0.

Determine the largest value in the middle 40 elements of array f[0..100) of int.

$$r = \langle \uparrow j \ : 30 \le j < 70 : f.j \rangle$$

Model the problem domain.

* (0) $C.n \quad = \quad \langle \uparrow j \ : 30 \le j < n : f.j \rangle \qquad , 30 \le n \le 70$

Consider

$C.30$

= {(0)}

$\langle \uparrow j \ : 30 \le j < 30 : f.j \rangle$

= { Empty Range }

$Id \uparrow$

Which gives us

- (1) $C.30 \quad = \quad Id \uparrow$

Consider

$C.(n+1)$

= {(0)}

$\langle \uparrow j \ : 30 \le j < n+1 : f.j \rangle$

= { Split j = n term }

$\langle \uparrow j \ : 30 \le j < n : f.j \rangle \ \uparrow f.n$

= {(0)}

$C.n \uparrow f.n$

Which gives us

- (2) $C.(n+1) \quad = \quad C.n \uparrow f.n \qquad ,30 \le n < 70$

Rewrite the postcondition using the model.

Post : $r = C.70$

Strengthen the postcondition.

Post : $r = C.n \ \wedge \ n = 70$

Choose invariants.

P0: $r = C.n$
P1: $30 \le n \le 70$

Guard.

$n \neq 70$

Establish invariants.

$n, r := 30, Id \uparrow$

Variant.

$70 - n$

Loop body.

$(n, r := n+1, E).P0$
$=$ { Text Substitution }
$E = C.(n+1)$
$=$ {(2)}
$E = C.n \uparrow f.n$
$=$ {P0}
$E = r \uparrow f.n$

Finished algorithm.

$n, r := 30, Id \uparrow$
;do $n \neq 70 \rightarrow$
$\quad n, r := n+1, r \uparrow f.n$
od

1.
Construct a program to compute the product of the natural numbers from 12 to 98.

$$r = \langle * j \ : 12 \le j < 99 : j \rangle$$

Model the problem domain.

$* \ (0) \ C.n \quad = \quad \langle * j \ : 12 \le j < n : j \rangle \qquad , 12 \le n \le 99$

Consider

$C.12$
$= \quad \{(0)\}$
$\langle * j \ : 12 \le j < 12 : f.j \rangle$
$= \quad \{ \text{Empty Range} \}$
$Id\ *$

Which gives us

$- \ (1) \ C.12 \quad = \quad Id\ *$

Consider

$C.(n+1)$
$= \quad \{(0)\}$
$\langle * j \ : 12 \le j < n+1 : j \rangle$
$= \quad \{ \text{Split } j = n \text{ term} \}$
$\langle * j \ : 12 \le j < n : j \rangle \ * n$
$= \quad \{(0)\}$
$C.n * n$

Which gives us

$- \ (2) \ C.(n+1) \quad = \quad C.n * n \qquad ,12 \le n < 99$

Rewrite the postcondition using the model.

Post : $r = C.99$

Strengthen the postcondition.

Post : $r = C.n \ \wedge \ n = 99$

Choose invariants.

P0: $r = C.n$
P1: $12 \le n \le 99$

Guard.

$n \ne 99$

Establish invariants.

$n, r := 12, Id *$

Variant.

$99 - n$

Loop body.

$(n, r := n+1, E).P0$
= { Text Substitution }
$E = C.(n+1)$
= {(2)}
$E = C.n * n$
= {P0}
$E = r * n$

Finished algorithm.

```
 n, r := 12, Id *
;do n ≠ 99 →
      n, r := n+1, r * n
 od
```

2.

Construct a program to count the number of single digit values in the array f[0..N) of int.

$$r = \langle + j : 0 \leq j < N : g.(f.j) \rangle$$

where

$$g.x \quad = \quad 1 \quad \Leftarrow \quad -10 < x \ \wedge \ x < 10$$

$$g.x \quad = \quad 0 \quad \Leftarrow \quad x \leq -10 \ \vee \ 10 \leq x$$

Strengthen postcondition.

$$r = \langle + j : 0 \leq j < n : g.(f.j) \rangle \quad \wedge n = N$$

Domain modelling.

