CORRECTNESS OF:
LOOP TO COMPUTE A PRODUCT
THE DIVISION ALGORITHM
THE EUCLIDEAN ALGORITHM

A LOOP TO COMPUTE A PRODUCT:
[pre-condition: m is a nonnegative integer,
x is a real number, $i = 0$, and product = 0.]

while (i # m)
1. product := product + x
2. $i := i + 1$

end while
[post-condition: product = $m \cdot x$]

PROOF:
Let the loop invariant be
$I(n)$: $i = n$ and product = $n \cdot x$
The guard condition G of the while loop is
G: $i \# m$

I.Basis Property:
[$I(0)$ is true before the first iteration of the loop.]
$I(0)$: $i = 0$ and product = $0 \cdot x = 0$
Which 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 before a loop iteration (where $k \geq 0$), then $I(k + 1)$ is true after the loop iteration.]
Before execution of statement 1,
$$product_{\text{old}} = k \cdot x.$$
Thus the execution of statement 1 has the following effect:
$$product_{\text{new}} = product_{\text{old}} + \text{x} = k \cdot x + \text{x} = (k + 1) \cdot x$$
Similarly, before statement 2 is executed,
$$i_{\text{old}} = k,$$
So after execution of statement 2,
$$i_{\text{new}} = i_{\text{old}} + 1 = k + 1.$$
Hence after the loop iteration, the statement $I(k+1)$ (i.e., $i = k + 1$ and product = $(k + 1) \cdot x$) is true. This is what we needed to show.

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.]

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 $0 \leq r < d$.]

PROOF:

Let the loop invariant be
$I(n)$: $r = a - n \cdot d$ and $n = q$.

The guard of the **while** loop is
G: $r \geq d$

I.Basis Property:

[$I(0)$ is true before the first iteration of the loop.]

$I(0)$: $r = a - 0 \cdot d = a$ and $0 = q$.

II.Inductive property:

[If the guard G and the loop invariant $I(k)$ are both true before a loop iteration (where $k \geq 0$), then $I(k + 1)$ is true after the loop iteration.]

$I(k)$: $r = a - k \cdot d \geq 0$ and $k = q$

$I(k + 1)$: $r = a - (k + 1) \cdot d \geq 0$ and $k + 1 = q$

$r_{\text{new}} = r - d$
$= a - k \cdot d - d$
$= a - (k + 1) \cdot d$

$q = q + 1$
$= k + 1$

also

$r_{\text{new}} = r - d$
$\geq d - d = 0$ (since $r \geq 0$)

Hence $I(k + 1)$ is true.

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.]

G is false and I(N) is true.

That is, $r \geq d$ and $r = a - N \cdot d \geq 0$ and $N = q$.

or $\quad r = a - q \cdot d$

or $\quad a = q \cdot d + r$

Also combining the two inequalities involving r we get

$$0 \leq r < d$$

<u>THE EUCLIDEAN ALGORITHM:</u>

The greatest common divisor (gcd) of two integers a and b is the largest integer that divides both a and b. For example, the gcd of 12 and 30 is 6.

The Euclidean algorithm takes integers A and B with $A > B \geq 0$ and compute their greatest common divisor.

<u>HAND CALCULATION OF gcd</u>:

Use the Euclidean algorithm to find gcd(330, 156)

<u>SOLUTION:</u>

$$\begin{array}{r|r} & 2 \\ \cline{2-2} 156 & 330 \\ & 312 \\ \cline{2-2} & 18 \end{array} \qquad \begin{array}{r|r} & 8 \\ \cline{2-2} 18 & 156 \\ & 144 \\ \cline{2-2} & 12 \end{array}$$

$$\begin{array}{r|r} & 1 \\ \cline{2-2} 12 & 18 \\ & 12 \\ \cline{2-2} & 6 \end{array} \qquad \begin{array}{r|r} & 2 \\ \cline{2-2} 6 & 12 \\ & 12 \\ \cline{2-2} & 0 \end{array}$$

Hence gcd(330, 156) = 6

<u>EXAMPLE:</u>

Use the Euclidean algorithm to find gcd(330, 156)

Solution:

1.Divide 330 by 156:

This gives $330 = 156 \cdot 2 + 18$

2.Divide 156 by 18:

This gives $156 = 18 \cdot 8 + 12$

3.Divide 18 by 12:

This gives $18 = 12 \cdot 1 + 6$

4.Divide 12 by 6:

This gives $12 = 6 \cdot 2 + 0$

Hence gcd(330, 156) = 6.

LEMMA:

If a and b are any integers with b# 0 and q and r are nonnegative integers such that

$$a = q \cdot d + r$$

then

$$\gcd(a, b) = \gcd(b, r)$$

[pre-condition: A and B are integers with
$A > B \geq 0$, $a = A$, b = B, $r = B$.]

while (b # 0)

1. $r := a \bmod b$
2. $a := b$
3. b := r

end while[post-condition: a = gcd(A, B)]

PROOF:

Let the **loop invariant** be

$I(n)$: $\gcd(a, b) = \gcd(A, B)$ and $0 \leq b < a$.

The guard of the **while** loop is

G: b # 0

I.Basis Property:

[$I(0)$ is true before the first iteration of the loop.]

$I(0)$: $\gcd(a, b) = \gcd(A, B)$ and $0 \leq b < a$.

According to the precondition,

a = A, b = B, r = B, and $0 \leq B < A$.

Hence $I(0)$ 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 before a loop iteration (where k $\geq$ 0), then $I(k + 1)$ is true after the loop iteration.]

Since $I(k)$ is true before execution of the loop we have,

$\gcd(a_{old}, b_{old}) = \gcd(A, B)$ and $0 \leq b_{old} < a_{old}$

After execution of statement 1,

$r_{new} = a_{old} \bmod b_{old}$Thus,

$a_{old} = b_{old} \cdot q + r_{new}$ for some integer q

with,

$$0 \le r_{new} < b_{old}.$$

But

$$\gcd(a_{old}, b_{old}) = \gcd(b_{old}, r_{old})$$

and we have,

$$\gcd(b_{old}, r_{new}) = \gcd(A, B)$$

When statements 2 and 3 are executed,

$$a_{new} = b_{old} \text{ and } b_{new} = r_{new}$$

It follows that

$$\gcd(a_{new}, b_{new}) = \gcd(A, B)$$

Also,

$$0 \le r_{new} < b_{old}$$

becomes

$$0 \le b_{new} < a_{new}$$

Hence $I(k + 1)$ is true.

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.]