UNIVERSITY OF CALIFORNIA, BERKELEY
Department of Electrical Engineering and Computer Sciences
Computer Science Division

CS10 Spring 2024 **TA: Vedansh Malhotra**

SOLUTIONS

Discussion-2: Number Representation, Control, Boolean Operators

Q1. Number Representation

First, your TA will do a mini-lecture on number-representation. Please pay attention!
Now, let's solve the following problems (yes, they're similar to the lecture-quiz!)

Problems:

1. How many digits do we have in the base 14 NRS? What are they?
14 digits: 0, 1, 2, … , 9, A, B, C, D

1.1 If a numeral is valid in Base-11, is it necessarily valid in Base-3?
No, consider the numeral: 61A

1.2 If a numeral is valid in Base-5, is it necessarily valid in Base-3?
No, consider the numeral: 24

1.3 If a numeral is valid in Base-3, is it necessarily valid in Base-7?
Yes, the valid digits in base-3 are a subset of the valid digits in base-7.

2. What's the value of 0b1101 in decimal?
13

3. $1232 = x_{16}$. Find the value of x.
4D0

4. Convert 1232 to binary.
0b10011010000

5. Convert 0xA0C to binary.
0b101000001100

6. What's the value of 0xCAFE in decimal?
51966

7. What's the value of 0x1011 in decimal?
4113

8. Convert 0b11010111 to hexadecimal.
0xD7

9. Convert 0xDEADBEEF to hexadecimal.
0xDEADBEEF

10. Convert 164_7 to base-9.

115

11. Convert 164_9 to base-7.

256

12. A Number-Representation System with a larger base will generally use a numeral with fewer digits to represent the same number.

13.

In the unsigned representation shown in lecture, the largest value we can represent using **n >= 1** bits in binary is __(A)__, and the number of unique values we can represent is __(B)__. Your answers must be expressions in terms of **n**, and you should simplify them as much as possible.

A: [ ] B: [ ]

Hint: Try n = 3, n = 4, …

A: $2^n - 1$

B: 2^n

Q2. Boolean Operators

First, your TA will do a mini-lecture on boolean operators and truth-tables. There will be new information here that isn't covered in lectures, please pay attention and ask questions!

Modeling:

1. Write boolean expressions that model the following scenarios:
A is a boolean variable that denotes whether I fed Alonzo.
B is a boolean variable that denotes whether Dan fed Alonzo.

- Write a boolean expression that returns True iff at least one of us fed Alonzo.
  A OR B

- Write a boolean expression that returns True iff both of us fed Alonzo.
  A AND B

- Write a boolean expression that returns True iff none of us fed Alonzo. First, write it without using OR operators. Then, write it without using AND operators.
  NOT A AND NOT B ≡ NOT (A OR B)

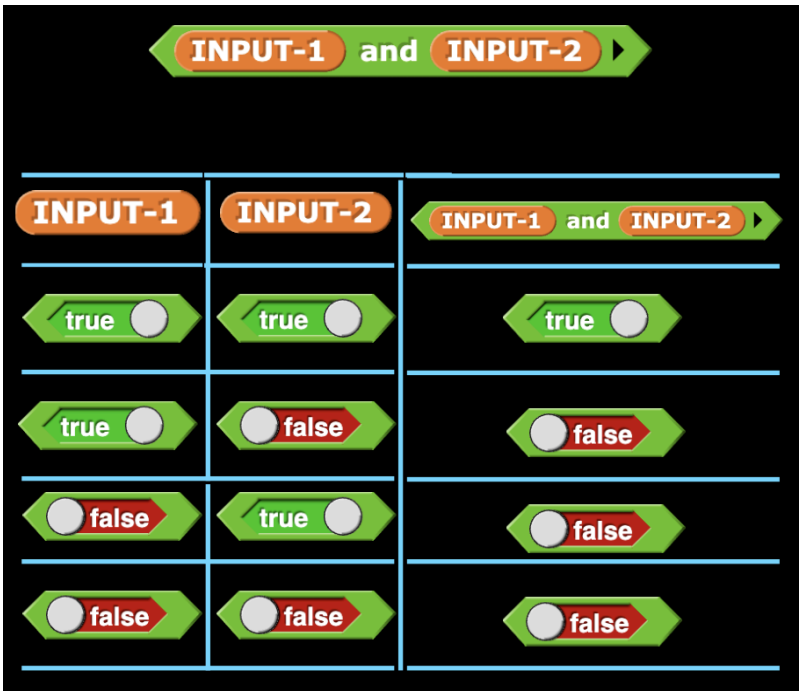
- Write a boolean expression that returns True iff exactly one of us fed Alonzo.
  (NOT (A AND B)) AND (A OR B) that is, not both but at least one!