* (0) $C.n \quad = \quad \langle + j : 0 \leq j < n : g.(f.j) \rangle \quad , 0 \leq n \leq N$

* (1) $g.x \quad = \quad 1 \quad \Leftarrow \quad -10 < x \ \wedge \ x < 10$

* (2) $g.x \quad = \quad Id+ \quad \Leftarrow \quad x \leq -10 \ \vee \ 10 \leq x$

Consider

$$\begin{array}{ll} & C.0 \\ = & \qquad \{(0)\} \\ & \langle + j : 0 \leq j < 0 : g.(f.j) \rangle \\ = & \qquad \{ \text{Empty Range} \} \\ & Id+ \end{array}$$

which gives us

\- (3) $C.0 \quad = \quad Id+$

Consider

$$\begin{array}{ll} & C.(n+1) \\ = & \qquad \{(0)\} \\ & \langle + j : 0 \leq j < n+1 : g.(f.j) \rangle \\ = & \qquad \{ \text{Split off } j{=}n \text{ term} \} \\ & \langle + j : 0 \leq j < n : g.(f.j) \rangle \ + g.(f.n) \\ = & \qquad \{ \text{Case } -10 < f.n \ \wedge \ f.n < 10 \ (1) \} \\ & \langle + j : 0 \leq j < n : g.(f.j) \rangle \ + 1) \\ = & \qquad \{(0)\} \\ & C.n + 1 \end{array}$$

Which gives us

- (4) $C.(n+1) \quad = \quad C.n + 1 \quad \Leftarrow \quad -10 < f.n \;\wedge\; f.n < 10 \quad , 0 \le n < N$

Also consider

$$
\begin{array}{ll}
 & C.(n+1) \\
= & \qquad \{(0)\} \\
 & \langle +j : 0 \le j < n+1 : g.(f.j) \rangle \\
= & \qquad \{ \text{Split off } j\text{=}n \text{ term} \} \\
 & \langle +j : 0 \le j < n : g.(f.j) \rangle \;\; + g.(f.n) \\
= & \qquad \{ \text{Case } f.n \le -10 \;\vee\; 10 \le f.n \text{ (2)} \} \\
 & \langle +j : 0 \le j < n : g.(f.j) \rangle \;\; + Id+ \\
= & \qquad \{(0)\} \\
 & C.n + Id+
\end{array}
$$

Which gives us

- (5) $C.(n+1) \quad = \quad C.n + Id+ \quad \Leftarrow \quad f.n \le -10 \;\vee\; 10 \le f.n \quad , 0 \le n < N$

This completes our model.

Rewrite postcondition using model.

$r = C.n \;\wedge\; n = N$

Choose invariants.

P0: $r = C.n$
P1: $0 \le n \le N$

Guard.

$n \ne N$

Establish invariants.

$n, r := 0, Id+$

Variant.

$N - n$

Loop body.

$(n, r := n{+}1, E).P0$

$=$ { Textual Substitution }

$E = C.(n{+}1)$

$=$ { Case $-10 < f.n \ \wedge\ f.n < 10$ (4) }

$E = C.n + 1$

$=$ { P0 }

$E = r + 1$

Considering the other case

$(n, r := n{+}1, E).P0$

$=$ { Textual Substitution }

$E = C.(n{+}1)$

$=$ { Case $f.n \leq -10 \ \vee\ 10 \leq f.n$ (5) }

$E = C.n + Id{+}$

$=$ { P0 }

$E = r + Id{+}$

Finished algorithm.

```
n, r := 0, Id+
; do n ≠ N  →
        if 10 < f.n  ∧  f.n < 10 →  n, r := n+1, r + 1
        [] f.n ≤ -10  ∨  10 ≤ f.n →  n, r := n+1, r + Id+
        fi
  od
```

3.
Construct a program to count the vowels in an array f[0..2000) of char.

$$r = \langle + j : 0 \le j < 2000 : g.(f.j) \rangle$$

where

g.x	=	1	$\Leftarrow$	vowel.x
g.x	=	0	$\Leftarrow$	$\neg$ vowel.x

Strengthen postcondition.

$$r = \langle + j : 0 \le j < n : g.(f.j) \rangle \quad \wedge n = 2000$$

Domain modelling.

