

# Algorytmy

## Dane skalarne - Liczby

Informatyka

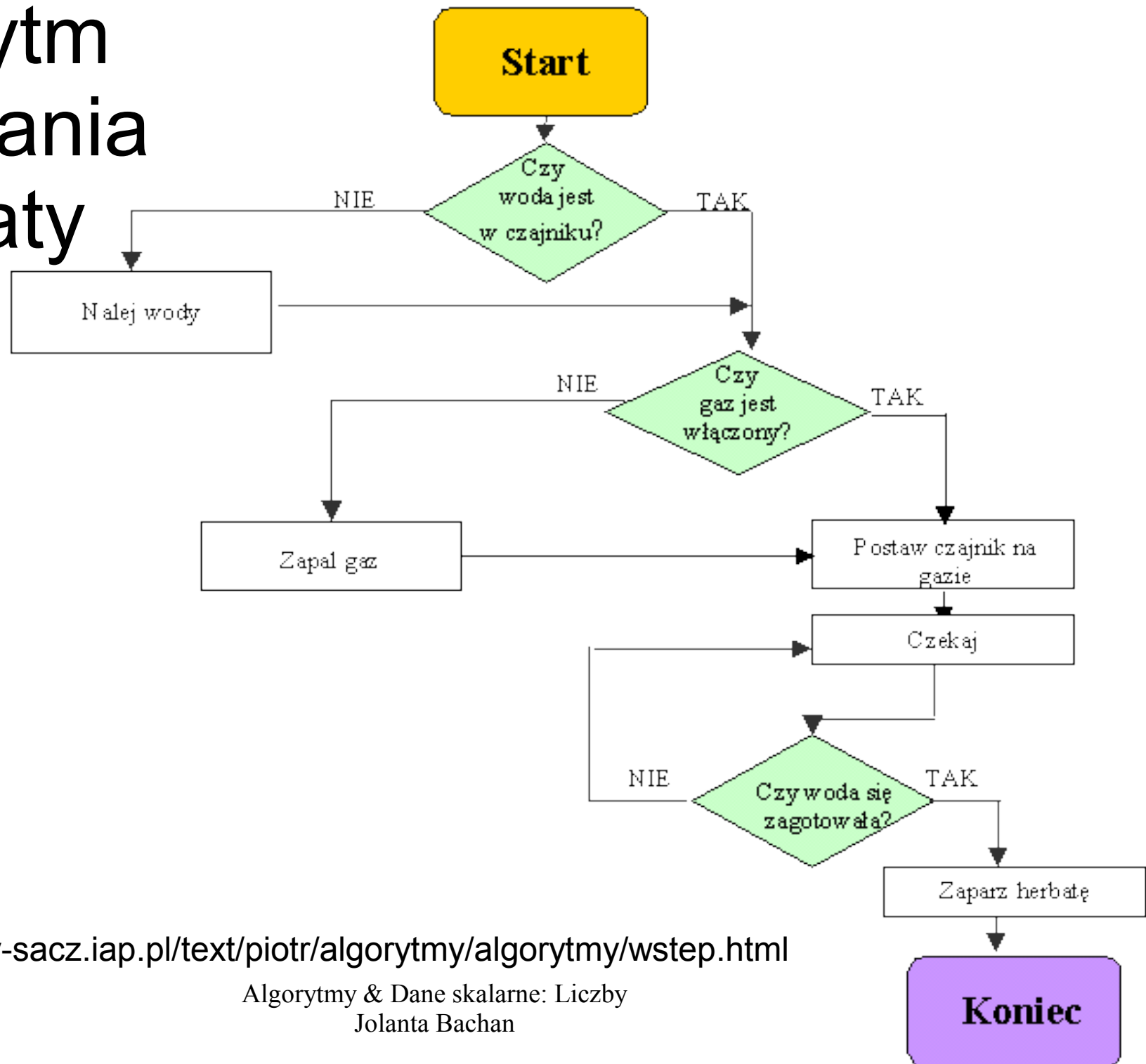
Jolanta Bachan  
2008-04-10

# Algorytmy

# Algorytmy

- Algorytm – w matematyce oraz informatyce to skończony, uporządkowany ciąg jasno zdefiniowanych czynności, koniecznych do wykonania pewnego zadania. (Wikipedia)

