1. (4 points) Consider the following function from FlowEdge.java. This function is used to allocate flow to an edge in a residual graph. Fill in the blanks with the appropriate type of edge.

```java
public void addResidualFlowTo(int vertex, double delta) {

    // allocate flow to a _______________ edge
    if (vertex == v) flow -= delta;

    // allocate flow to a _______________ edge
    else if (vertex == w) flow += delta;

    else throw new IllegalArgumentException("Illegal endpoint");

    if (Double.isNaN(delta))
        throw new IllegalArgumentException("Change in flow = NaN");
    if (!(flow >= 0.0))
        throw new IllegalArgumentException("Flow is negative");
    if (!(flow <= capacity))
        throw new IllegalArgumentException("Flow exceeds capacity");
}
```

**Answer:**

2 Backwards
2 Forward
2. (12 points) Consider the RSA public key \( n = 55, e = 17 \). Crack this key and give the values of the following:

- \( p \) & \( q \)
- \( \phi(n) \)
- \( d \)

**ANSWER:**

4. \( p \) & \( q = 11 \) & 5
2. \( \phi(n) = 40 \)
6. \( d = 33 \)
   - Will get -7 back from \( \text{XGCD}(40, 17) \), \( d \) should be positive, \( -7 \) (mod \( \phi(n) \)) is 33
   - -7 is worth 5 points
   - Stating that \( \text{XGCD}(40, 17) \) should be used to find \( d \) or attempting \( \text{XGCD}(40, 17) \) is 3 points
3. (9 points) Use Dijkstra’s algorithm to determine the shortest path from vertex A to vertex G in the following graph. Be sure to state:

- The shortest path from A to G
- The weight of that path
- The order in which you visit the vertices of the graph in finding the shortest path

**Answer:**

3. Shortest path: A D F G
2. Weight of 20
4. Visited order: A B C D E F G