9/10/18

Binary and Hexadecimal Numbers

Journal:
If you were to extend our numbering system to more digits, what digits would you use? Why those?

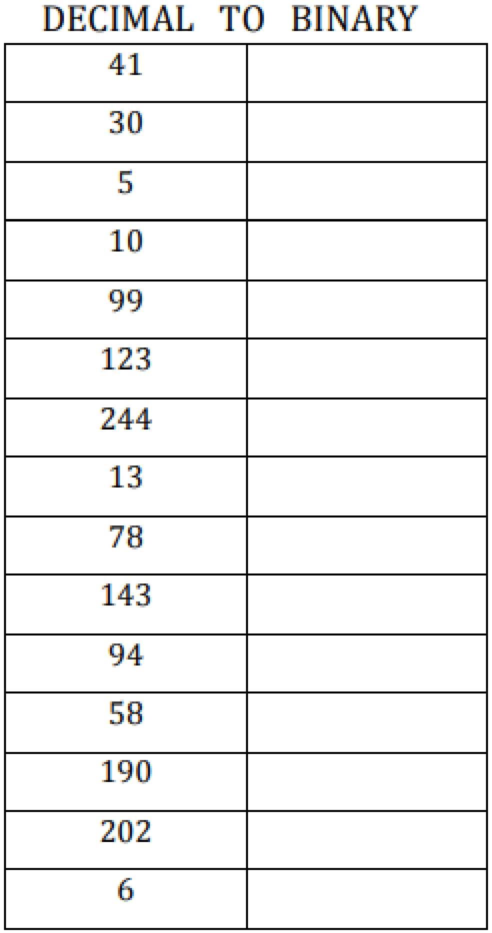
Objectives:

Content: I will be able to work with numbers from different numbering systems.

Social: I will explain my reasoning to other people.

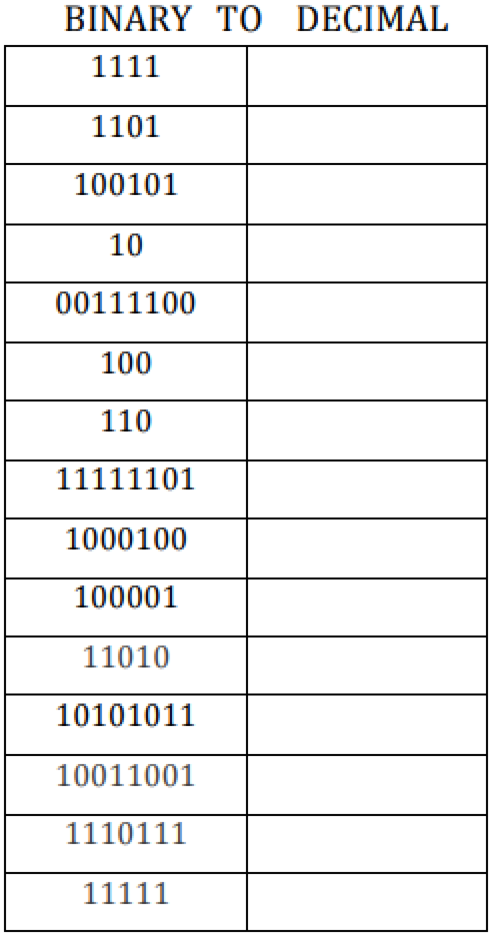
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Check Homework



DECIMAL TO BINARY

41	
30	
5	
10	
99	
123	
244	
13	
78	
143	
94	
58	
190	
202	
6	



BINARY TO DECIMAL

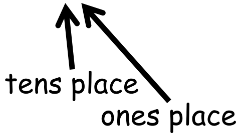
1111	
1101	
100101	
10	
00111100	
100	
110	
11111101	
1000100	
100001	
11010	
10101011	
10011001	
1110111	
11111	

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Binary Numbers

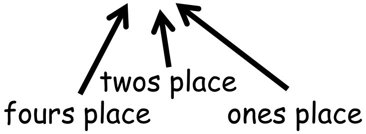
- A binary (base-two) number is just like a decimal (base-ten) number, except that instead of ten possible digits (0…9), we only have two (0…1)
  - 15_{10} 477_{10}



tens place
ones place

tens place
hundreds place
ones place

  - 111_2 111011101_2



twos place
fours place
ones place

QUESTION: What place value does this zero hold?

