PRE- AND POST-CONDITIONS OF AN ALGORITHM
LOOP INVARIANTS
LOOP INVARIANT THEOREM

ALGORITHM:

The word "algorithm" refers to a step-by-step method for performing some action. A computer program is, similarly, a set of instructions that are executed step-by-step for performing some specific task. Algorithm, however, is a more general term in that the term program refers to a particular programming language.

INFORMATION ABOUT ALGORITHM:

The following information is generally included when describing algorithms formally:

1.The name of the algorithm, together with a list of input and output variables.
2.A brief description of how the algorithm works.
3.The input variable names, labeled by data type.
4.The statements that make the body of the algorithm, with explanatory comments.
5.The output variable names, labeled by data type.
6.An end statement.

THE DIVISION ALGORITHM

THEOREM (Quotient-Remainder Theorem):

Given any integer n and a positive integer d, there exist unique integers q and r such that $n = d \cdot q + r$ and $0 \le r < d$.

Example:

a) $n = 54$, $d = 4$ $54 = 4 \cdot 13 + 2$; hence $q = 13$, $r = 2$
b) $n = -54$, $d = 4$ $-54 = 4 \cdot (-14) + 2$; hence $q = -14$, $r = 2$
c) $n = 54$, $d = 70$ $54 = 70 \cdot 0 + 54$; hence $q = 0$, $r = 54$

ALGORITHM (DIVISION):

{Given a nonnegative integer a and a positive integer d, the aim of the algorithm is to find integers q and r that satisfy the conditions $a = d \cdot q + r$ and $0 \le r < d$.
This is done by subtracting d repeatedly from a until the result is less than d but is still nonnegative.
The total number of d's that are subtracted is the quotient q. The quantity $a - d \cdot q$ equals the remainder r.}

Input: a {a nonnegative integer}, d {a positive integer}
Algorithm body: $r := a$, $q := 0$
{Repeatedly subtract d from r until a number less than d is obtained. Add 1 to d each time d is subtracted.}

while $(r \ge d)$
$r := r - d$ $q := q + 1$
end while
Output: q, r
end Algorithm (Division)

TRACING THE DIVISION ALGORITHM

Example:

Trace the action of the Division Algorithm on the input variables $a = 54$ and $d = 11$

Solution

Iteration Number

Variable Names

	0	1	2	3	4
a	54				
d	11				
r	54	43	32	21	10
q	0	1	2	3	4

PREDICATE

Consider the sentence
"Aslam is a student at the University."
let *P* stand for the words

"is a student at the University"

and let *Q* stand for the words

"is a student at."

Then both *P* and *Q* are *predicate symbols*.
The sentences "x is a student at the University" and "x is a student at y" are symbolized as $P(x)$ and $Q(x, y)$, where x and y are predicate variables and take values in appropriate sets. When concrete values are substituted in place of predicate variables, a statement results.

DEFINITION:

A predicate is a sentence that contains a finite number of variables and becomes a statement when specific values are substituted for the variables.
The domain of a predicate variable is the set of all values that may be substituted in place of the variable.

PRE-CONDITIONS AND POST-CONDITIONS:

Consider an algorithm that is designed to produce a certain final state from a given state. Both the initial and final states can be expressed as predicates involving the input and output variables.
Often the predicate describing the initial state is called the **pre-condition of the algorithm** and the predicate describing the final state is called the **post-condition of the algorithm**.

EXAMPLE:

1.Algorithm to compute a product of two nonnegative integers
pre-condition: The input variables m and n are nonnegative integers.
pot-condition: The output variable p equals $m \cdot n$.

2.Algorithm to find the quotient and remainder of the division of one positive integer by another
pre-condition: The input variables a and b are positive integers.
pot-condition: The output variable q and r are positive integers such that

$$a = b \cdot q + r \text{ and } 0 \leq r < b.$$

3.Algorithm to sort a one-dimensional array of real numbers
Pre-condition: The input variable $A[1], A[2], \ldots A[n]$ is a one-dimensional array of real numbers.
post-condition:The input variable $B[1], B[2], \ldots B[n]$ is a one-dimensional array of real numbers with same elements as $A[1], A[2], \ldots A[n]$ but with the property that $B[i] \leq B[j]$ whenever $i \leq j$.

THE DIVISION ALGORITHM:

[pre-condition: a is a nonnegative integer and
d is a positive integer, $r = a$, and $q = 0$]
while $(r \geq d)$
1. $r := r - d$
2. $q := q + 1$

end while
[post-condition: q and r are nonnegative integers
with the property that $a = q \cdot d + r$ and $0 \leq r < d$.]

LOOP INVARIANTS:

The method of loop invariants is used to prove correctness of a loop with respect to certain pre and post-conditions. It is based on the principle of mathematical induction.
[pre-condition for loop]

while (G)
[Statements in body of loop. None contain branching statements that lead outside the loop.]

end while[post-condition for loop]

DEFINITION:

A loop is defined as **correct with respect to its pre- and post-conditions** if, and only if, whenever the algorithm variables satisfy the pre-condition for the loop and the loop is executed, then the algorithm variables satisfy the post-condition of the loop.

THEOREM:

Let a **while** loop with guard G be given, together with pre- and post conditions that are predicates in the algorithm variables.
Also let a predicate $I(n)$, called the **loop invariant**, be given. If the following four properties are true, then the loop is correct with respect to its pre- and post-conditions.

I.Basis Property: The pre-condition for the loop implies that $I(0)$ is true before the first iteration of the loop.

II.Inductive property: If the guard G and the loop invariant $I(k)$ are both true for an integer $k \geq 0$ before an iteration of the loop, then $I(k + 1)$ is true after iteration of the loop.

III.Eventual Falsity of Guard: After a finite number of iterations of the loop, the guard becomes false.

IV.Correctness of the Post-Condition: If N is the least number of iterations after which G is false and $I(N)$ is true, then the values of the algorithm variables will be as specified in the post-condition of the loop.
PROOF:
Let I(n) be a predicate that satisfies properties I-IV of the loop invariant theorem.
Properties I and II establish that:
For all integers n $\geq$ 0, if the while loop iterates n times, then I(n) is true.
Property III indicates that the guard G becomes false after a finite number N of iterations.
Property IV concludes that the values of the algorithm variables are as specified by the post-condition of the loop.