

## IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

## KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

## TWÓJ KOSZYK

DODAJ DO KOSZYKA

## CENNIK I INFORMACJE

ZAMÓW INFORMACJE  
O NOWOŚCIACH

ZAMÓW CENNIK

## CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

# OpenOffice 2.0 PL. Funkcje arkusza kalkulacyjnego. Leksykon kieszonkowy

Autor: Bartosz Gajda  
ISBN: 83-246-0848-6  
Format: B6, stron: 272



Arkusz kalkulacyjny Calc to jeden ze składników dostępnego nieodpłatnie pakietu biurowego OpenOffice 2.0. Podobnie jak jego komercyjne odpowiedniki, Calc posiada ogromne możliwości w zakresie przeprowadzania nawet najbardziej złożonych obliczeń. Jego użytkownicy mogą tu wykorzystać nie tylko tworzone przez siebie formuły, ale także rozbudowany zbiór predefiniowanych funkcji. Funkcje te, podzielone na kategorie, wymagają jedynie odpowiedniego zdefiniowania argumentów – i to właśnie stanowi często przeszkodę w ich wykorzystaniu. W wielu przypadkach trudno zorientować się, czy argumentem jest pojedyncza komórka, czy też zakres komórek. Lakoniczny opis wyświetlany po wybraniu funkcji w programie rzadko okazuje się pomocny.

„OpenOffice 2.0 PL. Funkcje arkusza kalkulacyjnego. Leksykon kieszonkowy” zawiera zestawienie opisów wszystkich opcji, jakie oferuje swoim użytkownikom aplikacja OpenOffice Calc. W każdym rozdziale znajdziesz omówienie innej kategorii funkcji. Dowiesz się, jakie jest zastosowanie każdej z nich, poznasz liczbę i znaczenie argumentów funkcji oraz typ i interpretację zwracanej wartości. Każdy opis zawiera także przykład zastosowania funkcji, co znacząco ułatwia zrozumienie zadania przez nią realizowanego.

- Funkcje bazodanowe
- Funkcje operujące na datach i godzinach
- Funkcje finansowe
- Funkcje informacyjne
- Funkcje logiczne
- Funkcje matematyczne
- Funkcje macierzowe
- Funkcje statystyczne
- Funkcje arkusza kalkulacyjnego
- Funkcje tekstowe
- Inne funkcje

**Usprawnij i przyspiesz swoją pracę z arkuszem kalkulacyjnym!**



---

# Spis treści

Wstęp .....	5
1. Funkcje kategorii Baza danych .....	7
2. Funkcje kategorii Data i godzina .....	24
3. Funkcje kategorii Finanse .....	49
4. Funkcje kategorii Informacja .....	107
5. Funkcje kategorii Logiczne .....	122
6. Funkcje kategorii Matematyka .....	127
7. Funkcje kategorii Macierz .....	167
8. Funkcje kategorii Statystyka .....	181
9. Funkcje kategorii Arkusz kalkulacyjny .....	232
10. Funkcje kategorii Tekst .....	250
11. Funkcje kategorii Dodatek .....	FTP

# Rozdział 11. Funkcje kategorii Dodatek

## BESSELI

### Zastosowanie

Funkcja służy do obliczania wartości zmodyfikowanej funkcji Bessela dla czysto urojonych argumentów. Zmodyfikowana funkcja Bessela  $n$ -tego rzędu obliczana przez opisywaną funkcję ma postać:

$$I_n(x) = (i)^{-n} J_n(ix)$$

gdzie  $J_n$  jest funkcją Bessela J.

### Składnia

`BESSELI(liczba; rzqd)`

### Opis argumentów

*liczba*

liczba, dla której chcemy wyznaczyć wartość funkcji.

*rzqd*

rzęd funkcji Bessela.

## BESSELJ

### Zastosowanie

Funkcja służy do obliczania wartości funkcji Bessela.

### Składnia

`BESSELJ(liczba; rzqd)`

## Opis argumentów

*liczba*

liczba, dla której chcemy wyznaczyć wartość funkcji.

*rzqd*

rząd funkcji Bessela.

## BESSELK

### Zastosowanie

Funkcja służy do obliczania wartości zmodyfikowanej funkcji Bessela dla czysto urojonych argumentów. Zmodyfikowana funkcja Bessela  $n$ -tego rzędu obliczana przez opisywaną funkcję ma postać:

$$K_n(x) = \frac{p}{2} i^{n+1} [J_n(ix) + iY_n(ix)]$$

gdzie  $J_n$  oraz  $Y_n$  są odpowiednio funkcjami Bessela J i Y.

### Składnia

BESSELK(*liczba*; *rzqd*)

## Opis argumentów

*liczba*

liczba, dla której chcemy wyznaczyć wartość funkcji.