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

0x ______

hexadecimal

$A8901FB_{16}$

Hexadecimal

- It would be very inconvenient to write out a 64-bit address in binary:
  0010100111010110111110001001011000010001110011011110000011100000

1111

2'

0 - F

- Instead, we group each set of 4 bits together into a ***hexadecimal*** (base 16) digit:
  - The digits are 0, 1, 2, …, 9, A (10), B (11), …, E (14), F (15)

0010	1001	**1101**	0110	**1111**	1000	**1001**	0110	**0001**	0001	**1100**	1101	**1110**	0000	**1110**	0000
2	9	D	6	F	8	9	6	1	1	C	D	E	0	E	0

- …which we write, by convention, with a "0x" preceding the number to indicate it's a heXadecimal number:
  - 0x29D6F89611CDE0E0

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Representing Information

- Positive integers: Just use the binary number system
- Negative integers, letters, images, … not so easy!
  - There are many different ways to represent information
  - Some are more efficient than others
  - … but once we've solved the representation problem, we can ***use*** that information without considering ***how*** it's represented… via

Abstraction!

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Representing Integers

- Simplest idea ("ones' complement"):
  - Use one bit for a "sign bit":
    - 1 means negative, 0 means positive
  - The other bits are "complemented" (flipped) in a negative number
  - So, for example, +23 (in a 16-bit word) is represented as:
    0000000000010111
    and -23 is represented as:
    1111111111101000
  - But there are two different ways to say "zero" (0000… and 1111…)
  - It's tricky to do simple arithmetic operations like addition in the ones' complement notation
    - This is solved by the twos' complement representation, but we won't go over that

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Floating Point Numbers

- Non-integers are a problem…
- Remember that any rational number can be represented as a fraction
  - …but we probably don't want to do this, since
    - (a) we'd need to use two words for each number (i.e., the numerator and the denominator)
    - (b) fractions are hard to manipulate (add, subtract, etc.)
- Irrational numbers can't be written down at all, of course

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Floating Point Numbers

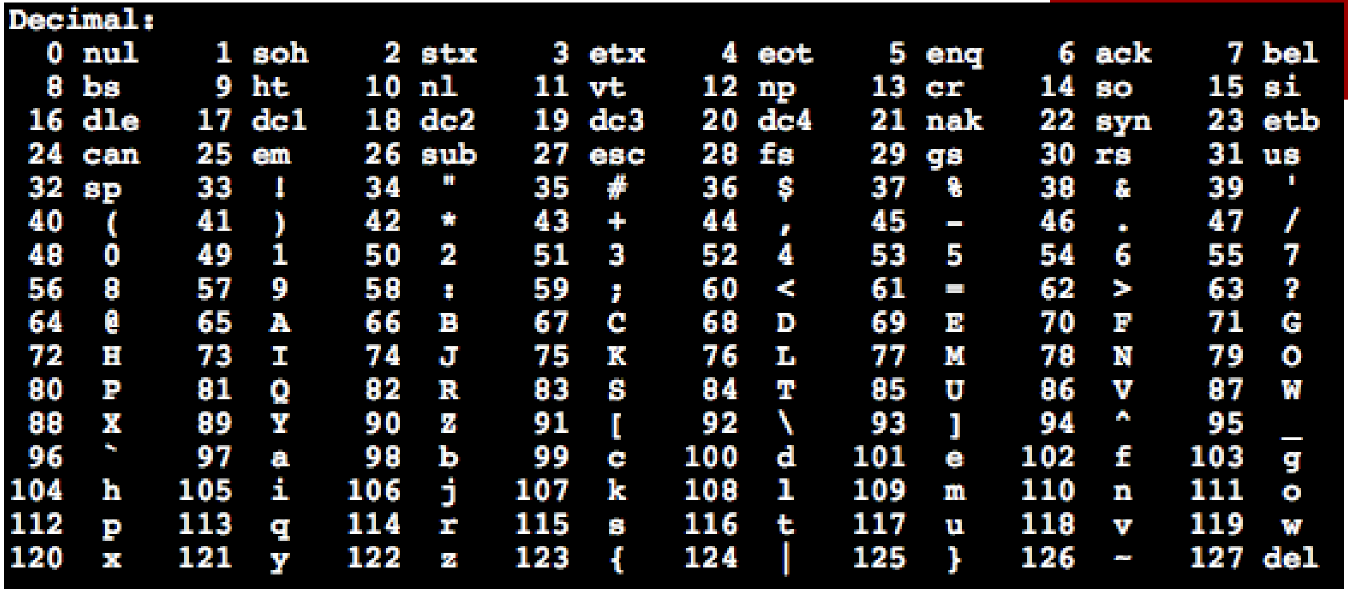
- We have ***limited precision***, since we can only represent 2^{32} different values in a 32-bit word
  - 1/3 isn't ***exactly*** 1/3 ***(let's try it on a calculator!)***
- In general, we also lose precision (introduce errors) when we operate on floating point numbers
- You don't need to know the details of how "floating point" numbers are represented