# Algorytm zaparzania herbaty

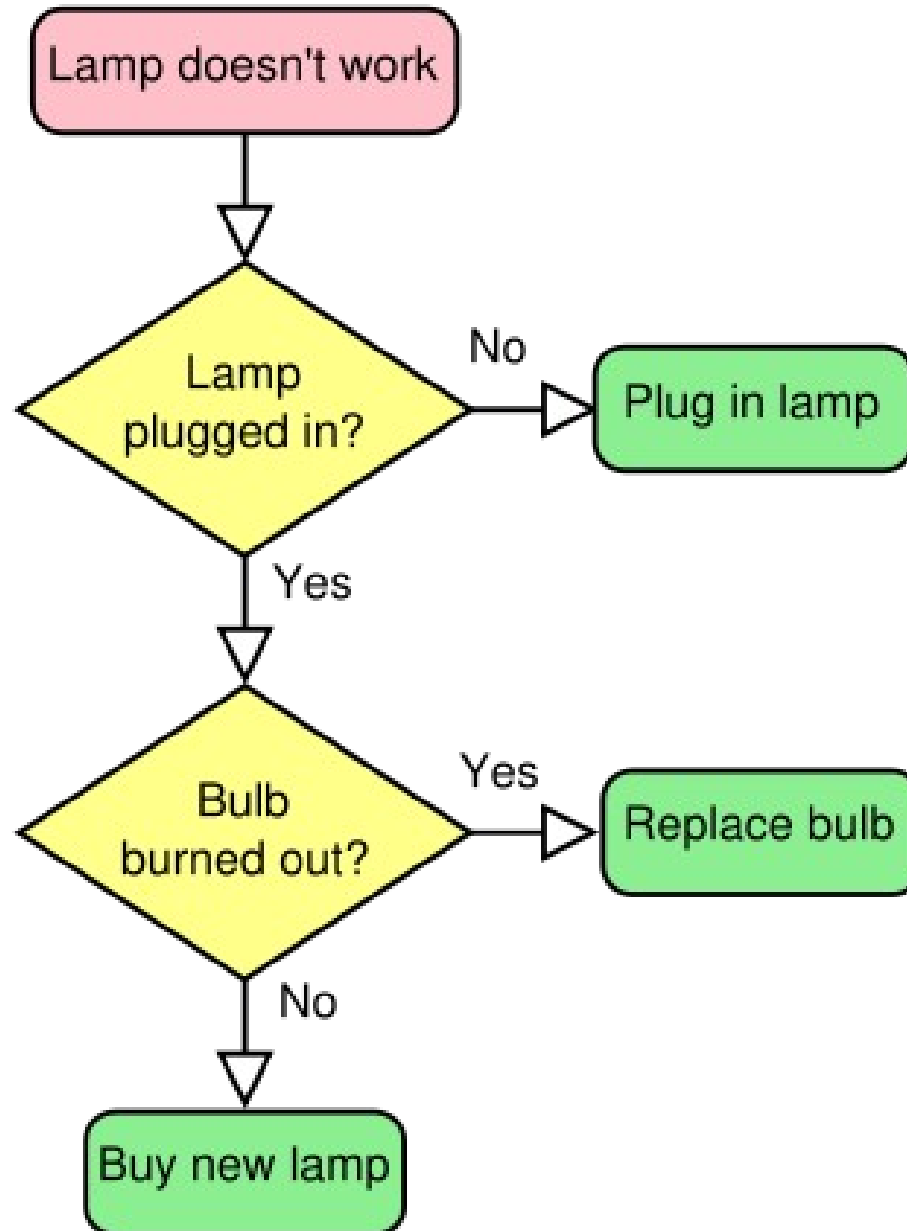


<http://www.lo.stary-sacz.iap.pl/text/piotr/algorytmy/algorytmy/wstep.html>

2008-04-10

Algorytmy & Dane skalarne: Liczby  
Jolanta Bachan

# Żarówka się nie pali. Co zrobić?



Wikipedia

