**CSE1100 – Programming Concepts with Python**

**Programming Concepts Lab Report**

**Fall 2021**

*By*

Cael Shoop

*Computer Science, B.S.*

*cshoop2018@my.fit.edu*

November 14, 2021

Problem Set 2

Teaching Instructor:

Stefan Joe-Yen, Ph.D.

# 1. Problem Statement

For L1, we simply had to print out a range of numbers with certain values being replaced with Fizz, Buzz, and Bang. For L3, it was required to create a reverse polish notation calculator.

# 2. Requirements

**L1**

1. The program must print the first 200 positive integers
   1. They must be in 20 lines of 10 comma separated values
2. If a number is a multiple of 3, it must be replaced with “Fizz”
3. If a number is a multiple of 5, it must be replaced with “\*Buzz\*”
4. If a number is a multiple of both 3 and 5, it must be replaced with “BANG!!”

**L3**

1. The program must be a Reverse Polish Notation calculator
2. It must support addition, multiplication, subtraction, and division of decimal values

# 3. Software Construction (Annotated Python Code)

**L1**

'''Challenge L1 written by Cael Shoop for Problem Set 2 in CSE1100.'''

def main():

for count in range(200):

ii = count + 1 # Because count is 0-199, ii is 1-200

value = str(ii) # Change iterable to a string

if ii % 3 == 0 and ii % 5 != 0: # If multiple of 3 and not 5, print Fizz

print('Fizz', end='')

elif ii % 5 == 0 and ii % 3 != 0: # If multiple of 5 and not 3, print Buzz

print('\*Buzz\*', end='')

elif ii % 3 == 0 and ii % 5 == 0: # If multiple of 3 and 5, print Bang

print('BANG!!', end='')

else: # If not multiple of 3 or 5, print value

print(value, end='')

if ii % 10 != 0: # If not last value of line, don't print newline

print(', ', end='')

else: # If last value of line, print newline

print()

if \_\_name\_\_ == '\_\_main\_\_':

main()

**L3**

'''Challenge L3 written by Cael Shoop for Problem Set 2 in CSE1100.'''

import math

import operator

def main():

while True:

# Creates a list out of the input separated by spaces

expression = input('Enter an Expression (enter \'Q\' to quit): ').split(' ')

if expression == 'Q' or expression == 'q':

exit()

values = [] # List of values that will be used during operations as well

ops = {'+':operator.add, # Predefined operators from operator library

'-':operator.sub, # This tuple allows for easy calling later on

'\*':operator.mul,

'/':operator.floordiv}

for char in expression: # Iterates through the expression list

if char.isnumeric(): # Numbers are inserted into the values list

values.insert(0, char)

else:

if len(values) < 2: # Throws an error if operands do not match up with operators

print('Error: Not enough operands in expression.')

exit()

else:

if len(char) == 1: # Does the specified operation on the specified operands

op0 = float(values.pop(1))

op1 = float(values.pop(0))

ans = ops[char](op0, op1)

values.insert(0, str(ans))

else: # Uses math.radians on specified operand

op0 = float(values.pop(0))

ans = ops[char](math.radians(op0))

values.insert(0, str(ans))

ans = int(ans) # Convert the answer to an integer

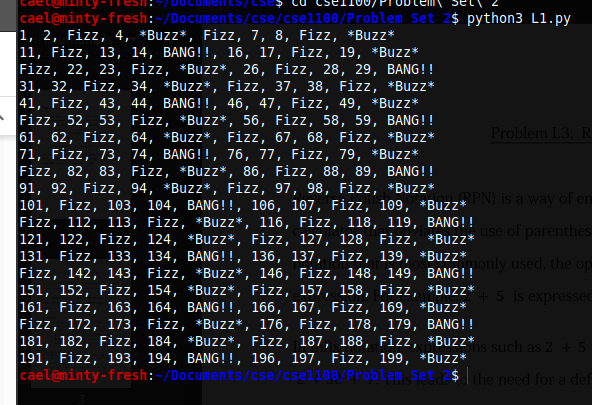
print(f'Result: {ans}')

if \_\_name\_\_ == '\_\_main\_\_':

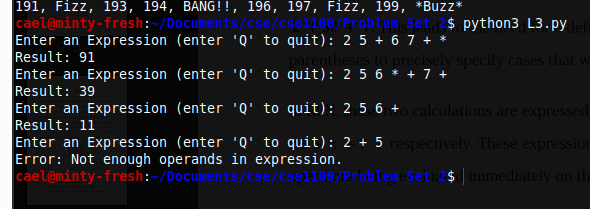
main()

# 4. Software Testing

**L1**

****

**L3**

****

# 5. Self-Reflection

I chose to do two of the problems because I had some time and it was really rewarding to write some Python code that actually works, unlike my Senior Project code at the moment. These were pretty easy to complete and I was able to get them done very quickly.