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Representing Characters

- ASCII representation: one byte [***actually 7 bits***…] == one letter == an integer from 0-128



```
Decimal:
  0 nul    1 soh    2 stx    3 etx    4 eot    5 enq    6 ack    7 bel
  8 bs     9 ht    10 nl    11 vt    12 np    13 cr    14 so    15 si
 16 dle   17 dc1   18 dc2   19 dc3   20 dc4   21 nak   22 syn   23 etb
 24 can   25 em    26 sub   27 esc   28 fs    29 gs    30 rs    31 us
 32 sp    33  !    34  "    35  #    36  $    37  %    38  &    39  '
 40  (    41  )    42  *    43  +    44  ,    45  -    46  .    47  /
 48  0    49  1    50  2    51  3    52  4    53  5    54  6    55  7
 56  8    57  9    58  :    59  ;    60  <    61  =    62  >    63  ?
 64  @    65  A    66  B    67  C    68  D    69  E    70  F    71  G
 72  H    73  I    74  J    75  K    76  L    77  M    78  N    79  O
 80  P    81  Q    82  R    83  S    84  T    85  U    86  V    87  W
 88  X    89  Y    90  Z    91  [    92  \    93  ]    94  ^    95  _
 96  `    97  a    98  b    99  c   100  d   101  e   102  f   103  g
104  h   105  i   106  j   107  k   108  l   109  m   110  n   111  o
112  p   113  q   114  r   115  s   116  t   117  u   118  v   119  w
120  x   121  y   122  z   123  {   124  |   125  }   126  ~   127 del
```

- No specific reason for ***this*** assignment of letters to integers!

- UNICODE is a popular 16-bit representation that supports accented characters like **é**

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Other Types of Data

- What kinds of data do we represent on computers, and what are the forms?

py

.h

jar

zip

rar

exe

dll

cpp

folders- organize

pictures
img
gif
jpeg
png

pixels
color

pdf

txt

mp3

mp4

videos

bat

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

7,200,000

The Scale of Data

7.2 million

- Work with an elbow partner to answer these questions

600×400=240,000
× 3

720,000
× 8
5,760,000

- Computer Screen
  - Each pixel is represented by three bytes (red, blue, green color values)
  - A standard VGA screen is 600 pixels wide by 400 pixels high
  - ***How many bits are needed to represent one VGA screen display?***

127,008,000

44,100
× 180
7,983,000
× 16

- CD
  - CDs are stored as digital samples
  - There are 44,100 samples per second
  - Each sample is represented by 16 bits
  - ***How many bits are needed to represent one 3-minute song on a CD?***

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

1 1 1
1 1 0
1 0 0 1

Representing instructions

- EVERYTHING in the computer is represented in binary, even the instructions.

- Create your own binary code to represent these equations: *(NOT the answers, represent the equations themselves)*

- 9 + 4
- ~~11 – 3~~
- ~~2 * 5~~
- ~~15 / 5~~

3 + 6

HINT:

How many bits do you need to represent the data?
They are whole numbers.
The largest number is _____ and requires _____ bits

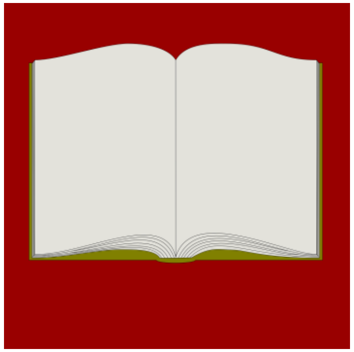
How many different operations are there? ____
Create a binary code for each operation.

Write each equation in binary.

Objectives:

Content: I will be able to work with numbers from different numbering systems.
Social: I will explain my reasoning to other people.
Language: I will use my words to explain my addition a nd calculation process to others in the class.

Homework



- Read the remainder of Chapter 1 in *Blown To Bits* (pages 13-17).
- Summarize by choosing one of each from the reading
  - Key sentence
  - Key phrase
  - Key word