CMSC121 Intro to Computer Science Blaheta

Exam 1 review

27 February 2024

Some topics we've covered in this unit:

- Algorithms and preciseness
  - Specifying algorithms
  - Precise language
- Procedures/Functions
- Recursion
- Conditional execution
- Operations and expressions
- Spreadsheets
  - Formulas
    - Absolute position ($)
  - Repetition
  - Decisions
    - IF
    - AND, OR, NOT
- QR process:
  - Formulate question
  - Execute/evaluate
  - Interpret the results
  - Reasonableness checks
- Data as numbers
  - Text
  - Images
- Positive integers in binary
  - To/from decimal
  - To/from hexadecimal
  - Relevance of hexadecimal
- Floating-point numbers
  - Sci notation/normal form
  - Really big/small
  - Fractional parts
- Negative numbers
  - Complement notation
- Limitations of representations
  - # possible distinct values
  - Max/min value
  - Limited precision/inexactness

Some connections to think about:

- Recall what was difficult in the Lego exercise. Did similar problems arise when working with Light-bot or the spreadsheets? Why or why not?
- When writing Light-bot programs, we talked about looking for the part that repeats in order to decide what to put in a function. How is this similar/different to what we did with spreadsheet data?
- To make Light-bot repeat an action (once you decided what to repeat), you called F1 inside of F1 (or F1 called F2 and vice versa). How is this the same as how we did repetition inside Excel? How is it different?
- Why didn't we need any conditionals ("IF") in our Lego exercise? What form did conditionals take in Light-bot, and why was this form limiting? Why is the IF in Excel much more powerful?
- When we first talked about image representation, we hadn't covered binary yet and just treated each pixel as three cells in a spreadsheet. Knowing what we now know about representing positive integers in binary form, what would an actual stored image file be like?
- If you accidentally opened a raw image file as if it were text (that is, the program interpreted the image data as text data instead), what would that look like? What about opening a text file as if it were image data?
- What does 01000100 represent?

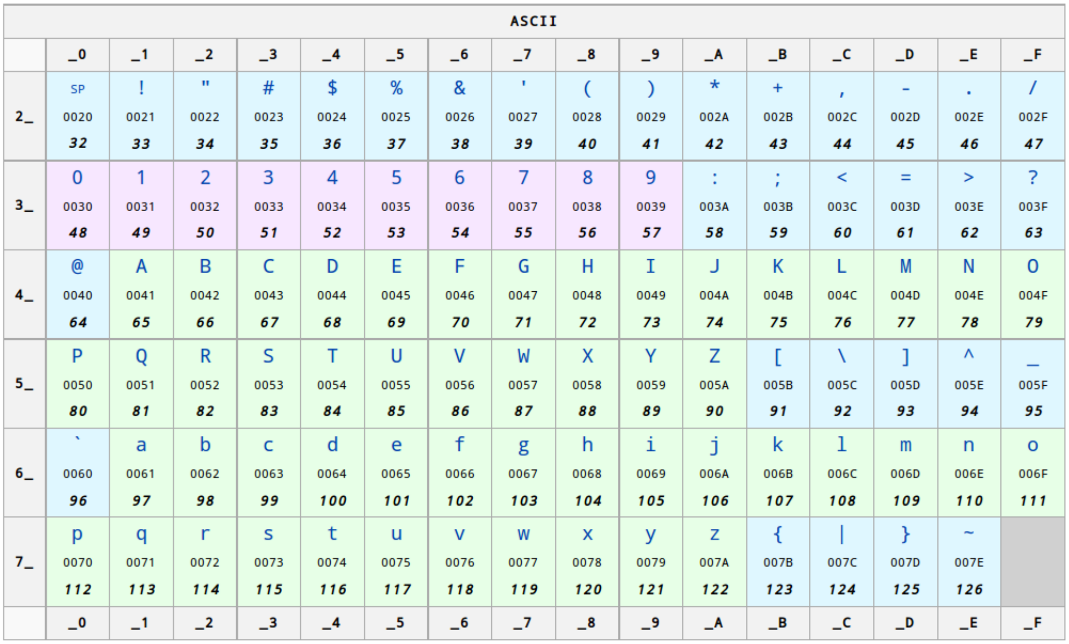
Some skills to think about and practice:

