Characters & Booleans

Textbook Section 2.5

Non-Numeric Data

Computers store/process numbers (bit patterns), and numbers only.

- What about characters? We use text all the time!
- What about Booleans? We are always doing logic operations.
- What about images, videos, sounds, other types of data?
    - We look at pictures, watch movies, listen to music, etc.
    - Will be covered in Comp 2659

To store anything we need to define a **mapping convention** to encode the real world item as numbers.

Characters - ASCII

- ASCII = American Standard Code for Information Interchange
  - 7-bit code, usually stored in 8-bit byte, i.e. 128 characters
  - extensions to ASCII use the 8^{th} bit, i.e. 256 characters
  - All the characters: http://www.asciitable.com/
  - E.g. in ASCII, ‘A’ is represented by 65_{10}

Characters

ASCII

- **control characters:** 0-31
- **decimal digits:** 48-57 ← note continuous ranges
- **uppercase letters:** 65-90 ← conversion between cases is easy
- **lowercase letters:** 97-122
- **punctuation:** interspersed between the above

Characters - EBCDIC

- ASCII not the only encoding that existed
- EBCDIC - Extended Binary Coded Decimal Interchange Code
    - Extended the BCDIC - Binary Coded Decimal Interchange Code
- Created by IBM
- Older than ASCII

EBCDIC Table

Left Digit(s) \ Right Digit	EBCDIC									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
6					□					
7					¢	.	<	(	+	|
8	&									
9	!	$	*	)	;	¬	-	/		
10							^	'	%	_
11	>	?								
12			:	#	@	'	=	"		a
13	b	c	d	e	f	g	h	i		
14						j	k	l	m	n
15	o	p	q	r						
16			s	t	u	v	w	x	y	z
17								\	{	}
18	[	]								
19				A	B	C	D	E	F	G
20	H	I								J
21	K	L	M	N	O	P	Q	R		
22							S	T	U	V
23	W	X	Y	Z						
24	0	1	2	3	4	5	6	7	8	9

Codes 00–63 and 250–255 are nonprintable control characters.

Characters - Unicode

- Newer standard
- The world has more than just Latin characters!
- Backwards-compatible with ASCII – Don’t want to have to redo all existing files.
- Very large character set for assigning a unique number to each human symbol!
- Encodes alphabetic characters, pictograms, etc.

Unicode

- Glyph – an elemental symbol. “a”, “*a*”, “T”, 漢字; …
- Code points: 0_{16}-$10FFFF_{16}$ which allows for 1,114,112 code points, divided up into 17 planes of 65,536 each.
- A code point is an abstraction. It does not define the encoding or the glyph.
- http://en.wikipedia.org/wiki/List_of_Unicode_characters

Unicode Encoding

- How do we store Unicode characters?
- Need a multi-byte system, compatible with ASCII.
- UTF-8 http://en.wikipedia.org/wiki/UTF-8
- 1-4 bytes per Unicode code point
- First 128 are the same as ASCII

Characters - Never Use Numerical Values

- Despite knowing the numeric values corresponding to characters you should NEVER use the numbers!!
- Why?
  - The encoding scheme could change
  - i.e. move your program to another computer.
- Always code using characters and let the compiler/assembler translate to the correct encoding scheme.

Booleans

- In theory, a Boolean (true/false) value can be represented using only one bit.
- In practice, an entire integer (one or more bytes long) is usually used (why?):
  - 0 interpreted as false
  - non-0 interpreted as true
- This is a C/C++ convention, but is an extremely poor decision. Why?

Booleans

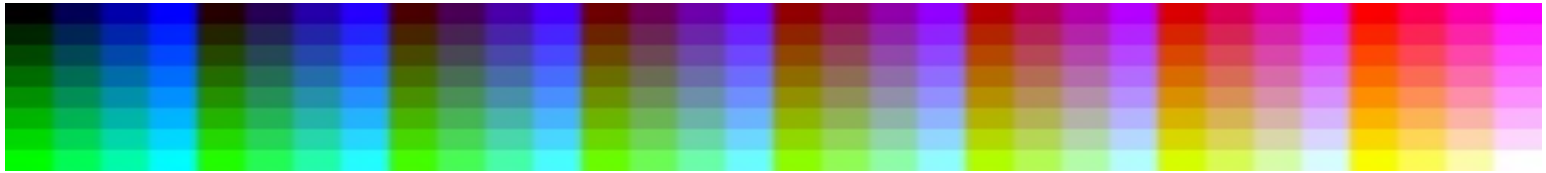
- Better decision is to use fixed values
- General thought is 0 and 1
- Better would be -1/0, why?
    - Could also be -0/0 → 1’s Complement
- Data representation is half of what needs to be done
    - Operating on the representation is the other half
- For both of the above representations boolean operations can be implemented very easily (e.g. logical AND)

Image (Pixels)

- There are numerous file formats for storing images (GIF, PNG, JPG, TIF). We will look at how pixels (picture elements) are stored.
- How much space do you need to store a pixel? It depends on the colour depth.
- A colour is made up of varying intensities of Red, Green and Blue light. We need to have a number to specify each.

Image (Pixels) - 8 Bit Color

- 3 bits for red (8 possibilities)
- 3 bits for green
- 2 bits for blue (4 possibilities)
- This gives 8 x 8 x 4 = 256 colours.



- Why are there only 2 bits for blue?

Image (Pixels) - 24 Bit Color

- True Colour, “Millions of Colours”
- 8 bits for red
- 8 bits for green
- 8 bits for blue
- How many colours? **256 x 256 x 256 = 16,777,216**
- Specify as hex: RRGGBB
- http://www.w3schools.com/tags/ref_colorpicker.asp