

## Obliczanie najmniejszej wspólnej wielokrotności

### 1 Wprowadzenie

Najlepszy sposób na obliczenie najmniejszej wspólnej wielokrotności dwóch liczb wymaga obliczenia najpierw ich największego wspólnego dzielnika.

### 2 Największy wspólny dzielnik

```
def nwd_iteracja(a,b):
    while b:
        a, b = b, a%b
    return a
```

```
def nwd_rekurencja(a, b):
    if b > 0:
        return nwd(b, a%b)
    return a
```

Napisana wyżej funkcja przyjmuje dwie zmienne - "a" oraz "b". Pętla "while b:" wykonywana będzie tak długo, aż zmienna "b" nie osiągnie wartości "0".

Python3 umożliwia zamianę wartości dwóch zmiennych w jednej linii, bez użycia zmiennej tymczasowej.

Użycie zmiennej pomocniczej ułatwia późniejsze przetłumaczenie funkcji do innych języków programowania.

```
def nwd_iteracja(a,b):
    while b:
        temp = a
        a = b
        b = temp%b
    return a
```

### 3 Najmniejsza wspólna wielokrotność

Najmniejszą wspólną wielokrotność przy użyciu ich dzielnika można obliczyć za pomocą wzoru:

$$NWW(a, b) = \frac{a \times b}{NWD(a, b)}$$

Funkcja obliczająca NWW dwóch liczb w Pythonie wygląda więc tak:

```
def nww(a,b):
    nww = (a * b) / nwd(a,b)
    return nww
```

### 4 Skończona funkcja

Aby zawrzeć cały kod w jednej funkcji, trzeba zachować początkowe wartości "a" oraz "b". Tutaj można wykorzystać tymczasowe zmienne "a2" oraz "b2".

```
def nww_iteracja(a,b):
    a2 = a
    b2 = b
    while b2:
        temp = a2
        a2 = b2
        b2 = temp%b2
    return (a * b) / a2
```

```
def nww_rekurencja(a, b):
    b2 = b
    if b2 > 0:
        b2 = nwd(b, a%b2)
    return (a*b) / b2
```