ 1111_2 = FF_{16}$

Question 6

What is the largest number that you can get with 4 bits, with 8 bits and 16 bits?
Solutions:
The maximum/largest **unsigned** number stored in an n-bit word is 2^n-1.

1) A 4 bit word stores 16 numbers between 0 and 15 (2^4-1) **inclusive.**

Largest number is $1111_2 = 2^4 - 1 = 2x2x2x2 - 1 = 15_{10}$

2) A 8 bit word stores 256 numbers between 0 and 255 (2^8-1) **inclusive.**

Largest number is $1111\ 1111_2 = 2^8 - 1 = 2x2x2x2x2x2x2x2 - 1 = 255_{10}$

3) A 16 bit word stores 65536 numbers between 0 and 65535 (2^{16}-1) **inclusive.**

Largest number is $1111\ 1111\ 1111\ 1111_2 = 2^{16} - 1 =$
$2x2x2x2x2x2x2x2x2x2x2x2x2x2x2x2 - 1 = 65535_{10}$

Question 7

1) Jim is 29 years old. Convert Jim's age to binary and hexadecimal by a mathematical method on paper – not a calculator!
2) Now convert your age to binary and hexadecimal by the same mathematical method. Add the 2 binary ages show your working.
3) Suppose the "simple" ALU (Arithmetic Logical Unit) handles 4 bits only (nibbles), is your addition "valid"?

Solutions:

(i) **Remember there is no set method.**

Binary form of Jim's age.

	Quotient	Remainder
29/2	14	1
14/2	7	0
7/2	3	1
3/2	1	1
1/2	0	1

Read up the Remainder column

29_{10}= 1 1101_2

Convert this binary number to hexadecimal:

= 1 1101_2

= 0001 1101_2, divide into nibbles

=$1D_{16}$

(ii) Suppose you are 19:

	Quotient	Remainder
19/2	9	1
9/2	4	1
4/2	2	0
2/2	1	0
1/2	0	1

Read up the Remainder column

Therefore, 19_{10}= 1 0011_2

Convert this binary number to hexadecimal:

= 1 0011_2

= 0001 0011_2, divide into nibbles

=13_{16}

Note in converting the Binary Numbers to Hexadecimal

One method is to go binary -> hexadecimal by grouping in 4 bits **(“nibbles”)**>

19_{10}= 0001 0011_2 = 13_{16}
29_{10}= 0001 1101_2 = $1D_{16}$

Another method the students may use for decimal ->to hexadecimal -> division by 16:

	Quotient	Remainder
19/16	1	3
1/16	0	1

Read up the Remainder column when you get 0 quotient

Again, 19_{10}= 13_{16}

Use same method for 29_{10} ->to hexadecimal -> division by 16:

	Quotient	Remainder
29/16	1	13=D
1/16	0	1

Read up the Remainder column when you get 0 quotient

Again, 29_{10}= $1D_{16}$

(ii)

Carries	+1	+1	+1	+1	
	1	0	0	1	1
+1	1	1	1	0	1
1	1	0	0	0	0

1 0011_2 + 1 1101_2 = 11 0000_2
(iii)The numbers and the result cannot be stored in 4 bits

Binary Addition
Question 8
Add the following 8 bit numbers and state whether the answer is valid to 8 bit arithmetic. Show your working especially any “carries”.

a. $1111\ 0000_2 + 1111\ 1111_2$
b. $0111\ 1111_2 + 0011\ 1111_2$
c. $0111\ 0000_2 + 1111\ 0000_2$

Solutions:

a) $1111\ 0000_2 + 1111\ 1111_2$

Carry	+1	+1	+1	+1					
		1	1	1	1	0	0	0	0
		1	1	1	1	1	1	1	1
Result	1	1	1	1	0	1	1	1	1

An overflow has occurred, a carry is in the “9th” column, the result will not fit into 8 bits, so addition is invalid!

b) $0111\ 1111_2 + 0011\ 1111_2$

Carry		+1	+1	+1	+1	+1	+1	+1	
		0	1	1	1	1	1	1	1
		0	0	1	1	1	1	1	1
Result		1	0	1	1	1	1	1	0

No overflow as occurred, result fits into 8 bits, so result is valid

c) $0111\ 0000_2 + 1111\ 0000_2$

Carry	+1	+1	+1	+1	0	0	0	0	
		0	1	1	1	0	0	0	0
		1	1	1	1	0	0	0	0
Result	1	0	1	1	0	0	0	0	0

An overflow has occurred, a carry is in the “9th” column, the result will not fit into 8 bits, so addition is invalid!

