Exploring Computer Science Concepts

Via ACSL Competitions

Number Systems

- Decimal
  - base 10
- Binary
  - base 2
- Octal
  - base 8
- HexaDecimal
  - base 16

Digits per base

- Decimal
  - Ten digits
  - 0 1 2 3 4 5 6 7 8 9
- Binary
  - Two Digits
  - 0 1
- Octal
  - 8 digits
  - 0 1 2 3 4 5 6 7
- Hexadecimal
  - 16 digits
  - 0 1 2 3 4 5 6 7 8 9 A B C D E F

Counting in Decimal

- Increment the lowest place value (right most digit)
- When last digit is reached
  - Set the current column to 0
  - Increment the column on the left by 1
- Lets count with decimal
  - 0 to 10
  - 95 to105
- How frequently do we add a new digit ?

Decimal	Decimal
0	95
1	96
2	97
3	98
4	99
5	**100**
6	101
7	102
8	103
9	104
10	105

Counting in other bases

- Counting in binary
  - We add a new digit frequently
    - At 2 , 4, 8, 16 … decimal values
- Counting in Octal
  - We add a new digit at every
    - At 8, 64,128 …
- Counting in HexaDecimal
  - We add a new digit at every
    - At 16, 256 … values

Decimal	Binary	Octal	HexaDecimal
0	0	0	1
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Place Values

- Decimal
  - *We deal in power of 10s*
  - 2452 = 2 * 1000 + 4 * 100 + 5 * 10 + 2 * 1
  - 101   =   1 * 100 + 0 * 10 + 1 * 1
- Binary
  - *We deal in power of 2s*
  - 111 =   1 * 4 + 1 * 2 + 1 * 1
  - 1010 = 1 * 8 + 0 * 4 + 1 * 2 + 0 * 1

Place values
Converting binary →decimal

Decimal	Binary
2452 → $$2 \times 10^3 = 2000$$ $$4 \times 10^2 = 400$$ $$5 \times 10^1 = 50$$ $$2 \times 10^0 = 2$$ $$= 2452$$	111 → $$1 \times 2^2 = 1 \times 4 = 4$$ $$1 \times 2^1 = 1 \times 2 = 2$$ $$1 \times 2^0 = 1 \times 1 = 1$$ $$= 7$$
101 → $$1 \times 10^2 = 100$$ $$0 \times 10^1 = 0$$ $$1 \times 10^0 = 1$$ $$= 101$$	1010 → $$1 \times 2^3 = 1 \times 8 = 8$$ $$0 \times 2^2 = 0 \times 4 = 0$$ $$1 \times 2^1 = 0 \times 2 = 2$$ $$0 \times 2^0 = 0 \times 1 = 0$$ $$= 10$$
756 →	1000 →

Converting Binary to Decimal

128	64	32	16	8	4	2	1

	1	0	0	1	0	0	1	= 64 + 16 + 1 = 81
1	0	0	0	0	1	1	1	= 128 + 4 + 2 + 1 = 135
1	1	1	0	0	0	0	0	= ?

Converting Decimal to Binary

128   64   32   16   8   4   2   1
_____

=   22

=   33

= 130

Exercises

- Convert following Decimal numbers to binary
  - 127_{10}
  - 128_{10}
  - 129_{10}
  - 255_{10}
  - 256_{10}
- Convert following Binary numbers to Decimal
  - 101101_2
  - 1110_2
  - 1111_2
  - 0110_2

Challenge

- How does binary addition and subtraction work ?

Place values
Converting octal,hexadecimal→decimal

Octal → Decimal	HexaDecimal → Decimal

777 →
 	
 7 x 8^2 = 7 x 64 = 448
 7 x 8^1 = 7 x   8 =   56
 7 x 8^0 = 7 x   1 =   7
                     = 511
	2AB →
	
	 2 x 16^2 = 2 x 256 = 512
	 A x 16^1 = 10 x 16 = 160
	 B x 16^0 = 11 x 1 = 11
	                             = 683

137 →
 	
 1 x 8^2 = 1 x 64 = 64
 3 x 8^1 = 3 x   8 = 24
 7 x 8^0 = 7 x   1 =   7
                   = 95
	101 →
	
	 1 x 16^2 = 1 x 256 = 256
	 0 x 16^1 = 0 x 16 = 0
	 1 x 16^0 = 1 x 1 = 1
	                             = 257

756 →	4A3 →

Converting Decimal to Octal

$(312)_{10}$ → $(?)_8$

$312 / 8$ → quotient : 39 , Remainder 0

$39 / 8$ → quotient : 4 , Remainder 7

$4 / 8$ → quotient : 0 , Remainder 4

$(\mathbf{470})_8$

$(112)_{10}$ → $(?)_8$

$112 / 8$ → quotient : 14 , Remainder 0

$14 / 8$ → quotient : 1 , Remainder 6

$1 / 8$ → quotient : 0 , Remainder 1

$(\mathbf{160})_8$

Exercises

- Convert following Decimal numbers to Octal
  - 111_{10}
  - 88_{10}
  - 511_{10}
  - 512_{10}
  - 513_{10}
- Convert following Octal numbers to Decimal
  - 45_8
  - 77_8
  - 100_8
  - 101_8