*rzqd*

rząd funkcji Bessela.

## BESSELY

### Zastosowanie

Funkcja służy do obliczania wartości zmodyfikowanej funkcji Bessela zwaney także funkcją Webera lub funkcją Neumanna. Zmodyfikowana funkcja Bessela  $n$ -tego rzędu obliczana przez opisywaną funkcję ma postać:

$$Y_n(x) = \lim_{\nu \rightarrow n} \frac{J_\nu(x) \cos(\nu\pi) - J_{-\nu}(x)}{\sin(\nu\pi)}$$

gdzie  $J_\nu$  jest funkcją Bessela J.

## Składnia

BESSELY(*liczba*; *rzqd*)

## Opis argumentów

*liczba*

liczba, dla której chcemy wyznaczyć wartość funkcji.

*rzqd*

rzqd funkcji Bessela.

## BIN2DEC

### Zastosowanie

Funkcja służy do konwersji liczby zapisanej w postaci binarnej do zapisu dziesiętnego.

## Składnia

BIN2DEC(*liczba*)

## Opis argumentów

*liczba*

liczba binarna, którą chcemy konwertować. Liczba może mieć maksymalnie 10 bitów. Bit najbardziej znaczący jest bitem znaku.

## Uwagi

Liczby ujemne są wprowadzane w zapisie dopełnienia do dwóch.

# BIN2HEX

## Zastosowanie

Funkcja służy do konwersji liczby zapisanej w postaci binarnej do zapisu szesnastkowego.

## Składnia

`BIN2HEX(liczba; miejsca)`

## Opis argumentów

*liczba*

liczba binarna, którą chcemy konwertować. Liczba może mieć maksymalnie 10 bitów. Bit najbardziej znaczący jest bitem znaku. Liczby ujemne są wprowadzane w zapisie dopełnienia do dwóch.

*miejsca*

liczba znaków, którą ma zwrócić funkcja.

## Uwagi

Jeżeli pominiemy argument *miejsca*, liczba zostanie przedstawiona za pomocą najmniejszej liczby znaków. Jeżeli argument *miejsca* określa liczbę znaków jako większą od najmniejszej koniecznej, liczba zostanie uzupełniona zerami (wstawianymi na jej początku). Liczby ujemne są reprezentowane w zapisie dopełnienia do dwóch. Jeśli argument *liczba* będzie liczbą ujemną, funkcja zwróci 10-bitową liczbę, w której najbardziej znaczący bit jest bitem znaku, a pozostałe bity reprezentują wartość. W takim przypadku argument *miejsca* jest ignorowany.

# BIN2OCT

## Zastosowanie

Funkcja służy do konwersji liczby zapisanej w postaci binarnej do zapisu ósemkowego.

## Składnia

```
BIN2OCT(liczba; miejsca)
```

## Opis argumentów

*liczba*

liczba binarna, którą chcemy konwertować. Liczba może mieć maksymalnie 10 bitów. Bit najbardziej znaczący jest bitem znaku. Liczby ujemne są wprowadzane w zapisie dopełnienia do dwóch.

*miejsca*

liczba znaków, którą ma zwrócić funkcja.

## Uwagi

Jeżeli pominiemy argument *miejsca*, liczba zostanie przedstawiona za pomocą najmniejszej liczby znaków. Jeżeli argument *miejsca* określa liczbę znaków jako większą od najmniejszej koniecznej, liczba zostanie uzupełniona zerami (wstawianymi na jej początku). Liczby ujemne są reprezentowane w zapisie dopełnienia do dwóch. Jeśli argument *liczba* będzie liczbą ujemną, funkcja zwróci 10-bitową liczbę, w której najbardziej znaczący bit jest bitem znaku, a pozostałe bity reprezentują wartość. W takim przypadku argument *miejsca* jest ignorowany.

## COMPLEX

### Zastosowanie

Funkcja służy do zapisu liczby zespolonej w postaci  $x+yi$  lub  $x+yj$  na podstawie podanej jako argumenty części rzeczywistej i urojonej.

## Składnia

```
COMPLEX(część_rzeczywista; część_urojona; jednostka_urojona)
```

## Opis argumentów

*część\_rzeczywista*

argument określa część rzeczywistą liczby zespolonej.

*część\_urojona*

argument określa część urojoną liczby zespolonej.

*jednostka\_urojona*

argument określa jednostkę urojoną, może przyjmować wartości *i* lub *j*.

## Uwagi

Jeżeli argument *jednostka\_urojona* zostanie pominięty, przyjmie wartość domyślną *i*.

## DEC2BIN

### Zastosowanie

Funkcja służy do konwersji liczby zapisanej w postaci dziesiętnej do zapisu binarnego.

