Discussion [0b10] [2_{10}] [0x2]: Number Representation

Conversion

Decimal	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

1. Convert the following binary numbers into decimal.

11001 → 25 ($1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 1 \times 2^4$)

1001001 → 73 ($1 \times 2^0 + 1 \times 2^3 + 1 \times 2^6$)

2. Convert the following decimal numbers into binary.

12 → 0b1100

We will solve this using one method shown in lecture. First, we create a table of the powers of the base we're converting to (2), stopping once we get to a power of 2 greater than the number we are trying to convert (12).

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1

Now, we work through the table from left to right, using the following rules to convert:

1. If the number in the column is greater than the number we are converting, we put a 0 in the column and move to the next one.
2. If the number is the column is less than or equal to the number we are converting, we put a 1 in that column, and subtract that value from the number we're converting. Then, we continue through the table and convert our new value.

Let's work through that algorithm:

Since 16 is less than 12, we put a 0 in that column and move to the next one:

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1
0				

Now, we move to 8. 8 is less than 12, so we put a 1 in that column and subtract 8 from 12 to get 4. Now, we will proceed through the table to convert the remaining 4.

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1
0	1			

Now, we continue through the table the same way. Since 4 is less than or equal to the number we are trying to convert, we put a 1 in that column and our new value to convert becomes 0.

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1
0	1	1		

We continue this until we get to the end of our table, which is eventually:

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1
0	1	1	0	0

Now, we read the bottom row from left to right to get our binary number, giving us 0b1100.

64 → 0b1000000

We will solve this problem using the other method taught. You are not expected to know how to convert numbers both ways, feel free to use the way you feel most comfortable with.

In this method, we continuously divide our number by our new base and record the remainder. To help us do so, we can draw out the following table:

Base	Number	Remainder
2	64	

In each row, we divide our number by our base and record the remainder. We continue this process until our number is 0, and then we read the "Remainder" column from **bottom to top**.

We start with 64 as our number. When we divide 64 by 2, we get 32 and no remainder:

Base	Number	Remainder
2	64	
2	32	0

We continue this process below:

Base	Number	Remainder
2	64	
2	32	0
2	16	0
2	8	0

2	4	0
2	2	0
2	1	0
2	0	1

Note that in the last row, 1 divided by 2 gives us 0.5, not 0. In this method, since we are recording remainders, we round numbers in the middle column down so we don't get any decimals.

Now that we've filled out our table, we read the rightmost column from bottom to top, giving us 0b1000000.

127 → 0b1111111

3. Convert the following binary numbers into hex.

10011001 → 0x99

To convert from binary to hex, we group our digits into groups of 4 and then match those groups of 4 using the table on page 1 to their corresponding hex digit. When we group our number, we get 1001 1001, and since 1001 is 0x9, our number is 0x99.

1111011 → 0x7B

This time, our digits don't evenly split into groups of 4, so we add 0's to the beginning of our number until its digits will evenly split into groups of 4. Once we pad our number, we get 0b01111011, and when we group this into 4's, we get 0111 1011. After translating these in our table, we get 0x7B.

110000001111111111101110 → 0xCOFFEE

Limits

1. What is the biggest number that can be represented with two decimal digits?
   99

2. What is the biggest number that can be represented with three binary digits?
   0b111, which translates to 7 in decimal.

3. What is the biggest number that can be represented with four hexadecimal digits?
   0xFFFF, which translates to 65535.

4. How many different numbers can you represent using three binary digits?

We can represent all numbers from 0 to 7 (0b111), which is 8 different numbers. In general, with n digits in base b, you can represent b^n different numbers, and the max value you can represent is $b^n - 1$.

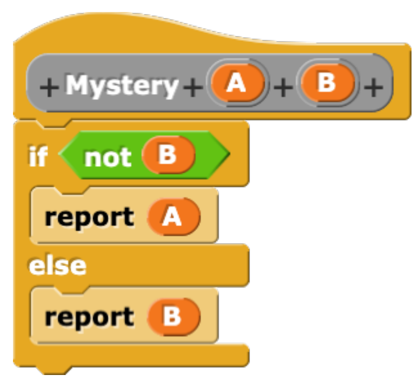
Boolean Logic

1. Fill out the truth tables for (and) and (or) below:

A	B	A and B
T	T	T
T	F	F
F	T	F
F	F	F

A	B	A or B
T	T	T
T	F	T
F	T	T
F	F	F

2. Consider the `Mystery` block on the right:

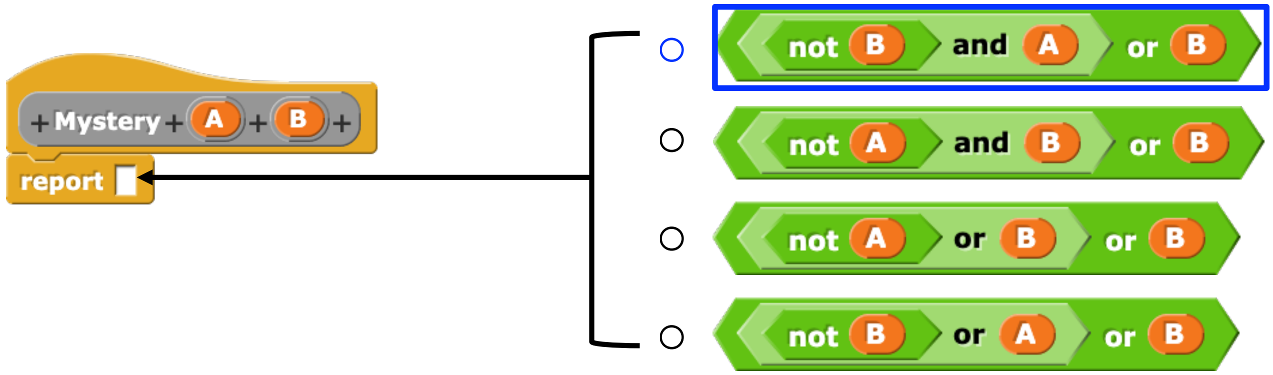


a. If Mystery outputs false, which of the following do we definitely know? (Hint: It may be helpful to draw out a truth table for Mystery to solve this problem)

A	B	`Mystery`
T	T	T
T	F	T
F	T	T
F	F	F

- ☐ A must be true
- ☐ A must be false
- ☐ B must be true
- ☐ B must be false
- ☐ None of these

b. We want to rewrite this block in one line. Which expression could we report so that our block has exactly the same behavior as the block above?



3. Fill out the table below:

	Does the value of foo affect the output of this expression?	If you answered “no” to the previous question, what will this expression output?
false and foo	no	false
false or foo	yes	
true and foo	yes	
true or foo	no	true

Challenge Problems

1. The original Pokemon are numbered 1-150. We want to store a binary encoding for all original Pokemon where each Pokemon has a binary code equivalent to their decimal number.

   a. How many bits do we need to use?
   We need 8 binary digits, since 7 digits can only represent 128 unique values, and 8 digits can represent up to 256 unique values (see the formula in the solution for #4 on “Limits”.

   b. What is the encoding for Pikachu (#25)?
   25 in binary is 11000.

   c. Ternary utilizes base 3 instead of base 2. For example, 10 in ternary is equivalent to 3 in decimal. Imagine that we wanted to store a ternary encoding for all 150 Pokemon where each Pokemon has a ternary code equivalent to their decimal number. What is the ternary encoding for Pikachu (#25)?
   We can use the algorithm we used for the second part of #2 on “Conversion.”

Base	Number	Remainder
3	25	
3	8	1
3	2	2
3	0	2

Our answer then is 221_3.