



## PROGRAMOWANIE NISKOPOZIOMOWE

### LABORATORIUM 3

## INSTRUKCJE SKOKÓW WARUNKOWYCH ORAZ ORGANIZACJA PĘTLI

dr inż. Bartosz Kowalczyk

*Częstochowa, 11 marca 2023*

## Spis treści

1	Skoki warunkowe dla znaczników flagowych	3
2	Skoki warunkowe dla liczb bez znaku	4
3	Skoki warunkowe dla liczb ze znakiem	5
4	Implementacja instrukcji wyboru	6
5	Implementacja pętli	7

# 1 Skoki warunkowe dla znaczników flagowych

Korzystając z instrukcji skoków warunkowych bazujących na wartościach flag oblicz wartość podanych wyrażeń:

1.  $y = \min(a, b)$
2.  $y = \max(a, b)$
3.  $y = \min(a, b, c)$
4.  $y = \max(a, b, c)$
5.  $y = \min(a, b, c, d)$
6.  $y = \max(a, b, c, d)$
7.  $y = |a|$
8.  $y = |15a + b|$
9.  $y = \begin{cases} 20 & \text{dla } a \geq b \\ -10 & \text{dla } a < b \end{cases}$
10.  $y = \begin{cases} a + b & \text{dla } a \geq b \\ a - b & \text{dla } a < b \end{cases}$
11. Sprawdzić, czy liczba  $a$  jest parzysta.
12. Sprawdzić, czy liczba  $a$  jest nieparzysta.
13. Sprawdzić, czy liczba  $a$  jest podzielna bez reszty przez 5.
14. Sprawdzić, czy liczba  $a$  jest podzielna bez reszty przez 16.
15. Sprawdzić, czy liczba  $a$  jest liczbą ze znakiem.
16. Sprawdzić, czy liczba  $a$  jest liczbą bez znaku.

Implementację powyższych funkcji należy rozważyć w następujących scenariuszach:

1. **Platforma x86.** Argumenty oraz wartość zwracana są typu 32-bit całkowitego bez znaku (`unsigned int`).
2. **Platforma x64.** Argumenty oraz wartość zwracana są typu 64-bit całkowitego bez znaku (`unsigned __int64`).
3. **Platforma x64.** Argumenty są typu 32-bit całkowitego bez znaku (`unsigned int`). Wartość zwracana jest typu 64-bit całkowitego bez znaku (`unsigned __int64`).

## 2 Skoki warunkowe dla liczb bez znaku

Korzystając z instrukcji skoków warunkowych dla liczb bez znaku oblicz wartość podanych wyrażeń:

1.  $y = \min(a, b)$
2.  $y = \max(a, b)$
3.  $y = \min(a, b, c)$
4.  $y = \max(a, b, c)$
5.  $y = \min(a, b, c, d)$
6.  $y = \max(a, b, c, d)$
7.  $y = |a|$
8.  $y = |15a + b|$
9.  $y = \begin{cases} 20 & \text{dla } a \geq b \\ -10 & \text{dla } a < b \end{cases}$
10.  $y = \begin{cases} a + b & \text{dla } a \geq b \\ a - b & \text{dla } a < b \end{cases}$
11. Sprawdzić, czy liczba  $a$  jest parzysta.
12. Sprawdzić, czy liczba  $a$  jest nieparzysta.
13. Sprawdzić, czy liczba  $a$  jest podzielna bez reszty przez 5.
14. Sprawdzić, czy liczba  $a$  jest podzielna bez reszty przez 16.
15. Sprawdzić, czy liczba  $a$  jest liczbą ze znakiem.
16. Sprawdzić, czy liczba  $a$  jest liczbą bez znaku.

Implementację powyższych funkcji należy rozważyć w następujących scenariuszach:

1. **Platforma x86.** Argumenty oraz wartość zwracana są typu 32-bit całkowitego bez znaku (`unsigned int`).
2. **Platforma x64.** Argumenty oraz wartość zwracana są typu 64-bit całkowitego bez znaku (`unsigned __int64`).
3. **Platforma x64.** Argumenty są typu 32-bit całkowitego bez znaku (`unsigned int`). Wartość zwracana jest typu 64-bit całkowitego bez znaku (`unsigned __int64`).