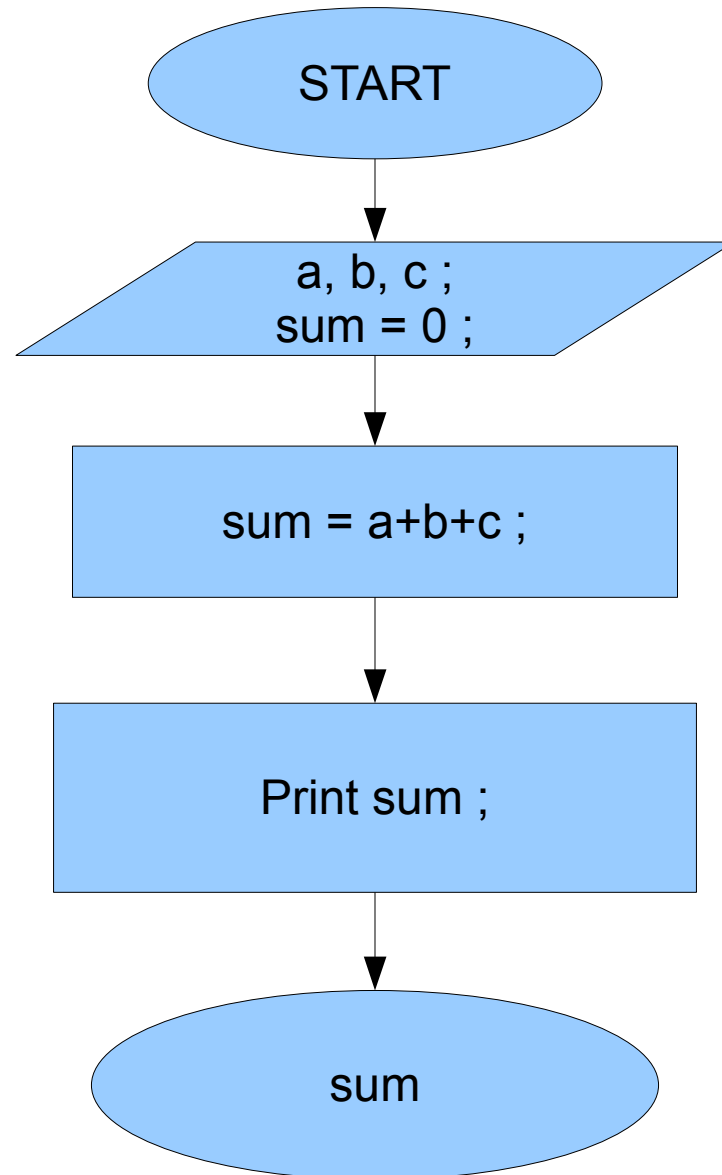
# Ćwiczenie

- Napisz algorytm opisujący jakąś czynność. Masz 10 minut.

# Suma $a, b, c$



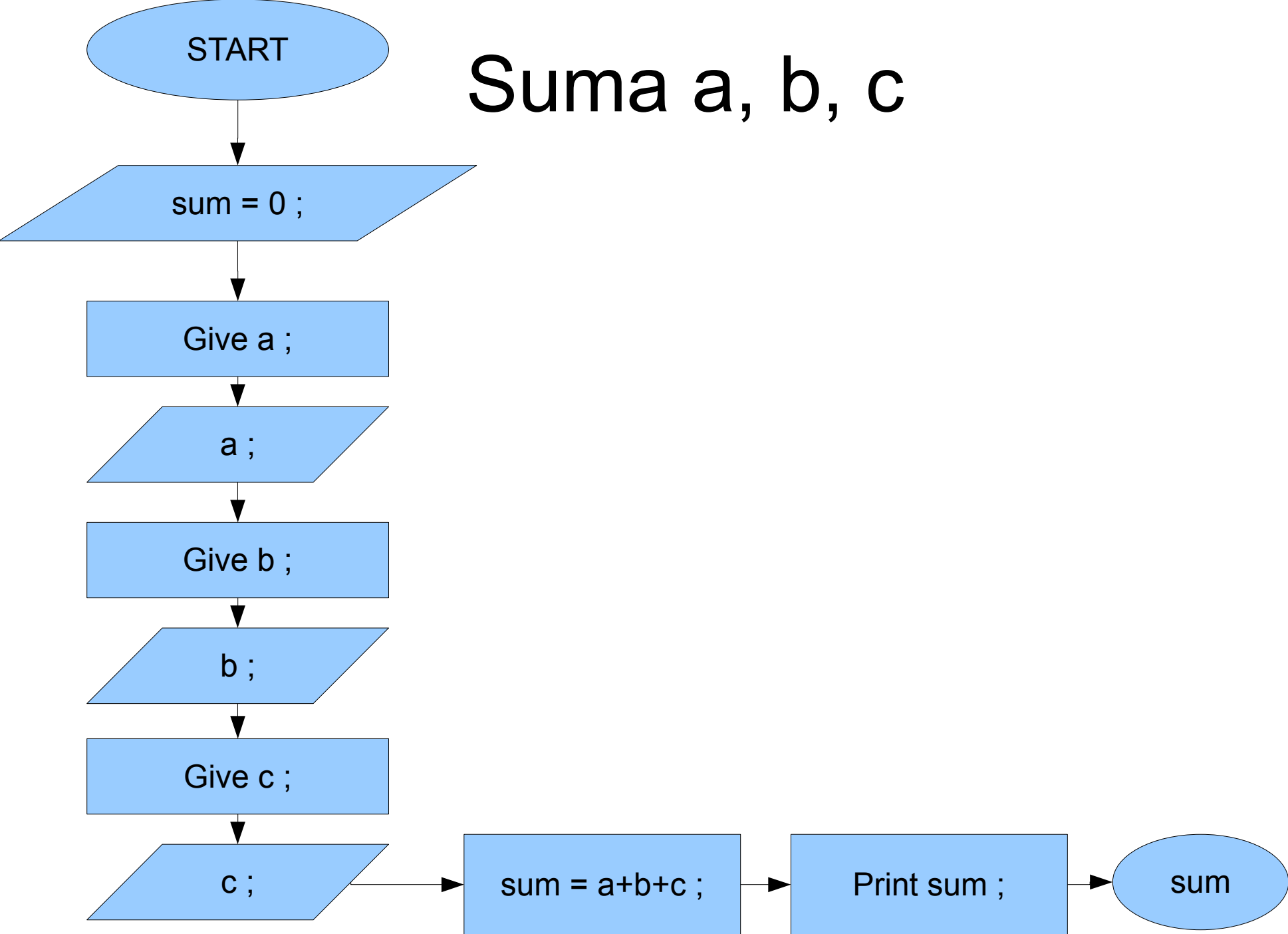
# Suma a, b, c



Suma  $a, b, c$

Lub ?

