

Semantyka i weryfikacja programów 2022/23.
Zadanie 3.

Dany jest następujący program obliczający logarytm dyskretny przy podstawie **b** z liczby *n* zapisany w języku TINY (rozszerzonym na potrzeby formułowania asercji o operację potęgowania określoną dla liczb całkowitych dodatnich w zwykły sposób).

```

{n>0 ∧ b>1}
  p:=0;
  bp:=1;
{bp = bp ∧ bp ≤ n ∧ n>0 ∧ b>1 }
  while {γ1: bp = bp ∧ bp ≤ n ∧ n>0 ∧ b>1 }
    bp*b ≤ n do
  (
{bp = bp ∧ bp ≤ n ∧ bp*b ≤ n ∧ b = b1 ∧ 1 ≥ 1 ∧ n>0 ∧ b>1 }
  p1:=1;
  bp1:=b;
{bp = bp ∧ bp ≤ n ∧ bp*bp1 ≤ n ∧ bp1 = bp1 ∧ n>0 ∧ b>1 }
  while {γ2: bp = bp ∧ bp ≤ n ∧ bp*bp1 ≤ n ∧ bp1 = bp1 ∧ n>0 ∧ b>1 }
    bp*bp1*bp1 ≤ n do
  (
{bp = bp ∧ bp ≤ n ∧ bp*bp1*bp1 ≤ n ∧ bp1 = b(p1+p1) ∧ n>0 ∧ b>1}
  bp1:=bp1*bp1;
{bp = bp ∧ bp ≤ n ∧ bp*bp1 ≤ n ∧ bp1 = b(p1+p1) ∧ n>0 ∧ b>1}
  p1:=p1+p1;
{bp = bp ∧ bp ≤ n ∧ bp*bp1 ≤ n ∧ bp1 = bp1 ∧ n>0 ∧ b>1}
  );
{bp = bp ∧ bp ≤ n ∧ bp*bp1 ≤ n ∧ bp1 = bp1 ∧ n>0 ∧ b>1 ∧ bp*bp1*bp1>n}
{bp * bp1 = b(p+p1) ∧ b(p+p1) ≤ n ∧ n>0 ∧ b>1}
  bp:=bp*bp1;
{bp = b(p+p1) ∧ b(p+p1) ≤ n ∧ n>0 ∧ b>1}
  p:=p+p1;
{bp = bp ∧ bp ≤ n ∧ n>0 ∧ b>1}
  );
{bp = bp ∧ bp ≤ n ∧ bp*b>n ∧ n>0 ∧ b>1 }
{bp ≤ n < b(p+1)}

```

Udowodnij częściową poprawność programu względem podanej specyfikacji tj.

1. Podaj niezmienniki obu pętli γ_1 i γ_2 .
2. Wstaw odpowiednie formuły w nawiasy $\{, \}$ tak, aby powstałe asercje umożliwiały przeprowadzenie dowodu częściowej poprawności. Jeśli w dwóch sąsiednich wierszach występują nawiasy $\{, \}$, to pomiędzy wstawionymi tam asercjami powinna zachodzić implikacja. Można też dostawić dodatkowe nawiasy.