# tutaj umieszczam komentarz #  
// mogę to zrobić również tutaj  
  
{ # kod rozpoczynam od utworzenia głównego bloku #  
 public int a; # deklaruję publiczną zmienną typu int #  
 private matrix b; # deklaruję publiczną zmienną typu matrix #  
  
 function max(int c, int d) : int # deklaruję prosta funkcję z dwoma argumentami i typem zwracanym UWAGA: nie można deklarować a oraz b (te nazwy są publiczne) #  
 {  
 private int e; # tworzę zmienna #  
 if( c > d )  
 {  
 e := c; # przypisuje zmienna #  
 }  
 else  
 {  
 e := d;  
 }  
 return e; # wykonuje return ktory jest mozliwy tylko w glownym bloku funkcji #  
 }  
  
 function odwroc(matrix up) : matrix  
 {  
 private matrix down;  
 down := [up[1,1],up[2,1] : up[1,2],up[2,2]];  
 return down;  
 }  
  
 function wyznacznik(matrix m) : int # deklaruję funkcję wyznacznik #  
 {  
 private int w; # tworzę zmienna #  
  
 w := m[1,1]\*m[2,2] - m[1,2]\*m[2,1]; # dostep poprzez indeks #  
  
 return w; # wykonuje return ktory jest mozliwy tylko w glownym bloku funkcji #  
 }  
  
 function main() : int # deklaruję funkcję main od której rozpocznie się wykonanie programu #  
 {  
 private int c; # deklaruję zmienne prywatne widoczną tylko w tym bloku (w bloku main)#  
 private int d;  
 private int e;  
  
 c := 5; # przypisuję wartości do zmiennych #  
 d := 10;  
  
 e := max(c,d); # wywołuję funkcję #  
  
 @ e ; # wypisuje wartosc #  
  
 # wykonamy proste operacje matematyczne #  
  
 @ e\*e+2\*7-14;  
  
 # teraz wykonamy pewne dzialania na macierzach #  
  
 private matrix B; # niedozwolonej jest jednolinijkowe przypisanie #  
 private matrix A;  
 A := [1,2,3 : 4,5,6 : 3,2,1]; # wiersze macierzy rozdzielamy dwukropkiem #  
 B := [ 1,2,3,4,5,6,7,8,9 : 9,8,7,6,5,4,3,2,1 : 0,1,0,2,0,3,0,4,0 ];  
  
 @ A\*B; # ogladamy wynik #  
  
 # teraz przetestujemy instrukcje while #  
 a := 0; # krzystam ze zmiennej globalnej #  
 while(a < 3)  
 {  
 A := A\*A;  
 a := a + 1;  
 }  
  
 @ A; # ogladamy wynik #  
  
 # policzymy jeszcze wyznacznik macierzy 2 x 2 #  
 b := [5,6 : 1,2]; # znow korzystam ze zmiennej publicznej #  
 private int r;  
 r := wyznacznik(b);  
  
 @ r; # ogladamy wynik #  
  
 # probujemy jeszcze odwrocic macierz #  
 private matrix val;  
 val := odwroc(b);  
  
 @ val;  
  
 }  
  
} # kończę główny blok programu #