IT251 Lab 2 Problems -- 15 Jan 2020

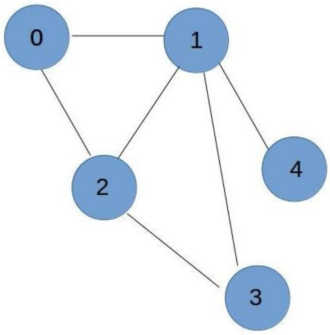
Problem 1: Write a python program to implement the **Fraction** class. First write a constructor for the class and then implement the following methods in it:

i. *__str__ (self):* returns a string representation of the fraction *self*.
ii. *inverse(self):* returns the inverse of the fraction *self*.
iii. *add(self,other):* adds fraction *other* to the fraction *self* and returns the result.
iv. *multiply(self,other):* multiplies fraction *other* to the fraction *self* and returns the result.
v. *equal(self):* returns True if the value of both the fractions are the same, False otherwise.

Both the add and multiply methods should return the reduced form of the fraction. For example if the result of an addition is 25/30, the method should return 5/6. Test each of your methods in your main function. Overload the +, * and == operators to work on fraction objects.

Problem 2: Complete the python program 'stack.py' to implement a **Stack** class. Since python does not have arrays, we will use the in-built *list* data structure to store the elements in the stack.

Problem 3: Write a program to store (and then print) the adjacency matrix and adjacency list representation of an undirected graph. Read in the graph from the user. A sample input/output of the program is shown for the graph below.



Input:

```
Enter the number of vertices: 5
Enter the edges:
0 1
0 2
1 2
1 3
1 4
2 3
```

Output:

```
The adjacency matrix is
0  1  1  0  0
1  0  1  1  1
1  1  0  1  0
0  1  1  0  0
0  1  0  0  0
```

```
The adjacency list is:
Vertex 0: 1, 2
Vertex 1: 0, 2, 3
Vertex 2: 0, 1, 3
Vertex 3: 1, 2
Vertex 4: 1
```