Kevin James

1.

```
method Abs(x : int) returns (y : int)
  ensures 0 <= y;
  ensures 0 <= x ==> y == x;
  ensures x < 0 ==> y == -x;
{
  if (x < 0)c
   {y := -x;}S1
  else
   {y := x;}S2
}
```

The post-state, R, is $\{(0 \le x \Rightarrow y == x) \,\&\&\, (x<0 \Rightarrow y == -x) \,\&\&\, (0 \le y)\}$ (conjunction of ensures clauses)

Using the rule for weakest precondition of an if-else:

$wp(\text{if } c \text{ then } s_1 \text{ else } s_2, R) = [c \wedge wp(s_1, R)] \vee [\neg c \wedge wp(s_2, R)]$

True Branch:

$[c \wedge wp(s_1, R)] = x<0 \wedge wp(y := -x, 0 \le x \Rightarrow y == x \,\&\&\, x<0 \Rightarrow y == -x \,\&\&\, 0 \le y)$ $y \leftarrow -x$

$= x<0 \wedge ((0 \le x \Rightarrow -x == x) \,\&\&\, (x<0 \Rightarrow -x == -x) \,\&\&\, (0 \le -x))$ by assignment rule

False Branch:

$[\neg c \wedge wp(s_2, R)] = x \ge 0 \wedge wp(y := x, 0 \le x \Rightarrow y == x \,\&\&\, x<0 \Rightarrow y == -x \,\&\&\, 0 \le y)$ $y \leftarrow x$

$= x \ge 0 \wedge (0 \le x \Rightarrow x == x \,\&\&\, x<0 \Rightarrow x == -x \,\&\&\, 0 \le x)$ by assignment rule

So now our overall weakest precondition is

$x<0 \wedge ((\cancel{0 \le x} \Rightarrow -x == x) \,\&\&\, (x<0 \Rightarrow -x == -x) \,\&\&\, (0 \le -x)) \vee$

$x \ge 0 \wedge ((0 \le x \Rightarrow x == x) \,\&\&\, (\cancel{x<0} \Rightarrow x == -x) \,\&\&\, (0 \le x))$

$x=0$ $x=0$

which from a quick inspection reduces to True.

Since no requires clause is declared, the pre-state is True. So, we must show that True $\Rightarrow$ True is a tautology. This is so simply by definition of an implication.

2.

```
method Q2(x : int, y : int) returns (big : int, small : int)
  ensures big > small; R
{
  if (x > y) c
   {big, small := x, y;} s1
  else
   {big, small := y, x;} s2
}
```

$$wp(\text{if } c \text{ then } s_1 \text{ else } s_2, R) = [c \wedge wp(s_1, R)] \vee [\neg c \wedge wp(s_2, R)]$$

True Branch:

$$[c \wedge wp(s_1,R)] = (x>y) \wedge wp(s_1, big > small) \quad \text{big} \leftarrow x,\ \text{small} \leftarrow y$$

$$= x>y \wedge x>y \quad \text{logical simplification (redundancy)}$$

$$= x>y$$

False branch

$$[\neg c \wedge wp(s_2,R)] = (x \le y) \wedge wp(s_2, big > small) \quad \text{small} \leftarrow x,\ \text{big} \leftarrow y$$

$$= (x \le y) \wedge (y > x) \quad \text{logical simplification (both sides are true precisely when } x<y)$$

$$= x<y$$

Putting these together,

$$wP(Q2, R) = (x>y) \vee (x<y)$$

A. So we need a precondition Q, for which $Q \Rightarrow (x>y) \vee (y>x)$ is a tautology.

In its current form there is no requires clause so the precondition is simply True. This doesn't satisfy the tautology; consider the counterexample $x == y$.

Consider imposing the precondition "requires $x\ != y$"

B. To prove this is a sufficient precondition, we can show that

$$x \neq y \Rightarrow ((x>y) \vee (y>x)) \text{ is a tautology.}$$

When the left side is true, the right side is always true, so this program holds.