- Write instructions for some process very precisely. For instance, could you describe in words how to tie a shoe? Remember to avoid ambiguous statements, and to occasionally orient the reader with what *should* have happened or what things *should* look like.
- Follow instructions very precisely. For instance, assembling Lego from instructions, cooking from recipe, etc. Critique the instructions—how could they be changed to make them easier to follow correctly?
- Arithmetic expressions.
  - Evaluating expressions. Follow the standard order of operations, and show each step along the way.
  - Converting expressions between different forms. Know what is meant by "in-line" form and what we're calling the "built-up" algebraic form, and how to convert back and forth. Pay close attention to parentheses!
- Spreadsheet formulas:
  - Use formulas in a repetition to discover things about data.
  - Refer to parameters using absolute cell references (`$`). Know when to use this and when not to!
  - Compute values by hand. No complicated math here, but for each of the functions we've seen, know how it would apply to given spreadsheet data.
  - Use functions effectively. Which of the spreadsheet functions should be used for a particular task?
- Given a character table and some raw data, translate the stored numbers into characters they represent, and vice versa.
- Given a set of image data, describe how to systematically transform that data in order to change the corresponding image in predictable ways (e.g. make everything darker, change image to greyscale, make the green parts of the image even greener, etc)
- Convert non-negative integers freely between binary, decimal, and hexadecimal format.
- Identify negative integers in complement notation and describe how they're used.

As I said, you do not have to memorise the list of available spreadsheet functions. This is the list of functions you'll be expected to know how to use, and this list will appear on the exam for you to refer to:

General functions:	Decision functions:
`COUNT(` *range* `)`	`IF(` *condition*, *value-if-true*, *value-if-false* `)`
`SUM(` *range* `)`	`ISNUMBER(` *cell* `)`
`AVERAGE(` *range* `)`	`AND(` *condition1*, *condition2* `)`
`MEDIAN(` *range* `)`	`OR(` *condition1*, *condition2* `)`
`MAX(` *range* `)`	`NOT(` *condition1* `)`
`MIN(` *range* `)`	

You also do not need to memorise any of the character tables we've looked at, but you will be expected to know how to read them and use them. Below, for instance, is the table for ASCII; if there are questions about other tables, I will provide the tables in a format similar to this one.



ASCII

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
2_	SP 0020 32	! 0021 33	" 0022 34	# 0023 35	$ 0024 36	% 0025 37	& 0026 38	' 0027 39	(0028 40	) 0029 41	* 002A 42	+ 002B 43	, 002C 44	- 002D 45	. 002E 46	/ 002F 47
3_	0 0030 48	1 0031 49	2 0032 50	3 0033 51	4 0034 52	5 0035 53	6 0036 54	7 0037 55	8 0038 56	9 0039 57	: 003A 58	; 003B 59	< 003C 60	= 003D 61	> 003E 62	? 003F 63
4_	@ 0040 64	A 0041 65	B 0042 66	C 0043 67	D 0044 68	E 0045 69	F 0046 70	G 0047 71	H 0048 72	I 0049 73	J 004A 74	K 004B 75	L 004C 76	M 004D 77	N 004E 78	O 004F 79
5_	P 0050 80	Q 0051 81	R 0052 82	S 0053 83	T 0054 84	U 0055 85	V 0056 86	W 0057 87	X 0058 88	Y 0059 89	Z 005A 90	[005B 91	\ 005C 92	] 005D 93	^ 005E 94	_ 005F 95
6_	` 0060 96	a 0061 97	b 0062 98	c 0063 99	d 0064 100	e 0065 101	f 0066 102	g 0067 103	h 0068 104	i 0069 105	j 006A 106	k 006B 107	l 006C 108	m 006D 109	n 006E 110	o 006F 111
7_	p 0070 112	q 0071 113	r 0072 114	s 0073 115	t 0074 116	u 0075 117	v 0076 118	w 0077 119	x 0078 120	y 0079 121	z 007A 122	{ 007B 123	| 007C 124	} 007D 125	~ 007E 126	
	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F