



## LAB WEEK 09

1. Assume we are given two lists named vect1 and vect2. Write a method that returns a list containing the meaningful values of  $\text{vect1}[i]/\text{vect2}[i]$ . Use exceptions and raising errors in your code as much as possible. For instance, you should throw an exception if entries are not float or integer. Similarly, you can throw an exception if the lists are not of same size. In these instances, one exception raise can be `raise ValueError('getRatios called with bad arguments')`
2. Write a method to read grades from a file and calculates the median of the grades. Use exceptions in your code as much as possible. For instance, you should throw an exception if entries are not float or integer. Similarly, you can throw an exception if the file does not exist. If there is an exception, print the exception to the screen.
3. Write a function that asks for an integer and prints the square of it. Use a while loop with a try, except, else block to account for incorrect inputs.
4. Debug the following program by using breakpoints in your IDE.

```
import math

def demo(a, b, c):
    d = b ** 2 - 4 * a * c
    if d > 0:
        disc = math.sqrt(d)
        root1 = (-b + disc) / (2 * a)
        root2 = (-b - disc) / (2 * a)

        return root1, root2
    elif d == 0:
        return -b / (2 * a)
    else:
        return "This equation has no roots"

if __name__ == '__main__':
    while True:
        a = int(input("a: "))
        b = int(input("b: "))
        c = int(input("c: "))
        result = demo(a, b, c)
        print(result)
```

5. Below is buggy code to get a list of primes less than equal to a given number n. Try to debug it, and fix it such that it properly returns list of primes.

```
def primes_list_buggy(n):  
    """  
    input: n an integer > 1  
    returns: list of all the primes up to and including n  
    """  
  
    # initialize primes list  
    if i == 2:  
        primes.append(2)  
    # go through each elem of primes list  
    for i in range(len(primes)):  
        # go through each of 2...n  
        for j in range(len(n)):  
            # check if not divisible by elem of list  
            if i%j != 0:  
                primes.append(i)
```