Question 9 — Binary Negative Numbers

1. *To get the two's complement negative notation of an integer, you write out the number in binary. You then invert the digits, and add one to the result. Show how -27_{10} would be expressed in two's complement notation.*

SOLUTION:

	Quotient	**Remainder**
27/2	13	1
13/2	6	1
6/2	3	0
3/2	1	1
1/2	0	1

So $27_{10} = 1\ 1011_2$ - read up

Write this in 8 bits:
So $27_{10} = 0001\ 1011_2$

To get 2's complement get 1's complement and add 1:
Step 1: 1's complement = $1110\ 0100_2$
Step 2: then 2's complement = $1110\ 0100_2 + 0000\ 0001_2$
= $1110\ 0101_2$
So $-27_{10} = 1110\ 0101_2$ in 2's complement

2. *Our numbers are 8-bits long, suppose we want to subtract 27_{10} from 115_{10}, show how to perform binary subtraction using the two's complement method.*

SOLUTION:

$115_{10} - 27_{10}$

Write this subtraction as an addition $115_{10} + (-27_{10})$. We will use the 2's complement and then binary addition.

	Quotient	**Remainder**
115/2	57	1
57/2	28	1
28/2	14	0
14/2	7	0
7/2	3	1
3/2	1	1
1/2	0	1

SOLUTION:

So $115_{10} = 0111\ 0011_2$ in 8 bits

Write these numbers in 8 bits:

$115_{10} + (-27_{10}) = 0111\ 0011_2 + 1110\ 0101_2$

So this as a binary addition:

Carries	+1(Cout)	+1(Cin)	+1			+1	+1	+1	
		0	1	1	1	0	0	1	1
	+	1	1	1	0	0	1	0	1
	1	0	1	0	1	1	0	0	0

The 9th bit overflow is disregarded – **but note not always in signed(2's complement)** -as we are only interested in the first 8 bits.

So $115_{10} + (-27_{10}) = 0111\ 0011_2 + 1110\ 0101_2 = 0101\ 1000_2$

Check $= 0101\ 1000_2 = 0*2^7 + 1*2^6+0*2^5+1*2^4+ 1*2^3+0*2^2+0*2^1+0*2^0$
$= 64+16+8 = 88_{10}$

In this example, note that C_{out} is equal to C_{in} – the adition is valid and you Ignore the "overflow".

3. Our numbers are 8-bits long, suppose we want to subtract 115_{10} from 27_{10}, show how to perform binary subtraction using the two's complement method. Show how to convert the result to decimal using the two's complement method.

SOLUTION:

27_{10} - 115_{10}

Write this subtraction as an addition 27_{10} + (-115_{10}). We will use the 2's complement for -115_{10} and binary for 27_{10} and then binary addition.

So 115_{10} = 0111 0011_2 in 8 bits

To get 2's complement get 1's complement and add 1:

1's complement = 1000 1100_2

then 2's complement = 1000 1100_2+1_2 = 1000 1101_2

So -115_{10} = 1000 1101_2 in 2's complement

27_{10} + (- 115_{10}) = 0001 1011_2 + 1000 1101_2

So this as a binary addition:

Carries				+1	+1	+1	+1	+1	
		0	0	0	1	1	0	1	1
		1	0	0	0	1	1	0	1
		1	0	1	0	1	0	0	0

No overflow in 9'th column:

So 27_{10} + (- 115_{10}) = 0001 1011_2 + 1000 1101_2 = 1010 1000_2

Convert the result to decimal using the two's complement method.

