Questão 1 (Semântica denotational do laço while)
A denotação do laço while é

$$C[\![\texttt{while b do c}]\!] = \operatorname{fix} F \qquad \text{com } F(\tau) = \operatorname{cond}(B[\![b]\!], \tau \circ C[\![c]\!], id).$$

A semântica do laço é definido como menor ponto fixo de um funcional F. F tem como argumento uma função entre estados e retorna uma outra função entre estados. Se τ é uma aproximação à semântica do laço while, $F(\tau)$ resulta numa outra aproximação, com mais uma iteração do laço executada. Como a semântica do laço deve ser independente dessa operação, ela é um ponto fixo desse funcional. Sem tem mais que um ponto, escolhemos o menor ponto fixo. A definição $\operatorname{fix} F$ corresponde com essa solução.

Questão 2 (Semântica axiomática: Média)
(a) $x \leq y + 1 \land x + y = x_0 + y_0$

(b)

$\{\ x \leq y \land x = x_0 \land y = y_0\ \}$
$\{\ x \leq y + 1 \land x + y = x_0 + y_0\ \}$
while x<y **do** (
  $\{\ x < y \land x \leq y + 1 \land x + y = x_0 + y_0\ \}$
  $\{\ x + 1 \leq y \land x + y = x_0 + y_0\ \}$
  x:=x+1
  $\{\ x \leq y \land x + y - 1 = x_0 + y_0\ \}$
  y:=y−1
  $\{\ x \leq y + 1 \land x + y = x_0 + y_0\ \}$
)
$\{\ x \geq y \land x \leq y + 1 \land x + y = x_0 + y_0\ \}$
$\{\ (x = y \lor x = y + 1) \land x + y = x_0 + y_0\ \}$

(c) $y - x + 1$

(d)

$\{x \leq y \land x = x_0 \land y = y_0\}$
$\{x \leq y + 1 \land x + y = x_0 + y_0 \land \mathbf{y - x + 1 \geq 0}\}$
while x<y **do** (
  $\{x < y \land x \leq y + 1 \land x + y = x_0 + y_0 \land \mathbf{0 \leq v_0 = y - x + 1}\}$
  $\{x + 1 \leq y \land x + y = x_0 + y_0 \land \mathbf{0 \leq y - x - 1 < v_0 = y - x + 1}\}$
  x:=x+1
  $\{x \leq y \land x + y - 1 = x_0 + y_0 \land \mathbf{0 \leq y - x < v_0}\}$
  y:=y−1
  $\{x \leq y + 1 \land x + y = x_0 + y_0 \land \mathbf{0 \leq y - x + 1 < v_0}\}$
)
$\{x \geq y \land x \leq y + 1 \land x + y = x_0 + y_0\}$
$\{(x = y \lor x = y + 1) \land x + y = x_0 + y_0\}$

Questão 3 (Avaliação da semântica denotational)
A equação semântica do laço é $C[\![\texttt{while n>0 do n:=2*n-3}]\!] = \operatorname{fix} F$ com $F(f) = \operatorname{cond}(B[\![n > 0]\!], f \circ C[\![\texttt{n:=2*n-3}]\!], \mathrm{id})$ e com $C[\![\texttt{n:=2*n-3}]\!] = (\lambda\sigma \in \Sigma_\perp.\sigma[n \mapsto 2\sigma(n) - 3])$ obtemos

$$F(f)\sigma = \begin{cases} f(\sigma[n \mapsto 2\sigma(n) - 3]) & \text{se } \sigma(n) > 0 \\ \sigma & \text{se } \sigma(n) \leq 0. \end{cases}$$

(a) Com as equações acima, temos

$$F^0(\bot) = \bot$$

$$F^1(\bot)\sigma = \begin{cases} \bot & \text{se } \sigma(n) > 0 \\ \sigma & \text{se } \sigma(n) \leq 0 \end{cases}$$

$$F^2(\bot)\sigma = \begin{cases} \bot & \text{se } \sigma(n) \geq 2 \\ \sigma[n \mapsto -1] & \text{se } \sigma(n) = 1 \\ \sigma & \text{se } \sigma(n) \leq 0 \end{cases}$$

$$F^3(\bot)\sigma = \begin{cases} \bot & \text{se } \sigma(n) \geq 3 \\ \sigma[n \mapsto -1] & \text{se } 1 \leq \sigma(n) \leq 2 \\ \sigma & \text{se } \sigma(n) \leq 0 \end{cases}$$

(b) Na próxima iteração obtemos

$$F^4(\bot)\sigma = \begin{cases} \bot & \text{se } \sigma(n) \geq 3 \\ \sigma[n \mapsto -1] & \text{se } 1 \leq \sigma(n) \leq 2 \\ \sigma & \text{se } \sigma(n) \leq 0 \end{cases}$$

e logo achamos o menor ponto fixo do F que é o limite superior mínimo da cadeia $F^i(\bot)$.

Questão 4 (Execução inversa)

(a) $C[\![\texttt{reverse } \mathbf{c_1}\ \mathbf{c_2}]\!] = C[\![\mathbf{c_1}]\!] \circ C[\![\mathbf{c_2}]\!]$

(b) Prova

$$C[\![\texttt{reverse } \mathbf{c_1}\ \texttt{(reverse } \mathbf{c_2}\ \mathbf{c_3}\texttt{)}]\!] = C[\![\mathbf{c_1}]\!] \circ C[\![\texttt{reverse } \mathbf{c_2}\ \mathbf{c_3}]\!] = C[\![\mathbf{c_1}]\!] \circ C[\![\mathbf{c_2}]\!] \circ C[\![\mathbf{c_3}]\!]$$

$$C[\![\mathbf{c_3}\texttt{;(}\mathbf{c_2}\texttt{;}\mathbf{c_1}\texttt{)}]\!] = C[\![\mathbf{c_2}\texttt{;}\mathbf{c_1}]\!] \circ C[\![\mathbf{c_3}]\!] = C[\![\mathbf{c_1}]\!] \circ C[\![\mathbf{c_2}]\!] \circ C[\![\mathbf{c_3}]\!]$$

Questão 5 (Exponenciação descendente)

A exponenciação descendente é definido por $n^{\underline{k}} = n(n-1)\cdots(n-k+1) = n!/(n-k)!$ para $n \geq k \geq 0$.

(a) Implementação

```
e:=1
n0:=n
while n0 ≠ n−k do
  e:=e∗n0
  n0:=n0−1
```

(b) Prova com invariante $e = n!/n_0!$

```
{ n ≥ k ≥ 0 }
{ 1 = 1 }
e:=1
{ e = 1 }
{ e = n!/n! }
n0:=n
{ e = n!/n0! }
while n0 ≠ n−k do (
  { e = n!/n0! }
  { en0 = n!/(n0 − 1)! }
  e:=e∗n0
  { e = n!/(n0 − 1)! }
  n0:=n0−1
  { e = n!/n0! }
)
{ n0 = n − k ∧ e = n!/n0! }
{ e = n!/(n − k)! }
{ e = n^k }
```