* (0) C.n $=$ $\langle + j : 0 \le j < n : g.(f.j) \rangle$, $0 \le n \le 2000$

* (1) g.x $=$ 1 $\Leftarrow$ vowel.x

* (2) g.x $=$ Id+ $\Leftarrow$ $\neg$ vowel.x

Consider

C.0
= {(0)}
$\langle + j : 0 \le j < 0 : g.(f.j) \rangle$
= { Empty Range }
Id+

which gives us

\- (3) C.0 $=$ Id+

Consider

C.(n+1)
= {(0)}
$\langle + j : 0 \le j < n+1 : g.(f.j) \rangle$
= { Split off j=n term }
$\langle + j : 0 \le j < n : g.(f.j) \rangle + g.(f.n)$
= { Case vowel.(f.n) (1) }
$\langle + j : 0 \le j < n : g.(f.j) \rangle + 1)$
= {(0)}
C.n + 1

Which gives us

- (4) $C.(n+1) \quad = \quad C.n + 1 \quad \Leftarrow \quad vowel.(f.n) \quad , 0 \le n < 2000$

Also consider

$$
\begin{array}{ll}
 & C.(n+1) \\
= & \quad \{(0)\} \\
 & \langle +j : 0 \le j < n+1 : g.(f.j) \rangle \\
= & \quad \{ \text{Split off } j{=}n \text{ term} \} \\
 & \langle +j : 0 \le j < n : g.(f.j) \rangle \ + g.(f.n) \\
= & \quad \{ \text{Case } \neg\, vowel.(f.n)\ (2) \} \\
 & \langle +j : 0 \le j < n : g.(f.j) \rangle \ + Id{+} \\
= & \quad \{(0)\} \\
 & C.n + Id{+}
\end{array}
$$

Which gives us

- (5) $C.(n+1) \quad = \quad C.n + Id{+} \quad \Leftarrow \quad \neg\, vowel.(f.n) \quad , 0 \le n < 2000$

This completes our model.

Rewrite postcondition using model.

$r = C.n \ \wedge n = N$

Choose invariants.

P0: $r = C.n$
P1: $0 \le n \le 2000$

Guard.

$n \neq 2000$

Establish invariants.

$n, r := 0, Id{+}$

Variant.

$2000 - n$

Loop body.

$$
\begin{array}{ll}
& (n, r := n+1, E).P0 \\
= & \quad \{ \text{ Textual Substitution } \} \\
& E = C.(n+1) \\
= & \quad \{ \text{ Case vowel.(f.n) (4) } \} \\
& E = C.n + 1 \\
= & \quad \{ \text{ P0 } \} \\
& E = r + 1
\end{array}
$$

Considering the other case

$$
\begin{array}{ll}
& (n, r := n+1, E).P0 \\
= & \quad \{ \text{ Textual Substitution } \} \\
& E = C.(n+1) \\
= & \quad \{ \text{ Case } \neg \text{ vowel.(f.n) (5) } \} \\
& E = C.n + Id+ \\
= & \quad \{ \text{ P0 } \} \\
& E = r + Id+
\end{array}
$$

Finished algorithm.

```
n, r := 0, Id+
; do n ≠ 2000  →
      if vowel.(f.n) →  n, r := n+1, r + 1
      [] ¬ vowel.(f.n) →  n, r := n+1, r + Id+
      fi
  od
```