To get 2's complement get 1's complement and add 1:

1's complement = 0101 0111_2

then 2's complement = 0101 0111_2+1_2

Carries						+1	+1	+1	
		0	1	0	1	0	1	1	1
		0	0	0	0	0	0	0	1
		0	1	0	1	1	0	0	0

Convert 0101 1000_2 to decimal: +88_{10} so -88_{10} was correct

So -115_{10} = 1000 1101_2 in 2's complement

Hexadecimal Addition
Question 10
Add the following hexadecimal numbers and state whether the answer is valid to 16 bit arithmetic. Show your working especially any “carries”.

a. $1ABC_{16} + 1234_{16}$
b. $ABBA_{16} + CAFE_{16}$

Solutions (a):

Method 1:

You can change C+4 to decimal ->$12_{10}+4_{10} = 16_{10}$
Then change decimal to hexadecimal $16_{10} = 10_{16}$

Carry	0	0	+1	
	1	A	B	C
+	1	2	3	4
Result	2	C	F	0

This result will fit into 16 bits.

Method 2
Step 1: Convert each hexadecimal digit to binary:
$1ABC_{16} = 0001\ 1010\ 1011\ 1100_2$
$1234_{16} = 0001\ 0010\ 0011\ 0100_2$

Step 2: Apply binary addition

Carry		0	0	+1	0	0	+1	0	0	0	+1	+1	+1	+1	0	0	
		0	0	0	1	1	0	1	0	1	0	1	1	1	1	0	0
		0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
Result		0	0	1	0	1	1	0	0	1	1	1	1	0	0	0	0

Step 3: convert the binary result to hexadecimal: $0010\ 1100\ 1111\ 0000_2 = 2CF0_{16}$
This result will fit into 16 bits.

Solutions (b):

Just a note, with hex I personally find it easier to just subtract 16 - since the most you will be adding is F+F so the carry is only ever 1.

if you were adding e.g. 3 numbers, you can subtract 32

Method 1

Carry	+1	+1	+1	+1	
		A	B	B	A
	+	C	A	F	E
Result	1	7	6	B	8

With the first column A+E , change to decimal “in your head” $10_{10}+14_{10}$ $=24_{10}$

Decimal to hexadecimal

24/16 = 1 r 8

1/16 = 0 r 1

Then convert 24 decimal to hexadecimal, $24_{10}= 18_{16}$

With the second column, change B+F+1 to decimal “in your head”, $11_{10}+15_{10} +1_{10}=27_{10}$

27/16 = 1 r 11(B)

1/16 = 0 r 1

Then change 27 decimal to hexadecimal, $27_{10} = 1B_{16}$

With the third column, change B+A+1 to decimal “in your head” $11_{10}+10_{10} +1_{10}=22_{10}$

22/16 = 1 r 6

1/16 = 0 r 1

Then change 22 decimal to hexadecimal, $22_{10} = 16_{16}$

With the fourth column, change A+C+1 to decimal “in your head” $10_{10}+12_{10} +1_{10}=23_{10}$

$= 17_{16}$

Then change 23 decimal to hexadecimal, $23_{10} = 17_{16}$

An **overflow** has occurred, the result will NOT fit into 16 bits.

Method 2

Step 1: convert each hexadecimal digit to binary:

$ABBA_{16}$ = $1010\ 1011\ 1011\ 1010_2$

$CAFE_{16}$ = $1100\ 1010\ 1111\ 1110_2$

Step 2: Apply binary addition

Carry	+1	0	0	0	+1	0	+1	+1	+1	+1	+1	+1	+1	+1	+1	0	
		1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	0
		1	1	0	0	1	0	1	0	1	1	1	1	1	1	1	0
Result	1	0	1	1	1	0	1	1	0	1	0	1	1	1	0	0	0

Step 3: convert the binary result to hexadecimal: $1\ 0111\ 0110\ 1011\ 1000_2$ = $176B8_{16}$

An overflow has occurred, the result will NOT fit into 16 bits.

