

## Mathematics 3670: Lab 3 Hints + Python Tips

Dr. William Slough

Mathematics and Computer Science Department  
Eastern Illinois University

Fall 2011

## Iterating over a list in Python

```
# Establish a sample list of values
A = [10, 20, 30, 40, 50, 60, 70]

# Obtain all the legal index values for A
indexValues = range(len(A))

# Process the elements of A, in left-to-right order
for i in indexValues:
    print(A[i])

# Process the elements of A, in right-to-left-order
indexValues.reverse()
for i in indexValues:
    print(A[i])
```

## Generating a list of the appropriate size

```
def add(A, B):
    """
    Performs binary addition on the bit strings A and B,
    yielding the sum. Assumes both arguments have the
    same number of bits.
    """

    # Create a list of 0's of the correct size
    sum = len(A) * [0]

    # Perform the addition, saving the bits in sum
    ... INCOMPLETE ...

    return sum
```

## Thinking about binary addition

One stage involves adding two bits and a carry:

$$x + y + c$$

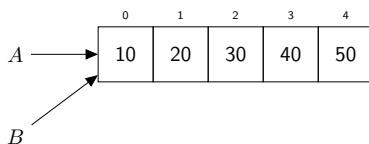
where  $x, y$ , and  $c$  are bits. From these three bits, we need a **sum** bit and a **carry** bit to pass on to the next stage.

$x$	$y$	$c$	carry	sum
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

The Python `/` and `%` operators are made to order for this task!

## The assignment operator used with lists

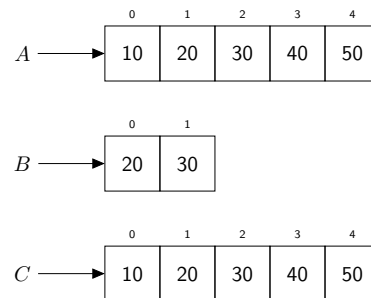
```
A = [10, 20, 30, 40, 50]
B = A
```



Be careful ... making a change to A also changes B!

## The Python "slice"

```
A = [10, 20, 30, 40, 50]
B = A[1:3]
C = A[ : ]
```



B and C are **copies** of a slice of A — a subsequent change to A leaves B and C unchanged.