### Składnia

DEC2BIN(*liczba*; *miejsca*)

### Opis argumentów

*liczba*

liczba dziesiętna, którą chcemy konwertować. Argument musi być liczbą większą od -512 i mniejszą od 511.

*miejsca*

liczba znaków, którą ma zwrócić funkcja.

## Uwagi

Jeżeli pominiemy argument *miejsca*, liczba zostanie przedstawiona za pomocą najmniejszej liczby znaków. Jeżeli argument *miejsca* określa liczbę znaków jako większą od najmniejszej koniecznej, liczba zostanie uzupełniona zerami (wstawianymi na początku). Liczby ujemne są reprezentowane w zapisie dopełnienia do dwóch. Jeśli argument *liczba* będzie liczbą ujemną, funkcja zwróci 10-bitową liczbę, w której najbardziej znaczący bit jest bitem znaku, a pozostałe bity reprezentują wartość. W takim przypadku argument *miejsca* jest ignorowany.



## DEC2HEX

### Zastosowanie

Funkcja służy do konwersji liczby zapisanej w postaci dziesiętnej do zapisu szesnastkowego.

### Składnia

```
DEC2HEX(liczba; miejsca)
```

### Opis argumentów

*liczba*

liczba dziesiętna, którą chcemy konwertować.

*miejsca*

liczba znaków, którą ma zwrócić funkcja.

### Uwagi

Jeżeli pominiemy argument *miejsca*, liczba zostanie przedstawiona za pomocą najmniejszej liczby znaków. Jeżeli argument *miejsca* określa liczbę znaków jako większą od najmniejszej koniecznej, liczba zostanie uzupełniona zerami (wstawianymi na początku). Liczby ujemne są reprezentowane w zapisie dopełnienia do dwóch. Jeśli argument *liczba* będzie liczbą ujemną, funkcja zwróci 10-znakową liczbę gdzie najbardziej znaczący bit jest bitem znaku, a pozostałe bity reprezentują wartość. W takim przypadku argument *miejsca* jest ignorowany.

## DEC2OCT

### Zastosowanie

Funkcja służy do konwersji liczby zapisanej w postaci dziesiętnej do zapisu ósemkowego.

### Składnia

```
DEC2OCT(liczba; miejsca)
```

## Opis argumentów

*liczba*

liczba dziesiętna, którą chcemy konwertować.

*miejsca*

liczba znaków, którą ma zwrócić funkcja.

## Uwagi

Jeżeli pominiemy argument *miejsca*, liczba zostanie przedstawiona za pomocą najmniejszej liczby znaków. Jeżeli argument *miejsca* określa liczbę znaków jako większą od najmniejszej koniecznej, liczba zostanie uzupełniona zerami (wstawianymi na początku). Liczby ujemne są reprezentowane w zapisie dopełnienia do dwóch. Jeśli argument *liczba* będzie liczbą ujemną, funkcja zwróci 10-znakową liczbę, w której najbardziej znaczący bit jest bitem znaku, a pozostałe bity reprezentują wartość. W takim przypadku argument *miejsca* jest ignorowany.

## DELTA

### Zastosowanie

Funkcja służy do sprawdzania, czy dwie liczby podane jako argumenty są sobie równe.

### Składnia

```
DELTA(liczba1; liczba2)
```

### Opis argumentów

*liczba1*

pierwsza z porównywanych liczb.

*liczba2*

druga z porównywanych liczb.

### Uwagi

Jeżeli pominiemy argument *liczba2*, przyjmie on wartość domyślną 0.

## ERF

### Zastosowanie

Funkcja służy do wyznaczania wartości całki błędu Gaussa w zadanych argumentami granicach.

### Składnia

$ERF(\textit{granica1};\textit{granica2})$

### Opis argumentów

*granica1*

argument określa dolną granicę całkowania.

*granica2*

argument określa górną granicę całkowania.

### Uwagi

Całka błędu Gaussa ma postać:

$$ERF(a,b) = \int_a^b \frac{2}{\sqrt{\pi}} e^{-t^2} dt$$

## ERFC

### Zastosowanie

Funkcja służy do wyznaczania wartości całki błędu Gaussa w granicach od podanej argumentem granicy do nieskończoności.

### Składnia

$ERFC(\textit{granica})$

### Opis argumentów

*granica*

argument określa dolną granicę całkowania.

## Uwagi

Funkcja oblicza następującą całkę:

$$ERFC(x) = \int_x^{+\infty} \frac{2}{\sqrt{\pi}} e^{-t^2} dt$$

## FACTDOUBLE

### Zastosowanie

Funkcja służy do obliczania silni liczby z przyrostami o 2.

### Składnia

`FACTDOUBLE(liczba)`

### Opis argumentów

*liczba*

liczba, której silnię chcemy policzyć.