You are encouraged to discuss these questions and share useful links you've found with others in the discussion forum on Canvas.

Practical Questions

Question 1

The English units of measure ('imperial') were used as examples in the lecture. We also discussed the clock system (24h 60m 60s) and the metric system. But every country has its own system.

1. *As a class activity, describe some units of measure from your own country / culture and share with the class.*

Solution:

Use this whole question as an ice breaker. The answers are not that important and will obviously differ between classes. In you look at the Canvas discussion for week 1 please publish your own examples.

Question 2

A deck of cards contains 4 suits , Spades, Hearts, Diamonds, Clubs 2 colours, 4 aces, 12 royals, and 36 numbers, and 2 jokers. Is it a number system? If not, how do we make it one. Think of points scoring.

Solution:

For a number system to work, all the symbols must be sortable. That is we must always be able to tell A > B, where A, B are any symbols in the system. In many card games with points, it is not possible to have a tie in points for any two cards. So in that case, this can be a number system.

Question 3

What is the 'duosexagesimal' system'? Should we add a space, a comma (',') and a period ('.') to this? What would you call it then? How many bits are required if converting to binary?

Solution:

An example *'duosexagesimal' system' namely* base 62 = with 62 symbols.
0-9, A-Z, a-z

If we add 3 symbols, . and , and a space then this becomes useful for simple messages. But adding 3 more symbols makes this 65 symbols, requiring 7 bits.
$000\ 0000_2$ to $111\ 1111_2$

$111\ 1111_2 = 1*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 1*2^1 + 1*2^0$

$= 64 + 32 + 16 + 8 + 4 + 2 + 1 = 127_{10}$

However, if only 2 symbols were added, then 6 bits are enough.
$00\ 0000_2$ to $11\ 1111_2$ that is 0 to 63_{10}

It would be base Pentasexagesimal (65) or Tetrasexagesimal (64) system.

Question 4

Could you use a change of number base as a form of encryption? Why / why not?
Solution:

Not really, since there is no password or key involved. It is like translating letters in Greek letters. Any adversary can easily do the reverse.

So questionable.
Some studies [1][2] are undergoing. However, no confirmation is from the mainstream domain experts.
References:

[1] Po-Han Lin, Base Encryption: Dynamic algorithms, Keys, and Symbol Set. Retrieved from:

http://www.edepot.com/baseencryption.html

[2]
https://www.dcode.fr/base-26-cipher#1
https://www.dcode.fr/ascii-code

[3] Prabhas Tiwar, Nishtha Madaan, Md.Tabrez Nafis. Cryptographic Technique: Base Change Method. International Journal of Computer Applications (0975 – 8887) Volume 118 – No.14, May 2015

Practical- 1- Question 5

The Australian Telephone System

Did you know that the 04 for mobile phones you are now using used to be an area code for NSW country? Telstra will run out of 04 mobile numbers by 2017 and introduce the 05 prefix. See the Wikipedia article on the Aussie telephone numbering system. At the bottom, the **See-also** part points to an older system.

Practical- 1- Question 6

Why a Number System needs a Zero

Why a number system needs a zero?

Do you agree with the conclusion in the above link? That a zero is needed in order to use 'place value'. How else could you do this?

Think of Roman Numerals. There is no zero there. Even negative numbers were not written any different to positive numbers. Is what they have a number system?

Practical- 1- Question 7

What are Gray Codes?

Gray codes are a number system based on the binary system with interesting properties. But you cannot do maths with it. Do some research on Gray Codes.

Solution

https://www.allaboutcircuits.com/technical-articles/gray-code-basics/

https://ncalculators.com/digital-computation/binary-gray-code-converter.htm

Research

The number systems do not end with just what we covered. There are some more and interesting number systems which are either in use or exist in fiction books.

Duodecimal system, (https://en.wikipedia.org/wiki/Duodecimal) for instance, is one of the most popular number systems amongst mathematicians. In fact, even we use this almost every day without paying much attention.

Some of the other fictional ones are mentioned here and also on a List of Numeral Systems on Wikipedia