### 3 Skoki warunkowe dla liczb ze znakiem

Korzystając z instrukcji skoków warunkowych dla liczb bez znaku oblicz wartość podanych wyrażeń:

1.  $y = \min(a, b)$
2.  $y = \max(a, b)$
3.  $y = \min(a, b, c)$
4.  $y = \max(a, b, c)$
5.  $y = \min(a, b, c, d)$
6.  $y = \max(a, b, c, d)$
7.  $y = |a|$
8.  $y = |15a + b|$
9.  $y = \begin{cases} 20 & \text{dla } a \geq b \\ -10 & \text{dla } a < b \end{cases}$
10.  $y = \begin{cases} a + b & \text{dla } a \geq b \\ a - b & \text{dla } a < b \end{cases}$
11. Sprawdzić, czy liczba  $a$  jest parzysta.
12. Sprawdzić, czy liczba  $a$  jest nieparzysta.
13. Sprawdzić, czy liczba  $a$  jest podzielna bez reszty przez 5.
14. Sprawdzić, czy liczba  $a$  jest podzielna bez reszty przez 16.
15. Sprawdzić, czy liczba  $a$  jest liczbą ze znakiem.
16. Sprawdzić, czy liczba  $a$  jest liczbą bez znaku.

Implementację powyższych funkcji należy rozważyć w następujących scenariuszach:

1. **Platforma x86.** Argumenty oraz wartość zwracana są typu 32-bit całkowitego ze znakiem (`int`).
2. **Platforma x64.** Argumenty oraz wartość zwracana są typu 64-bit całkowitego ze znakiem (`__int64`).
3. **Platforma x64.** Argumenty są typu 32-bit całkowitego ze znakiem (`int`). Wartość zwracana jest typu 64-bit całkowitego ze znakiem (`__int64`).

## 4 Implementacja instrukcji wyboru

Korzystając z instrukcji skoków warunkowych oblicz wartość podanych wyrażeń:

$$1. y = \begin{cases} 10 & \text{dla } i = 0 \\ 20 & \text{dla } i = 1 \\ 30 & \text{domyślnie} \end{cases}$$

$$2. y = \begin{cases} a + b & \text{dla } i = 0 \\ a - b & \text{dla } i = 1 \\ ab & \text{dla } i = 2 \\ \frac{a}{b} & \text{dla } i = 3 \\ 0 & \text{domyślnie} \end{cases}$$

$$3. y = \begin{cases} 32a + 16b & \text{dla } i = 10 \\ \frac{a - b}{4} & \text{dla } i = 20 \\ a \% 8 & \text{domyślnie} \end{cases}$$

$$4. y = \begin{cases} ab + cd - ex & \text{dla } x = 100 \\ \frac{-1000x^2 - 500abc}{dex + 1} & \text{dla } x = 200 \\ 42a - 16b + 8c & \text{domyślnie} \end{cases}$$

$$5. y = \begin{cases} |a + x| & \text{dla } x < 10 \\ 16a \% x & \text{dla } 10 \leq x < 20 \\ \frac{a - 20b}{x} & \text{dla } 20 \leq x < 30 \\ \frac{x}{30} & \text{dla } x \geq 30 \end{cases}$$

$$6. y = \begin{cases} \frac{-500a}{20b} & \text{dla } x < 5 \\ \frac{-244ax + 12b}{16x} & \text{dla } 5 \leq x < 10 \\ \frac{-128a + b - 16c}{|a + x + 1|} & \text{dla } x \geq 10 \end{cases}$$

Implementację powyższych funkcji należy rozważyć w następujących scenariuszach:

1. **Platforma x86.** Argumenty oraz wartość zwracana są typu 32-bit całkowitego ze znakiem (`int`).
2. **Platforma x64.** Argumenty oraz wartość zwracana są typu 64-bit całkowitego ze znakiem (`__int64`).
3. **Platforma x64.** Argumenty są typu 32-bit całkowitego ze znakiem (`int`). Wartość zwracana jest typu 64-bit całkowitego ze znakiem (`__int64`).

## 5 Implementacja pętli

Oblicz wartość podanych wyrażeń nie korzystając w sposób jawny z instrukcji `mul`, `imul`, `div`, ani `idiv`:

1.  $y = ab$
2.  $y = 40a$
3.  $y = 150a + b$
4.  $y = \frac{a+b}{10}$
5.  $y = \frac{a}{b}$ , gdzie  $b \neq 0$
6.  $y = \frac{a+b}{c}$ , gdzie  $c \neq 0$
7.  $y = ab - cd$
8.  $y = \frac{a}{18} - \frac{b}{34}$
9.  $y = a!$  (iteracyjnie)
10. Wyznacz  $n$ -ty wyraz ciągu Fibonacciego (iteracyjnie).
11. Policz, ile bitów o wartości 0 znajduje się w liczbie  $a$ .
12. Policz, ile bitów o wartości 1 znajduje się w liczbie  $a$ .

Implementację powyższych funkcji należy rozważyć w następujących scenariuszach:

1. **Platforma x86.** Argumenty oraz wartość zwracana są typu 32-bit całkowitego ze znakiem (`int`).
2. **Platforma x64.** Argumenty oraz wartość zwracana są typu 64-bit całkowitego ze znakiem (`__int64`).
3. **Platforma x64.** Argumenty są typu 32-bit całkowitego ze znakiem (`int`). Wartość zwracana jest typu 64-bit całkowitego ze znakiem (`__int64`).