Converting Decimal to Hexadecimal

$(312)_{10}$ → $(?)_8$

312 / 16 → quotient : 19 , Remainder 8

19 / 16 → quotient : 1 , Remainder 3

1 / 16 → quotient : 0 , Remainder 1

(138)$_{16}$

$(112)_{10}$ → $(?)_8$

112 / 16 → quotient : 7 , Remainder 0

7 / 16 → quotient : 0 , Remainder 7

(70)$_{16}$

Hexadecimal → Octal

$(AC)_{16}$→ $(\textbf{1010})_2$ x 16 + $(\textbf{1100})_2$ x 1 //Replace hexa with binary

→ $(\textbf{1010 1100})_2$ // Convert to binary

→ $(\textbf{010 101 100})_2$ // group by 3s , added extra 0s in the front

→ $(\textbf{254})_8$ // Replace each group by octal value

$(1EF)_{16}$→ $(\textbf{0001})_2$ x 256 + $(\textbf{1110})_2$ x 16 + (1111) x 1

→ $(0001\ 1110\ 1111)_2$ // Convert to binary

→ $(\cancel{000}\ 111\ 101\ 111)_2$ // group by 3 , removed 0s in the front

→ $(\textbf{757})_8$ // Replace each group by octal value

Octal → Hexadecimal

- $(757)_8$ → $(?)_{16}$
- $(254)_8$ → $(?)_{16}$

Exercises

- Convert following Decimal numbers to HexaDecimal/Octal
  - 255_{10}
  - 256_{10}
  - 257_{10}
- Convert following HexaDecimal numbers to Decimal, Octal
  - 99_{16}
  - 100_{16}
  - 101_{16}

Recursion : programming

- Create factorial
  - 5! = 5 x 4 x 3 x 2 x 1 = **120**
  - 2! = **2**
- Provide sum of Fibonacci numbers using recursion
  - fibonacci(13)
    - 0 + 1 + 1 + 2 + 3 + 5 + 8 + 13 = **33**
  - fibonacci(3)
    - 0 + 1 + 1 + 2 + 3 = **7**

Relations

- A **relation** : connection/mapping between elements of two or more sets, Some characteristics …
  - It is a **mapping** as *shown in figure*
  - Can be written as **Ordered pairs**
    - {(1,c), (2,n),(5,a),(7,n)}
  - Not always unique
    - E.g. $y^2 = 4$ has multiple solutions (how many?)(

- Examples
  - Numerical relationship : 4+3 = 7
  - Equation : y = 2x+3
  - Geometry : two congruent triangles
  - Set theory : A is a subset of B
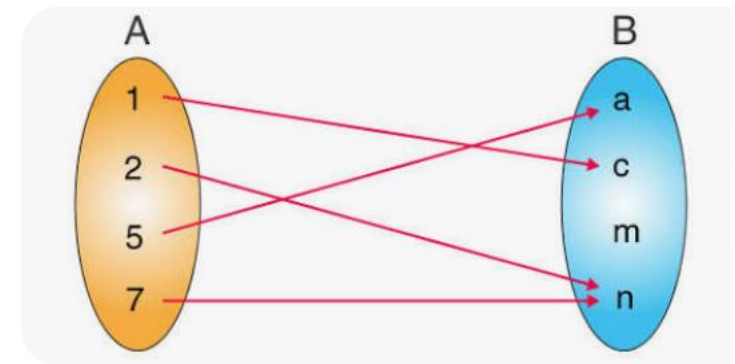
Functions , mathematical kind

- A **relation** that gives exactly **one unique output for each input**

x → [Rule] → y   :this is a function as you always get one answer

x → [Rule] ⟹ a, b, c :Not a function as you get multiple answers

- Representation
  - $f(x) = 2x+1$: this is a function
  - $g(x) = \pm 3x$: this is **not** a function , why?
- Follow up reading : pg 11-13 : <u>functions</u>

Evaluating functions

- Solving
  - Substitute variables with numerals
  - Evaluate
    - Follow PEMDAS/BODMAS

- Solve for x = 0,1,2,3
  - $f(x) = 3x + 1$
  - $g(x) = 2x^2 + 3$
  - $h(x) = x^2 + 2x + 1$

Recursive Functions

- Functions calling themselves

Fibonacci numbers	$fib: \mathbb{N} \rightarrow \mathbb{N}$ $fib(n) = \begin{cases} 0, & \text{if } n = 0 \\ 1, & \text{if } n = 1 \\ fib(n-1) + fib(n-2), & \text{if } n \geq 2. \end{cases}$
Factorial	**fact(n) = n * fact(n-1)** {given, *fact(1) = fact(0) =1*}
Lucas numbers (*same rule as Fibonacci but with different starting values*)	**l (n) = l(n-1) + l(n-2)** {given , *l(0) = 2, l(1) = 1*}

- Evaluate for 'n' = 2 , 4, 5, 6
- Follow up reading : nested functions visualized

Programming, , Test #1

- Need to know
  - Data types
  - Conditionals
    - If/else
  - Loops
    - for
  - Arrays
    - 2 dimensional
- Practice
  - [Arrays- DS (Hackerrank)](#) : 6 problems