Sample Pelican Execution

Execution	Environment	Store
program show **is var** fib	[fib $\mapsto$ var(0)]	[0 $\mapsto$ undef]
procedure f(n) **is**	[fib $\mapsto$ var(0), f $\mapsto$ proc1(proc)]	same as above
begin f(2)	[fib $\mapsto$ var(0), f $\mapsto$ proc1(proc), n $\mapsto$ var(1)]	[0 $\mapsto$ undef, 1 $\mapsto$ int(2)]
var t:int	[fib $\mapsto$ var(0), f $\mapsto$ proc1(proc), n $\mapsto$ var(1), t $\mapsto$ var(2)]	[0 $\mapsto$ undef, 1 $\mapsto$ int(2), 2 $\mapsto$ undef]
if n≤1 **else** f(n-1);	[fib $\mapsto$ var(0), f $\mapsto$ proc1(proc), n $\mapsto$ var(3), t $\mapsto$ var(2)]	[0 $\mapsto$ undef, 1 $\mapsto$ int(2), 2 $\mapsto$ undef, 3 $\mapsto$ int(1)]
var t:int	fib $\mapsto$ var(0), f $\mapsto$ proc1(proc), n $\mapsto$ var(3), t $\mapsto$ var(4)]	[0 $\mapsto$ undef, 1 $\mapsto$ int(2), 2 $\mapsto$ undef, 3 $\mapsto$ int(1), 4 $\mapsto$ undef]
if n≤1 **then** fib:=n	same as above	[0 $\mapsto$ int(1), 1 $\mapsto$ int(2), 2 $\mapsto$ undef, 3 $\mapsto$ int(1), 4 $\mapsto$ undef]
end {f}	[fib $\mapsto$ var(0), f $\mapsto$ proc1(proc), n $\mapsto$ var(1), t $\mapsto$ var(2)]	same as above
t:=fib; f(n-2)	[fib $\mapsto$ var(0), f $\mapsto$ proc1(proc), n $\mapsto$ var(5), t $\mapsto$ var(2)]	[0 $\mapsto$ int(1), 1 $\mapsto$ int(2), 2 $\mapsto$ int(1), 3 $\mapsto$ int(1), 4 $\mapsto$ undef, 5 $\mapsto$ int(0)]
var t:int	[fib $\mapsto$ var(0), f $\mapsto$ proc1(proc), n $\mapsto$ var(5), t $\mapsto$ var(6)]	[0 $\mapsto$ int(1), 1 $\mapsto$ int(2), 2 $\mapsto$ int(1), 3 $\mapsto$ int(1), 4 $\mapsto$ undef, 5 $\mapsto$ int(0), 6 $\mapsto$ undef]
if n≤1 **then** fib:=n	same as above	[0 $\mapsto$ int(0), 1 $\mapsto$ int(2), 2 $\mapsto$ int(1), 3 $\mapsto$ int(1), 4 $\mapsto$ undef, 5 $\mapsto$ int(0), 6 $\mapsto$ undef]
end {f}	[fib $\mapsto$ var(0), f $\mapsto$ proc1(proc), n $\mapsto$ var(1), t $\mapsto$ var(2)]	same as above
fib:= t+fib	same as above	[0 $\mapsto$ int(1), 1 $\mapsto$ int(2), 2 $\mapsto$ int(1), 3 $\mapsto$ int(1), 4 $\mapsto$ undef, 5 $\mapsto$ int(0), 6 $\mapsto$ undef]
end {f}		

where proc =
λ loc **.** (perform ["definition of proc f"]
[fib $\mapsto$ var(0), f $\mapsto$ proc1(proc), n $\mapsto$ var(loc)])

```
program showLocal is
var fib: integer;
```

Sample Pelican Execution

```
  procedure f(n: integer) is
  var t: integer
  begin {at return fib = Fibonacci(n)}
  if n≤1
     then fib:= n
     else f(n-1); t:= fib;
          f(n-2); fib:= t+fib
  end if
  end
begin f(2)
end
```