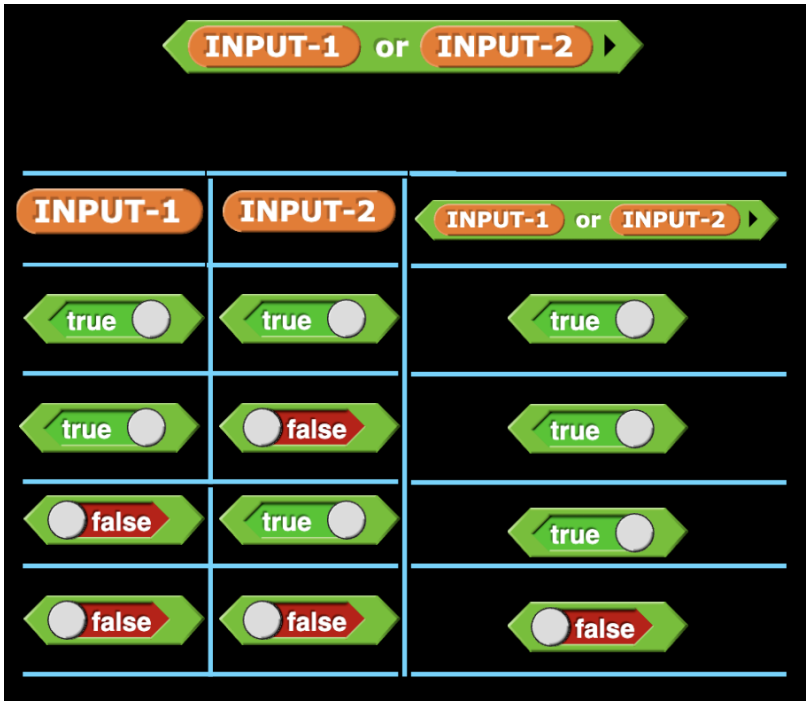
Truth Tables:

1. Fill out the following Truth-Table:

A	B	A and B

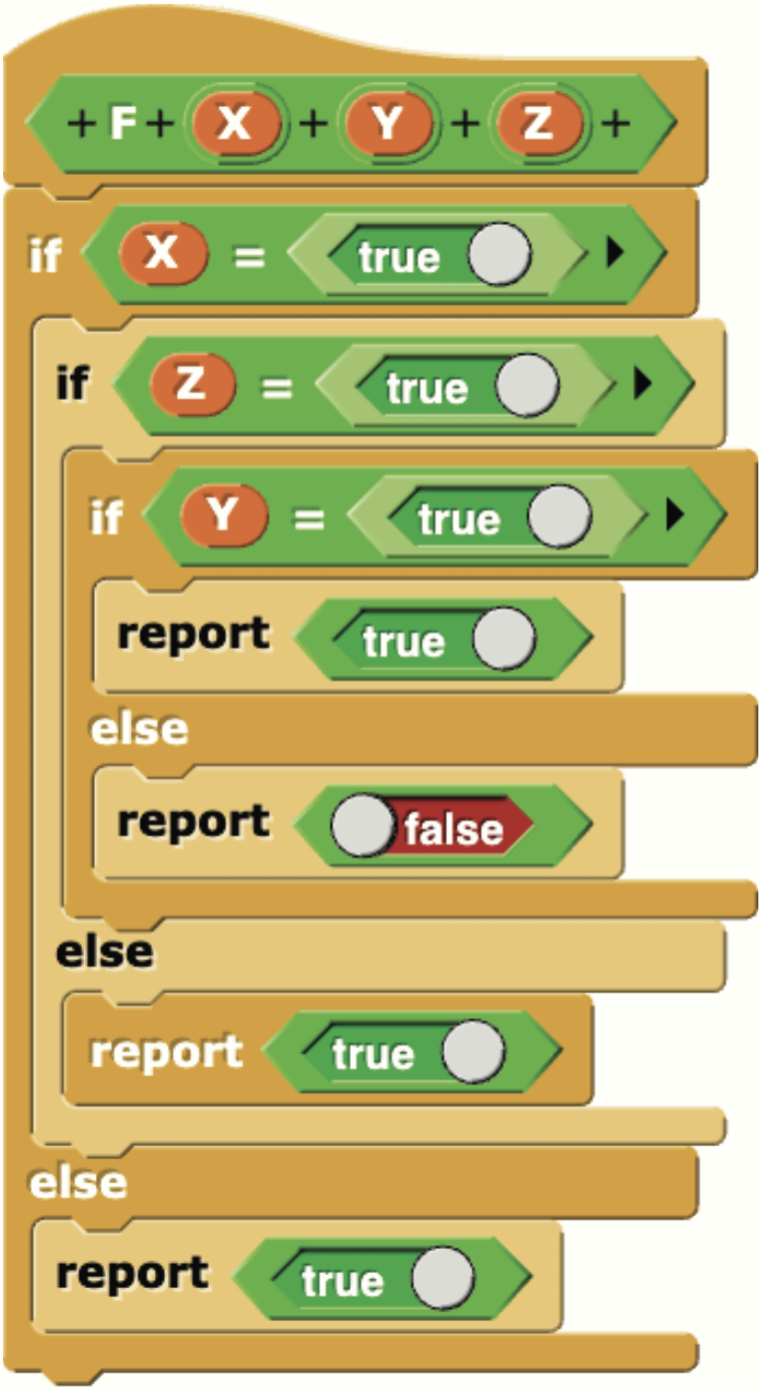
A	B	A or B


INPUT-1 and INPUT-2
INPUT-1
INPUT-2
INPUT-1 and INPUT-2
true
true
true
true
false
false
false
true
false
false
false
false


INPUT-1 or INPUT-2
INPUT-1
INPUT-2
INPUT-1 or INPUT-2
true
true
true
true
false
true
false
true
true
false
false
false

2.

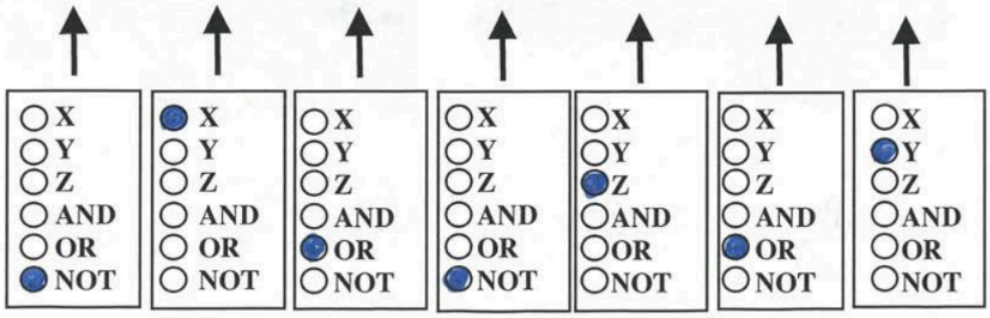
Consider the following script, wherein `X`, `Y`, and `Z` are boolean variables.



X	Y	Z	F(X, Y, Z)
T	T	T	T
T	T	F	T
T	F	T	F
T	F	F	T
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

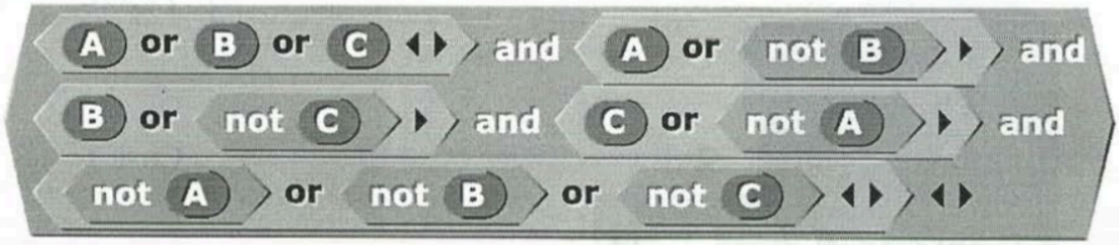
2. We now wish to rewrite the predicate block in the previous page using just one boolean expression. Please fill out the appropriate bubbles such that the resulting boolean expression is equivalent to the definition of F shown in the previous page.

F(X, Y, Z) = (NOT X) OR (NOT Z) OR Y



19. For questions 19 and 20:

Suppose that A, B, and C are boolean variables (`True` or `False`.) Is there some assignment of A, B, and C such that the following expression evaluates to `True`? If yes, fill out the **table** with a satisfying assignment (choose `True` / `False` values for A, B, and C). If not, then fill the bubble next to "Impossible"

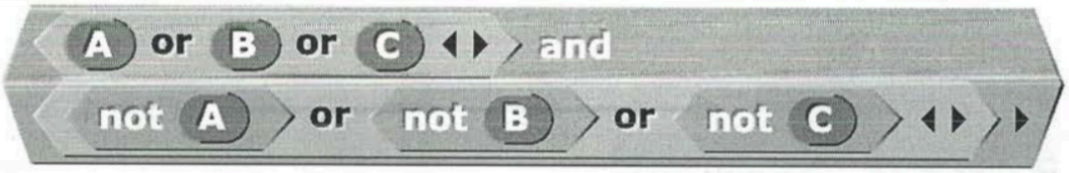


A = True ⇔ C = True ⇔ B = True
A = False ⇔ B = False ⇔ C = False

● Impossible

A	B	C
○ True ○ False	○ True ○ False	○ True ○ False

20.



