1) I have the following Spec# program that computes an array of squares. I want to eliminate the usage of multiplication in the program. Make the appropriate modifications to the program text and introduce the appropriate new invariants for the new variables you introduce. [Rather than rewrite the whole program, you can mark it to indicate clearly the new and modified text]

```specsharp
class Q1 {
    void Squares (int[] a)
        modifies a[*];
        ensures forall{int i in (0: a.Length); a[i] == i*i};
    {
        int n = 0;
        while (n < a.Length)
            invariant forall{int i in (0: n); a[i] == i*i};
        {
            a[n] = n*n;
        }
    }
}
```

2) Show how the following method to find minimum value should be annotated in Spec#

```specsharp
class Q1 {
    int minimum (int[] a)
    {
        int n = 1;
        int result = a[0];
        while (n < a.Length) {
            if (a[n] < result) {
                result = a[i];
            }
        }
    }
}
```
3) What is the thesis statement (central claim) of the paper: Formal vs Agile: Survival of the Fittest? and what does this claim have to do with the Agile Manifesto virtues of:
   Individuals and interactions over processes and tools
   Working software over comprehensive documentation
   Customer collaboration over contract negotiation
   Responding to change over following a plan