ASSIGNMENT ANSWERS:

1A.

```
def number_game():
    user_turn = True
    max_num = 0
    while max_num < 20:
        if user_turn:
            user_input = input("Enter 1, 2, or 3 digits starting from {}:".format(max_num+1))
            user_nums = list(map(int, user_input.split()))
            if len(user_nums) < 1 or len(user_nums) > 3:
                print("Invalid input. Please enter 1, 2, or 3 digits.")
                continue
            if user_nums[0] != max_num + 1:
                print("Invalid input. Please start from the next number in sequence.")
                continue
            max_num = user_nums[-1]
            user_turn = False
        else:
            computer_nums = list(range(max_num + 1, max_num + 4))
            max_num = computer_nums[-1]
            print("Computer plays: {}".format(computer_nums))
            user_turn = True
    if user_turn:
        print("Computer wins!")
    else:
        print("User wins!")
number_game()
```

2A

```
def ncr(n, r):
    if r > n - r:
        r = n - r
    res = 1
    for i in range(r):
        res *= (n - i)
        res /= (i + 1)
    return int(res)
def print_pascal_triangle(n):
    for i in range(n):
        for j in range(n - i):
            print(" ", end="")
        for k in range(i + 1):
            print(ncr(i, k), end=" ")
        print()
# Input the number of rows
n = int(input("Enter the number of rows: "))
print_pascal_triangle(n)
```

3A

```
def print_repeated_elements(nums):
    freq_dict = {}
    for num in nums:
        if num in freq_dict:
            freq_dict[num] += 1
        else:
            freq_dict[num] = 1

    for num, freq in freq_dict.items():
        print(f"Element {num} has come {freq} times")
nums = [1,2,3,4,5,2,3,3,4,4,5,4,5,5,6,6,2,3]
print_repeated_elements(nums)
```

4A

```
def read_matrix_from_file(filename):
    with open(filename, 'r') as file:
    return matrix
def add_matrices(A, B):
    result = [[A[i][j] + B[i][j] for j in range(len(A[0]))] for i in range(len(A))]
    return result
def print_matrix(matrix):
    for row in matrix:
        print(" ".join(map(str, row)))
# Read matrices A and B from file
A = read_matrix_from_file("matrixA.txt")
B = read_matrix_from_file("matrixB.txt")
# Perform matrix addition
result = add_matrices(A, B)
# Print the result
print("Matrix A:")
print_matrix(A)
print("Matrix B:")
print_matrix(B)
print("Result of A + B:")
print_matrix(result)
```

5A

```
class Fraction:
    def __init__(self, numerator, denominator):
        self.numerator = numerator
        self.denominator = denominator
    def __add__(self, other):
        new_numerator = self.numerator*other.denominator + other.numerator * self.denominator
        new_denominator = self.denominator * other.denominator
```

```
        return Fraction(new_numerator, new_denominator)
    def __str__(self):
        return f"{self.numerator}/{self.denominator}"
frac1 = Fraction(1, 4)   #1,4 are examples
frac2 = Fraction(1, 8)   #1,8 are examples
frac3 = frac1 + frac2
print(frac3)
```