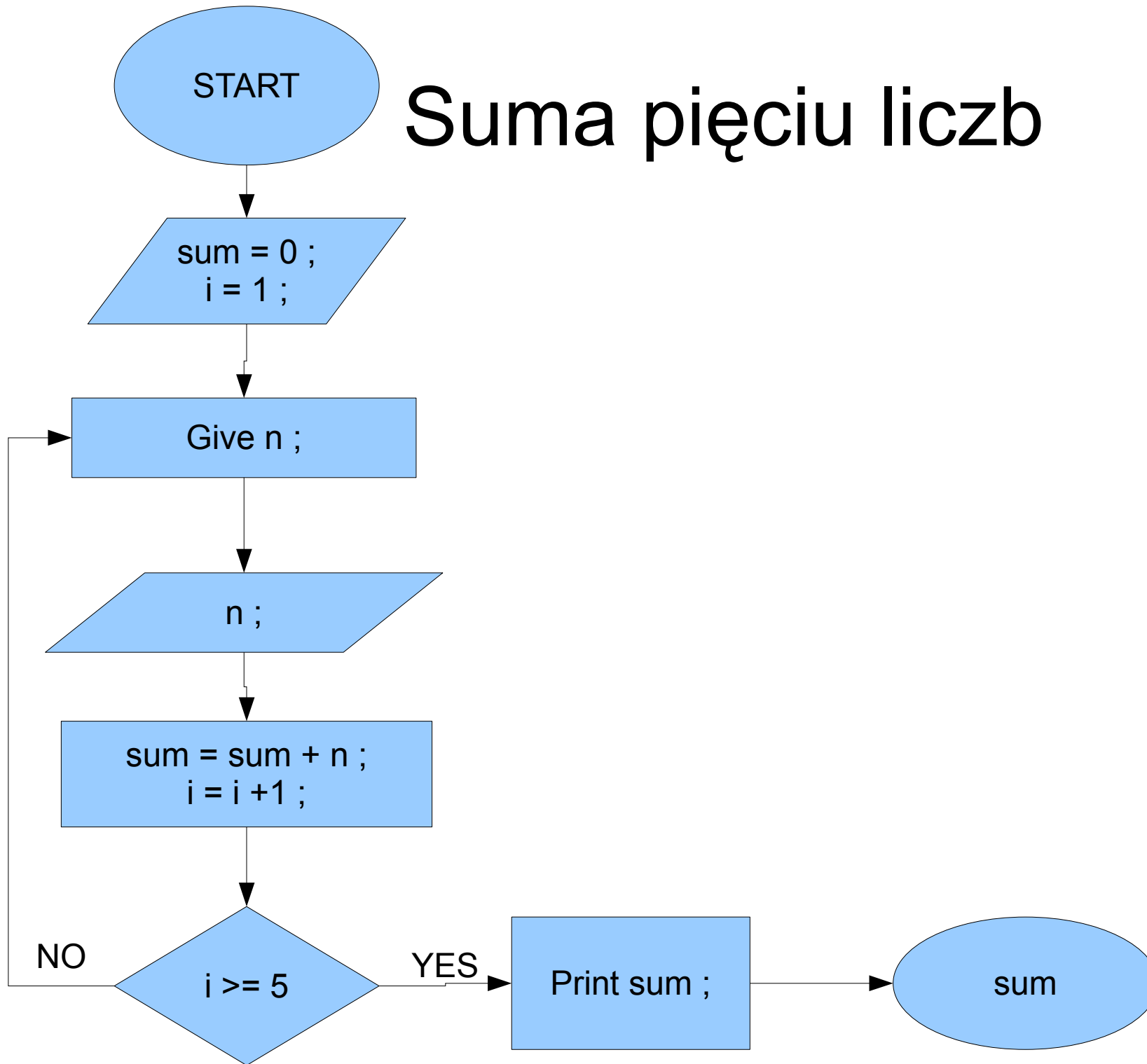
# Suma a, b, c



# Suma pięciu liczb



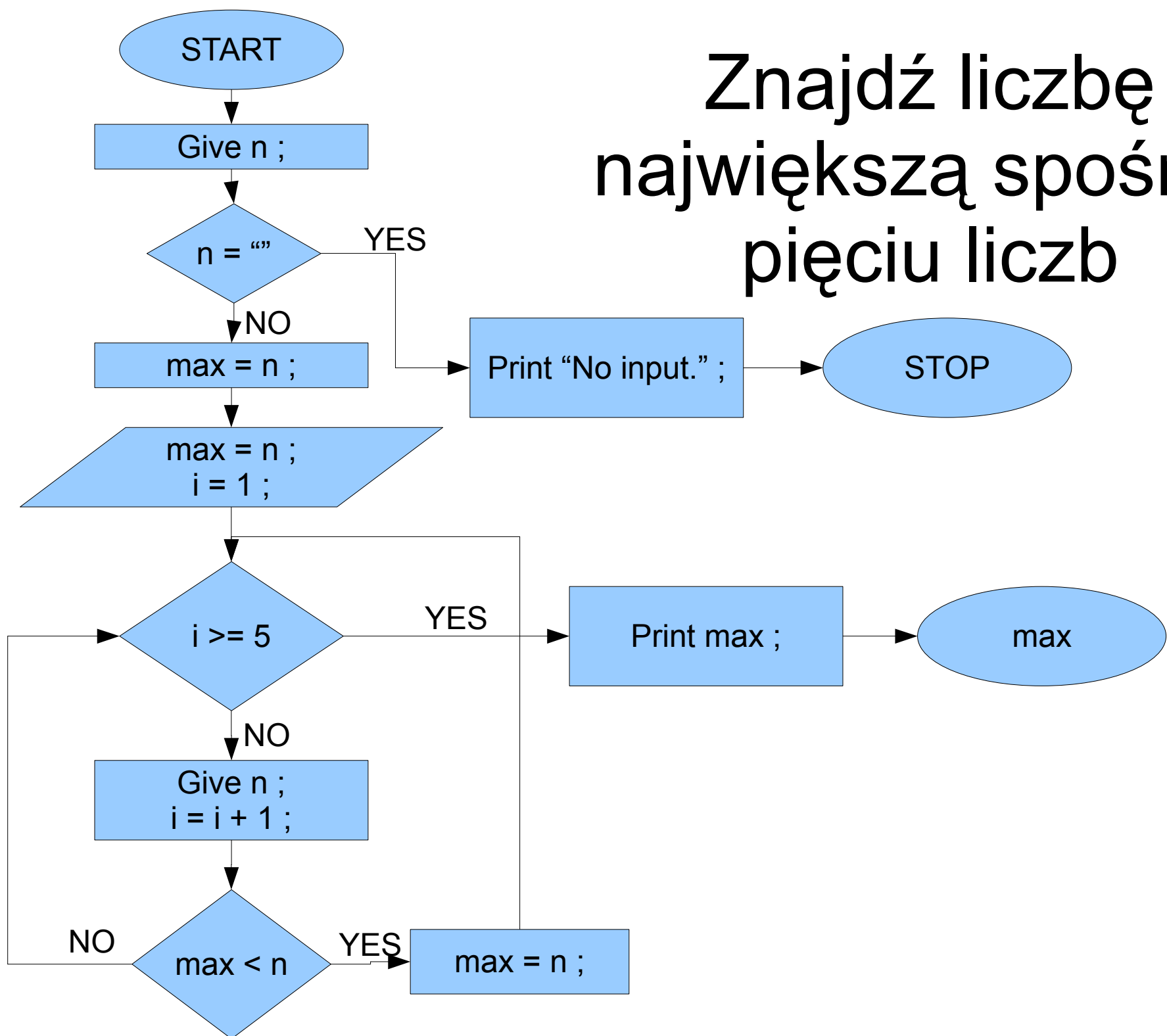
# Suma pięciu liczb



Znajdź liczbę  
największą spośród  
pięciu liczb



# Znajdź liczbę największą spośród pięciu liczb



# Dane skalarne

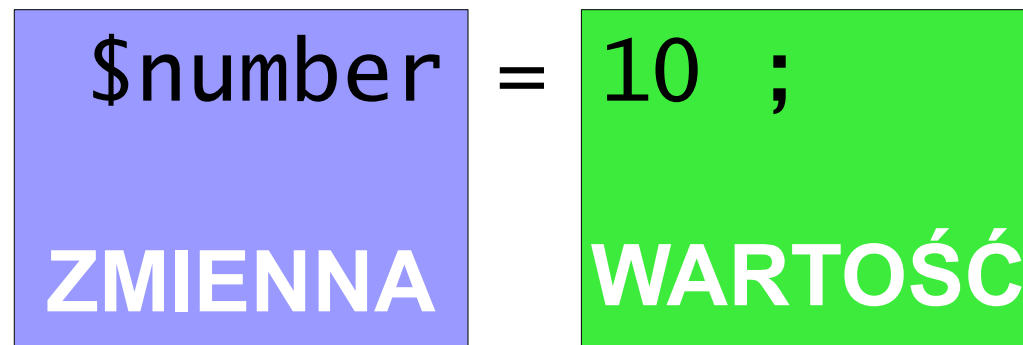
# Dane skalarne

- Dana skalarne to najprostsz y rodzaj danych. Mo że ni ą by ć liczba (jak 2008 lub 3.14) albo ci ąg znak ów (jak "Hello!")
- Zmienne skalarne zawsze zaczynaj ą si ę od znaku "\$".

```
$number = 10 ;
```

# Dane skalarne

- Dana skalarne to najprostszty rodzaj danych. Może nią być liczba (jak 2008 lub 3.14) albo ciąg znaków (jak "Hello!")
- Zmienne skalarne zawsze zaczynają się od znaku "\$".



