```
{ n > 0 }
i := 1;
{ n > 0 ∧ i = 1 }
ii := 3;
{ n > 0 ∧ i = 1 ∧ ii = 3 }
ki := 3;
{ n > 0 ∧ i = 1 ∧ ii = 3 ∧ ki = 3 }
ski := 0;
{ n > 0 ∧ i = 1 ∧ ii = 3 ∧ ki = 3 ∧ ski = 0 }
si := 0;
{ n > 0 ∧ i = 1 ∧ ii = 3 ∧ ki = 3 ∧ ski = 0 ∧ si = 0 }
while {  (i - 1) * (i - 1) * (i - 1) <= n ∧ si = 3 * i * (i - 1) / 2 ∧ ii = 6 * i - 3 ∧ ki =
3 * i * i ∧ ski = (i - 1) * i * (2 * i - 1) / 2 } ski + si + i <= n  do
   { i * i * i <= n ∧ si = 3 * i * (i - 1) / 2 ∧ ii = 6 * i - 3 ∧ ki = 3 * i * i ∧ ski = (i
- 1) * i * (2 * i - 1) / 2 }
   ski := ski + ki;
   { i * i * i <= n ∧ si = 3 * i * (i - 1) / 2 ∧ ii = 6 * i - 3 ∧ ki = 3 * i * i ∧ ski = i *
(i + 1) * (2 * i + 1) / 2 }
   ii := ii + 6;
   { i * i * i <= n ∧ si = 3 * i * (i - 1) / 2 ∧ ii = 6 * i + 3 ∧ ki = 3 * i * i ∧ ski = i *
(i + 1) * (2 * i + 1) / 2 }
   ki := ki + ii;
   { i * i * i <= n ∧ si = 3 * i * (i - 1) / 2 ∧ ii = 6 * i + 3 ∧ ki = 3 * (i + 1) * (i + 1)
∧ ski = i * (i + 1) * (2 * i + 1) / 2 }
   si := si + 3 * i;
   { i * i * i <= n ∧ si = 3 * (i + 1) * i / 2 ∧ ii = 6 * i + 3 ∧ ki = 3 * (i + 1) * (i + 1)
∧ ski = i * (i + 1) * (2 * i + 1) / 2 }
   i := i + 1;
   { (i - 1) * (i - 1) * (i - 1) <= n ∧ n > 0 ∧ si = 3 * i * (i - 1) / 2 ∧ ii = 6 * i - 3 ∧ ki
= 3 * i * i ∧ ski = (i - 1) * i * (2 * i - 1) / 2 }
done
{ (i - 1) * (i - 1) * (i - 1) <= n ∧ n < i * i * i }
```