○ Impossible

At least one set to T & 1 to F

A	B	C
● True ○ False	● True ○ False	○ True ● False

Q3. Iteration + Control

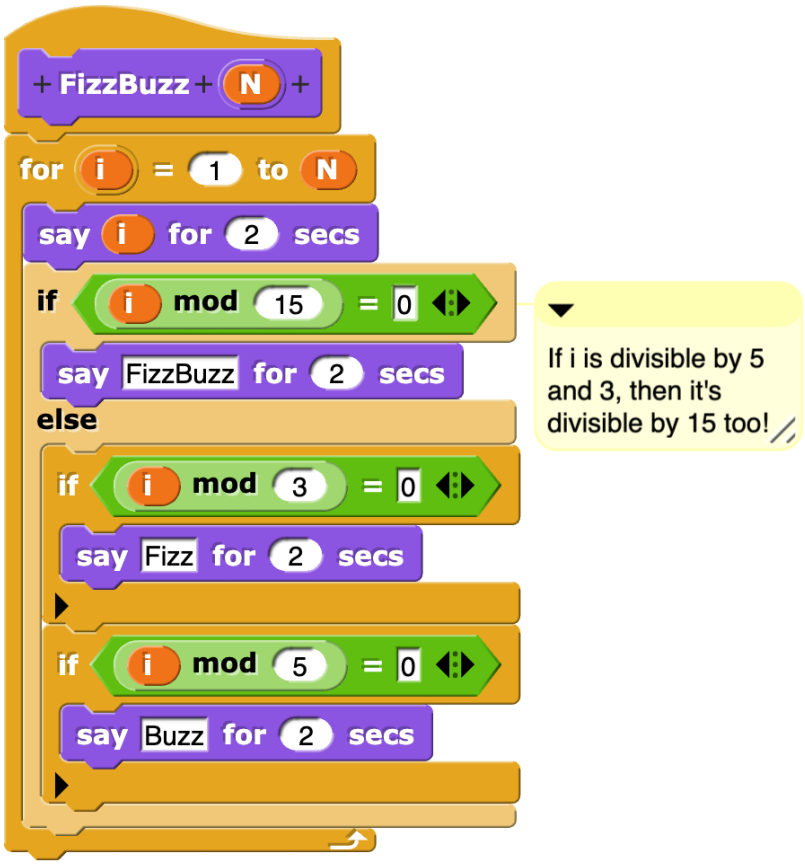
First, your TA will do a mini-lecture on different ways to accomplish iteration and control structures in Snap*!* Please pay attention and ask any questions you might have.

Problems:

1. FizzBuzz

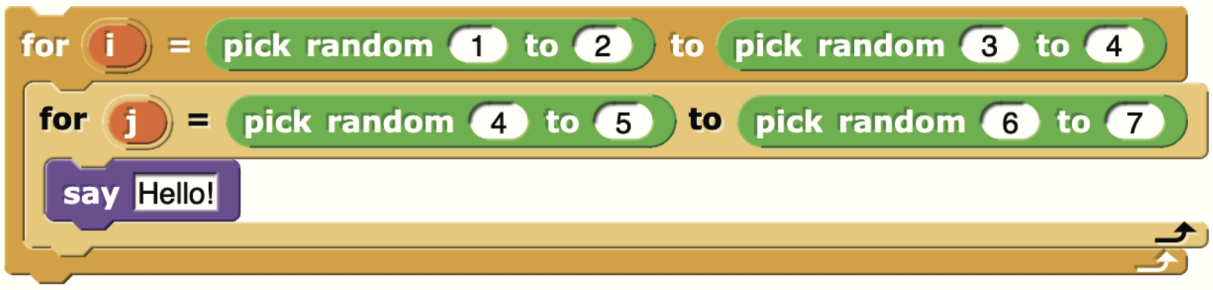
Write a block called **`FizzBuzz`**, which takes in an integer N. It should say all the integers from one to N, and additionally say “Fizz” if an integer is divisible by three, “Buzz” if an integer is divisible by five, and “FizzBuzz” if an integer is divisible by both three and five.

As an example, **`FizzBuzz(15)`** should say the following in order:
1, 2, 3, Fizz, 4, 5, Buzz, 6, Fizz, 7, 8, 9, Fizz, 10, Buzz, 11, 12, Fizz, 13, 14, 15, FizzBuzz



2.

Consider the following block:



The minimum number of times Snap*!* will say "Hello!" is: ☐

The maximum number of times Snap*!* will say "Hello!" is: ☐

Minimum: 4 Maximum: 16