# Nazywanie zmiennych

- Nazwa może być taka długa jak chcesz
- Nazwa może być kombinacją liter, podkreślników; może też zawierać liczby
- Po “\$” nie może następować liczba
- Wybieraj znaczące nazwy, np.  
`sum_of_two_numbers` ;

```
$x = 1 ;  
$y = 2 ;  
$z = $x + $y ;
```

**ŹLE!!!**

```
$x = 1 ;  
$y = 2 ;  
$sum = $x + $y ;
```

**lepiej**

```
$first_number = 1 ;  
$second_number = 2 ;  
$sum = $first_number  
+ $second_number ;
```

**IDEALNIE!!!**

# Literały

- **Literał** (ang. *literal*) określa sposób, w jaki wartość reprezentowana jest w kodzie źródłowym programu.
  - literały zmiennoprzecinkowe (ang. *float literals*)
  - literały całkowitoliczbowe (ang. *integer literals*)

# Literały zmiennoprzecinkowe

- 1.25
- 7.25e45 oznacza 7.25 razy 10 do potęgi 45
- -6.5e24 oznacza -6,5 razy 10 do potęgi 24 (duża liczba ujemna)
- -12e-24 oznacza -12 razy 10 do potęgi -24 (bardzo mała liczba ujemna)
- -1.2E-23 to inny sposób zapisu liczby powyżej – litera E może być wielka

# Literały całkowitoliczbowe

- 12
- 15
- -2007
- 189099

# Operatory skalarne

- Operatory dla liczb
  - +, -, \*, /, \*\*, sqrt
- Numeryczne operatory porównania
  - <, >, <=, >=, ==, !=

35 != 30 + 5 to Fałsz

35 == 35.0 to Prawda

# Przykłady

- $2 + 3$  oznacza 2 plus 3 lub inaczej 5
- $5.1 - 2.4$  oznacza 5,1 minus 2,4 lub inaczej 2,7
- $3 * 12$  oznacza 3 razy 12 równa się 36
- $14 / 2$  oznacza 14 dzielone przez 2 lub inaczej 7
- $10.2 / 0.3$  oznacza 10,2 dzielone przez 0,3 lub inaczej 34
- $10 / 3$  - Zawsze dzielenie zmiennoprzecinkowe, więc daje w rezultacie 3,3333333...

# Przykłady

`$a = 17;`

`$b = $a + 3 ; # 20`

`$b = $b * 2 ; # 40`

`$b = $b / 10 ; # 4`

`$b = $b**2 ; # 16`

`$b = sqrt($b) ; # 4`

# Dwuargumentowe operatory przypisania

- `$fred = 5 ;` “=” to operator przypisania
- `$fred = $fred + 5;` to zapis bez dwuargumentowego operatora przypisania
- `$fred += 5;` to zapis z dwuargumentowym operatorem przypisania
- `$barney = $barney * 3;`
- `$barney *= 3;`

# Dwuargumentowy operatory przypisania

- ++ -- (autoinkrementacja,  
autodekrementacja)

```
$a++ ;  
$b-- ;
```

# Operatory logiczne

- `&&` (logiczne i)
- `||` (logiczne lub)

