Precision of value-contexts based context-sensitive analysis:

1. Precision of value-contexts is equivalent to the Infinite-CFA.
2. Ideally, if we ignore the class fields, the method return value depends on the parameters.
3. The value contexts allows us to segregate the call graph into and out of a function based on the values at the point of invocation.
4. The current sign analysis of vasco stores the sign of input as relevant values. We could have further improved the precision by taking the exact value of the input parameter as relevant values. This further allows us to do constant propagation and opens many more opportunities.

Termination Guarantee:

1. The termination is guaranteed if the number of contexts is finite.
2. In case of value contexts, number of value contexts is limited by the number of call sites. Thus the termination is guaranteed.

Design Principles:

1. The VASCO project is using the maven build system to provide convenience.
2. The project uses Single Responsibility Principle. Each class is responsible for a single task/data.
3. Each class is parametrized, allowing users to use the framework in a much more flexible way.

Outputs:

Test 1:

1. The process function can only return 1. Thus the results for x,p,q in the main function are straight forward. {x: -, p: +, q: +}
2. Next, we have multiplication. q and p are of same sign, so {l: + }
3. Next we have addition, x and l are of opposite signs, so {x: _|_ }

Test 2:

1. method op can return all three signs, but the signs it can possibly return are restricted by the input parameters.
2. If our analysis was context insensitive, we would not have been able to get more precise signs based on the input parameters.
3. Ideally signs should be:
   {x2: +, x3: -, x4: +, x5: 0, x6: 0}
4. But sign analysis gives us:
   {x2: +, x3: _|_, x4: +, x5: 0, x6: _|_ }
   This is because we are not able to resolve the exact Control Flow path taken in the op method.

Test 3:

1. In this test case, the sign returned by recurNeg should be positive if input is negative and vice versa.
2. staticField method depends on a static field and parameter p.
3. Ideally, the signs should be:
   {x: -, a: +, b: -, y: +, c: 0},
   if we ignore take im precision of if condition into consideration, the signs should be:
   {x: -, a: +, b: -, y: _|_, c: 0}
4. The signs returned by Sign Analysis are:
   { x: -, a: _|_, b: _|_. y: _|_, c: _|_ }
   The imprecision is because static fields are unhandled.