

Pseudokod

- abstrakcyjna notacja algorytmów
- wygląda podobnie do popularnych języków (Java, C/C++, Pascal, Python)
- rola bardziej informacyjna niż formalna (możliwie rozluźniony formalizm, o ile nie prowadzi do niejednoznaczności)
 - literały (liczby, znaki, łańcuchy, NULL)
 - zmienne (bez konieczności uprzedniej deklaracji)
 - tablice i operator indeksowania "[]" (zakładamy indeksowanie tablic od 0)
 - operatory (przypisanie '=', relacyjne "==, <, >", logiczne "&, ||, !", matematyczne "+, ++, +=", etc.
 - funkcje (w tym rekursja), instrukcja return (zakładamy przesyłanie argumentu dla typów złożonych przez referencję)
 - konstrukcje sterujące: wyrażenia warunkowe (IF, THEN, ELSE), pętle (WHILE, FOR).

Algorytm – specyfikacja

- nazwa i argumenty
- warunek początkowy (wejście)
- warunek końcowy (wyjście)

Całkowicie poprawny algorytm

- własności stopu
 - ↳ algorytm zatrzymuje się po skończonej liczbie kroków
- częściowa poprawność
 - ↳ algorytm przy zatrzymaniu zwraca poprawny wynik

Definition

Niezmiennik pętli to logiczny predykat spełniający następujący warunek:

jeśli predykat jest spełniony **przed** wejściem w pewną (dowolną) iterację pętli **to** jest także spełniony **po wyjściu z tej iteracji** pętli.

Znalezienie niezmiennika

- zapisać predykat wyrażający warunek końcowy
- przekształcić warunek końcowy aby:
 - zawierał wszystkie istotne zmienne
 - wyraził bieżącą wart. zmiennej zwracanej przez algorytm
 - spełniał definicję niezmiennika pętli
- sprawdzić czy proponowany niezmiennik jest też spełniony przed pierwszą iteracją

Zadanie:

- pokazać, że zmienna x reprezentuje maksimum w tablicy Arr

Warunek końcowy:

- x jest nie mniejsze, niż dowolna liczba w tablicy Arr i x jest zawarte w Arr

```
algor1(Arr, len){  
  i = 1  
  x = Arr[0]  
  while(i < len)  
    if(Arr[i] > x){  
      x = Arr[i]  
    }  
    i++  
  return x  
}
```

$$(\forall_{0 \leq j \leq n} x \geq Arr[j]) \wedge (\exists_{0 \leq j \leq n} (x == Arr[j]))$$