# Wejście standardowe

- `<STDIN>` - *standard input as scalar data*

# Wyjście Standardowe

- `print` function – *standard output*

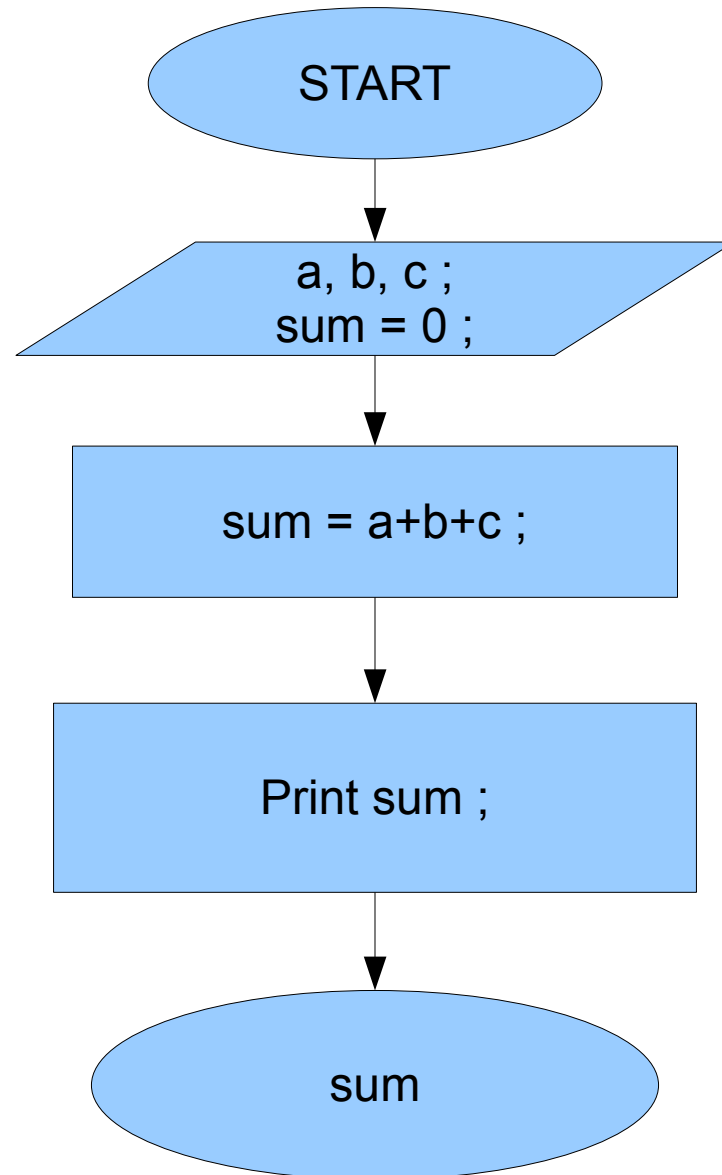
# Ćwiczenia

- Zaimplementuj w Perlu algorytmy obliczające sumę oraz algorytm znajdujący liczbę największą

# Suma $a, b, c$



# Suma a, b, c



# Suma a, b, c

```
$a = 1 ;
```

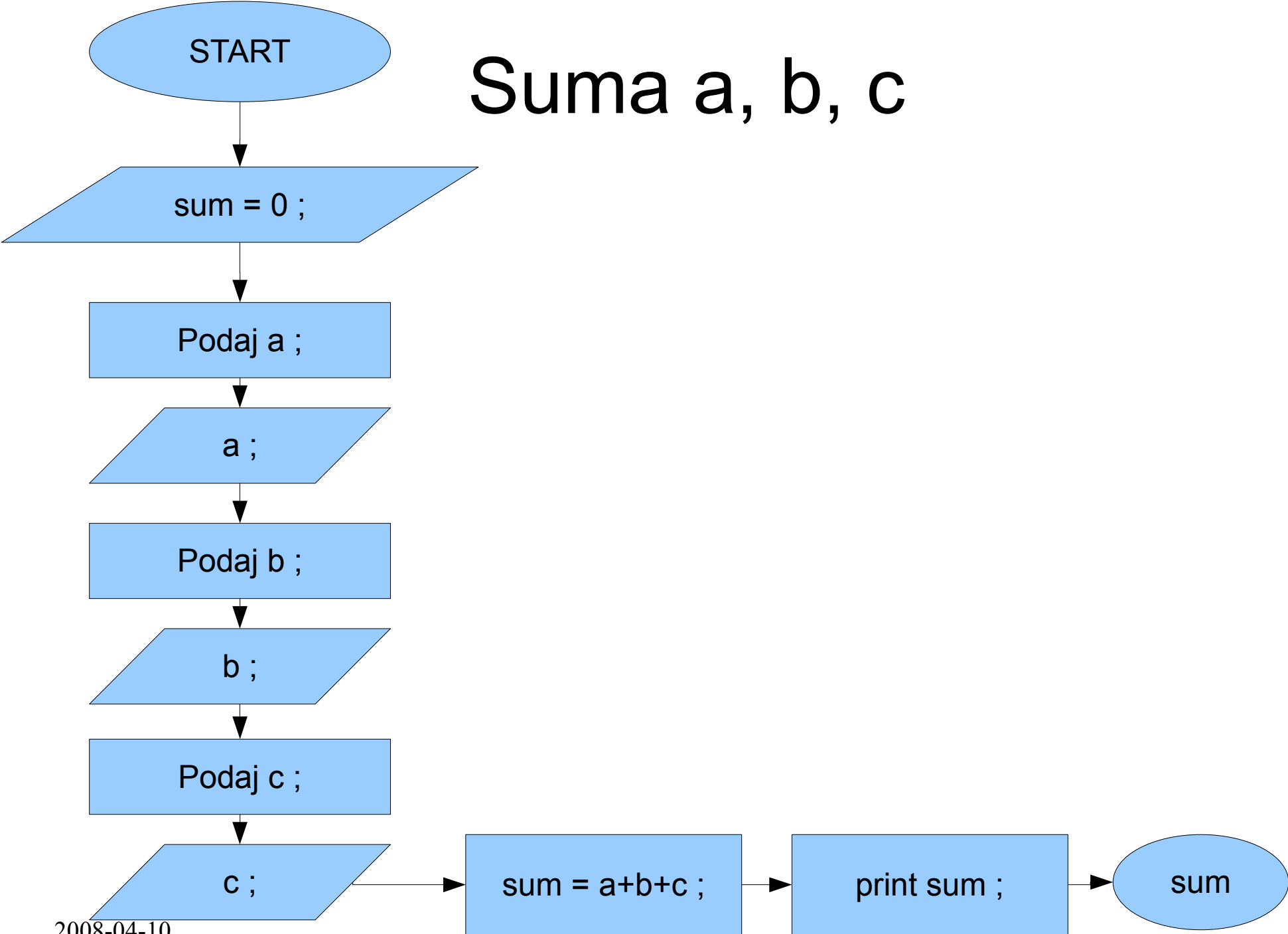
```
$b = 5 ;
```

```
$c = 4 ;
```

```
$sum = $a + $b + $c ;
```

```
print "The sum of a, b and c is: $sum.";
```

