Correct Algorithm and Complete Postcondition (1.1)

```
------- MODULE ExampleModule -------
EXTENDS Integers, Sequences, TLC
CONSTANT input
--algorithm SomeAlgo {
  variables
    output = ..., ...
  {
    \* Preconditions
    assert Q;

    (* Implementation in PlusCal *)
    Imp.

    \* Postcondition 1
    assert R1;
    \* Postcondition 2
    assert R2;
  }
}
```

Q complete?

Correct Algorithm and Complete Postcondition (1.2)

```
------- MODULE ExampleModule -------
EXTENDS Integers, Sequences, TLC
CONSTANT inputSeq, inputVal
--algorithm BinarySearch {
  variables
    output = FALSE, …
  {
    \* Preconditions
    assert Q;

    (* Implementation in PlusCal *)
    Imp.

    \* Postcondition 1
    assert R1;
    \* Postcondition 2
    assert R2;
  }
}
```

Q complete?

Correct Algorithm and Complete Postcondition (2.1)

```
------- MODULE ExampleModule -------
EXTENDS Integers, Sequences, TLC
CONSTANT input
(* algorithm SomeAlgo {
  variables
    output = ..., ...
  {
    \* Preconditions
    assert Q;

    (* Implementation in PlusCal *)
    Imp.

    \* Postcondition 1
    assert R1;
    \* Postcondition 2
    assert R2;
  }
}
```

Imp. correct?

Correct Algorithm and Complete Postcondition (2.2)

```
------- MODULE ExampleModule -------
EXTENDS Integers, Sequences, TLC
CONSTANT inputSeq, inputVal
(* algorithm BinarySearch {
  variables
    output = FALSE, ...
  {
    \* Preconditions
    assert /* inputSeq is sorted */;

    (* Implementation in PlusCal *)
    Imp.

    \* Postcondition 1
    assert /* inputSeq unchanged */;
    \* Postcondition 2
    assert /* output computed correctly */;
  }
}
```

Imp. correct?

Correct Algorithm and Complete Postcondition (3.1)

```
------- MODULE ExampleModule -------
EXTENDS Integers, Sequences, TLC
CONSTANT input
(* algorithm SomeAlgo {
  variables
    output = ..., ...
  {
    \* Preconditions
    assert Q;

    (* Implementation in PlusCal *)
    Imp.

    \* Postcondition 1
    assert R1;
    \* Postcondition 2
    assert R2;
  }
}
```

R1 and R2 complete?

Correct Algorithm and Complete Postcondition (3.2)

```
------- MODULE ExampleModule -------
EXTENDS Integers, Sequences, TLC
CONSTANT inputSeq, inputVal
(* algorithm BinarySearch {
  variables
    output = FALSE, ...
  {
    \* Preconditions
    assert /* inputSeq is sorted */;

    (* Implementation in PlusCal *)
    Imp.

    \* Postcondition 1
    assert R1;
    \* Postcondition 2
    assert R2;
  }
}
```

R1 and R2 complete?

Predicate Logic: Exercise 1

Consider the following predicate:

$$\forall\, x, y \bullet x \in \mathbb{N} \land y \in \mathbb{N} \Rightarrow x * y > 0$$

Choose **all** statements that are **correct**.

1. It is a theorem, provable by (5, 4).
2. It is a theorem, provable by (2, 3).
3. It is not a theorem, witnessed by (5, 0).
4. It is not a theorem, witnessed by (12, -2).
5. It is not a theorem, witnessed by (12, 13).