2. [0.6pt, 0.3 each] Loop invariants.

(a) The short answer is: \( i \leq j \land j + 2i = 24 \) will do.

A lengthy answer with some explanation is below.

A first candidate you might try is \( I = i \leq j \). This holds initially, and together with the negation of the loop’s guard it implies the post-condition to establish the latter when the loop terminates. Unfortunately, we cannot prove that this invariant is maintainable. That is, we cannot prove \( \{ I \land g \} \text{ loop } \{ I \} \). Or, if we rephrase this using \( \text{wp} \), we can’t prove \( I \land g \Rightarrow \text{wp body } I \).

To show you this, let’s calculate:

\[ \text{wp} (j := j + 2; i := i + 1) \ i \leq j \]

This yields: \( i + 1 \leq j - 2 \), which does not follow from \( I \land g = (i \leq j \land i < j) \).

However, the predicate actually holds before and after iteration. It is just that we cannot prove it, which implies that our invariant does not contain enough information. In particular, there is a tighter relation between \( i \) and \( j \) which is not well described with just \( i \leq j \).

I propose to strengthen it to: \( I = i \leq j \land j + 2i = 24 \).

If we again calculate:

\[ \text{wp} (j := j + 2; i := i + 1) \ i \leq j \land j + 2i = 24 \]

This yields: \( i + 1 \leq j - 2 \land j - 2 + 2(i + 1) = 24 \). The second conjunct can be simplified to \( j + 2i = 24 \), which is clearly implied by the invariant.

For the first conjunct, let’s first simplify it:

\[
\begin{align*}
  i + 1 & \leq j - 2 \\
  i & \leq j - 3 \\
  i & \leq 24 - 2i - 3 \\
  3i & \leq 21 \\
  i & \leq 7 \\
  (*) & i < 8
\end{align*}
\]

Now, we can also rewrite \( g \) as follows:

\[
\begin{align*}
  i & < j \\
  i & < 24 - 2i \\
  3i & < 24 \\
  i & < 8 \quad \text{which obviously implies (*) above.}
\end{align*}
\]

(b) LN 6.14 no 16: a program to multiply two non-negative integers. We can use the following invariant:

\[ (z + xy = XY) \land 0 \leq x \]

where \( X \) and \( Y \) represent the initial values of \( x \) and \( y \).
3. [0.4] Termination

Invariant:

\[(k=1 \lor \text{even } k) \land 0 \leq k \leq 10\]

Note that simply \(0 \leq k \leq 10\) is not enough. Although it is good enough to establish the post-condition when the loop terminates, it is not strong enough to be maintained invariant on its own. If we calculate the \(wp\) over the loop body, and then further simplify the resulting expression:

\[
wp \ (\text{if } k=1 \ \text{then } k := 0 \ \text{else } k := k+2) \ 0 \leq k \leq 10 \\
= \ // \ \text{calculating } wp \\
(k=1 \Rightarrow 0 \leq 0 \leq 10) \land (k\neq1 \Rightarrow 0 \leq k+2 \leq 10) \\
= \ // \ 0 \leq 0 \leq 10 \ \text{is a tautology} \\
(k\neq1 \Rightarrow 0 \leq k+2 \leq 10)
\]

In particular, \(k+2 \leq 10\) can’t be proven from \(0 \leq k \leq 10 \land k\neq10\).

Termination metric: \(k=1 \rightarrow 12 \mid 10-k\). To help you convincing yourself, below is table showing how the termination metric evolves during the iterations; this is of course not a formal proof :)

<table>
<thead>
<tr>
<th>iteration</th>
<th>value of (k)</th>
<th>value of the termination metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>