# Suma a, b, c



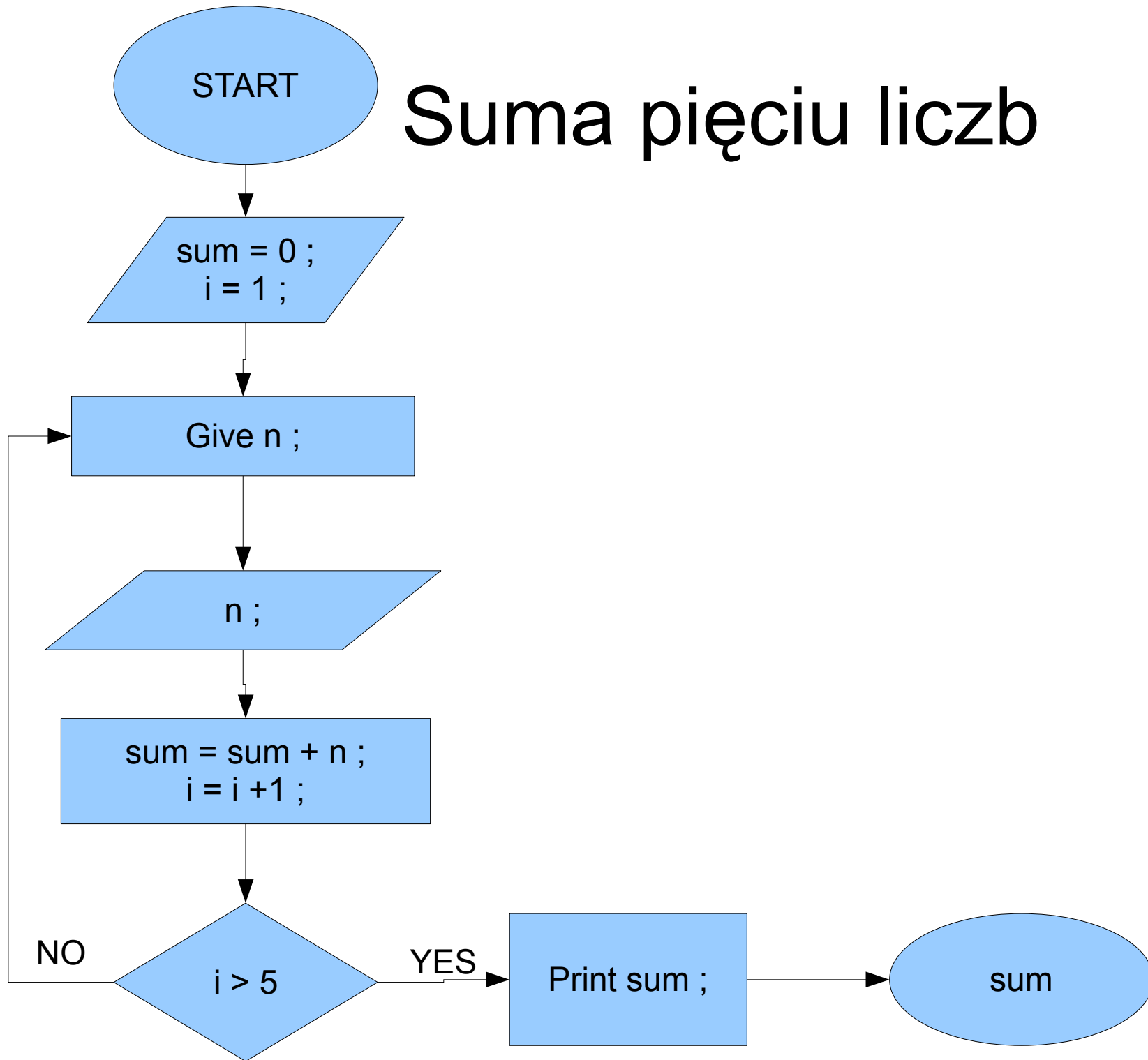
# Suma a, b, c

```
print "Give a: " ;  
$a = <STDIN> ;  
print "Give b: " ;  
$b = <STDIN> ;  
print "Give c: " ;  
$c = <STDIN> ;  
$sum = $a + $b + $c ;  
print "The sum of a, b and c is $sum." ;
```

# Suma pięciu liczb



# Suma pięciu liczb



# Suma pięciu liczb

```
$i = 0 ;  
$sum = 0 ;  
while ($i < 5) {  
    print "Give a number: " ;  
    $n = <STDIN> ;  
    $sum = $sum + $n ;  
    $i++ ;  
}  
print "The sum is $sum." ;
```

# Zadanie domowe

- Napisz algorytm znajdujący liczbę najmniejszą spośród 10 liczb
- Zainstaluj ActivePerl
  - <http://www.activestate.com/Products/activeperl/>
- Przetestuj ActivePerl
  - napisz program w Perlu
  - uruchom program