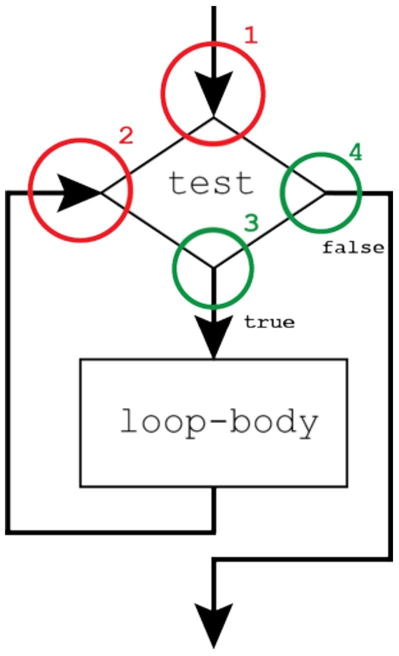
```
void flip(StackOfT& s)
//! updates s
//! ensures s = rev(#s)
```

S	Code	Assume		Confirm
0		true		true
	StackOfT t;			
1		t1 = <>	Unchanged s	rev(t1) * s1 = rev(t1) * s1 ①
	while(s.length() > 0) { //! updates s, t //! maintains //! rev(t) * s = //! rev(#t) * #s //! decreases |s|			
2		|s2| > 0 ^ rev(t2) * s2 = rev(t1) * s1 ③	Unchanged s, t	
	T y;			
3		T.Init(y3)	Unchanged s, t	s3 /= <>
	s.pop(y);			
4		s4 = s3[1,|s3|) ^ <y4> = prefix of s3	Unchanged t	
	t.push(y);			
5		T.Init(y5) ^ t5 = <y4> * t4	Unchanged s	|s5| < |s2| ^ rev(t5) * s5 = rev(t1) * s1 ②
	}			
6		~(|s6| > 0) ^ rev(t6) * s6 = rev(t1) * s1 ④		true
	s.transferFrom(t)			

7		s7 = t6 ^
		t7 = <>
				s7 = rev(s0)

Some of the Stack operations

```
template <class T>
class Stack1
    //! is modeled by string of T
    //! exemplar self
{
public: // Standard Operations
    Stack1 ();
        //! replaces self
        //! ensures: self = <>
    void transferFrom (Stack1& source);
        //! replaces self
        //! clears source
        //! ensures: self = #source
    void push (T& x);
        //! updates self
        //! clears x
        //! ensures: self = <#x> * #self
    void pop (T& x);
        //! updates self
        //! replaces x
        //! requires: self /= <>
        //! ensures: <x> is prefix of #self  and  self = #self[1, |#self|)
    Integer length (void);
        //! restores self
        //! ensures: length = |self|
}
```


1
2
4
test
3
false
true
loop-body