## Uwagi

Dla parzystej liczby  $n$  jest obliczana następująca silnia:  $n*(n-2)* (n-4)* \dots *4*2$ , a dla nieparzystej  $n$  jest obliczana silnia:  $n*(n-2)*(n-4)* \dots *3*1$ .

## GESTEP

### Zastosowanie

Funkcja służy do testowania wartości liczbowej w taki sposób, że kiedy wartość testowana jest większa lub równa zadanej wartości progowej, funkcja zwraca wartość 1, a kiedy wartość testowana jest mniejsza, funkcja zwraca wartość 0.

## Składnia

```
GESTEP(liczba; próg)
```

## Opis argumentów

*liczba*

wartość, którą testujemy. Argument powinien być liczbą.

*próg*

wartość progowa. Argument powinien być liczbą.

## Uwagi

Jeżeli argument *próg* zostanie pominięty, przyjmie wartość domyślną 0.

## HEX2BIN

### Zastosowanie

Funkcja służy do konwersji liczby zapisanej w postaci szesnastkowej do zapisu binarnego.

## Składnia

```
HEX2BIN(liczba; miejsca)
```

## Opis argumentów

*liczba*

liczba szesnastkowa, którą chcemy konwertować.

*miejsca*

liczba znaków, którą ma zwrócić funkcja.

## Uwagi

Jeżeli pominiemy argument *miejsca*, liczba zostanie przedstawiona za pomocą najmniejszej liczby znaków. Jeżeli argument *miejsca* określa liczbę znaków jako większą od najmniejszej koniecznej, liczba zostanie uzupełniona zerami wiodącymi na początku. Liczby ujemne są reprezentowane

w zapisie dopełnienia do dwóch. Jeśli argument *liczba* będzie liczbą ujemną, funkcja zwróci 10-bitową liczbę, w której najbardziej znaczący bit jest bitem znaku, a pozostałe znaki reprezentują wartość. W takim przypadku argument *miejsca* jest ignorowany.

## HEX2DEC

### Zastosowanie

Funkcja służy do konwersji liczby zapisanej w postaci szesnastkowej do zapisu dziesiętnego.

### Składnia

```
HEX2DEC(liczba)
```

### Opis argumentów

*liczba*

liczba szesnastkowa, którą chcemy konwertować.

### Uwagi

Liczby ujemne są wprowadzane w zapisie dopełnienia do dwóch.

## HEX2OCT

Funkcja służy do konwersji liczby zapisanej w postaci szesnastkowej do zapisu ósemkowego.

### Składnia

```
HEX2BIN(liczba; miejsca)
```

### Opis argumentów

*liczba*

liczba szesnastkowa, którą chcemy konwertować.

*miejsca*

liczba znaków, którą ma zwrócić funkcja.

## Uwagi

Jeżeli pominiemy argument *miejsca*, liczba zostanie przedstawiona za pomocą najmniejszej liczby znaków. Jeżeli argument *miejsca* określa liczbę znaków jako większą od najmniejszej koniecznej, liczba zostanie uzupełniona zerami wiodącymi na początku. Liczby ujemne są reprezentowane w zapisie dopełnienia do dwóch. Jeśli argument *liczba* będzie liczbą ujemną, funkcja zwróci 10-znakową liczbę, w której najbardziej znaczący bit jest bitem znaku, a pozostałe bity reprezentują wartość. W takim przypadku argument *miejsca* jest ignorowany.

## IMABS

### Zastosowanie

Funkcja służy do wyznaczania modułu (wartości bezwzględnej) liczby zespolonej.

### Składnia

```
IMABS(liczba)
```

### Opis argumentów

*liczba*

liczba zespolona, której moduł chcemy wyznaczyć. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

```
=IMABS("4+3i")
```

zwróci wartość 5.

# IMAGINARY

## Zastosowanie

Funkcja służy do wyodrębniania części urojonej liczby zespolonej.

## Składnia

`IMAGINARY(liczba)`

## Opis argumentów

*liczba*

liczba zespolona, której część urojoną chcemy wyodrębnić. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

```
=IMAGINARY("4+3i")  
zwróci wartość 3.
```

# IMARGUMENT

## Zastosowanie

Funkcja służy do wyznaczania argumentu (kąta) liczby zespolonej. Wynik jest zwracany w radianach.

## Składnia

`IMARGUMENT(liczba)`

## Opis argumentów

*liczba*

liczba zespolona, której argument chcemy wyznaczyć. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.



## Uwagi

Przykładowa funkcja

```
=IMARGUMENT("2+2i")
```

zwróci wartość 0,79, co odpowiada  $\pi/4$  (45°).

## IMCONJUGATE

### Zastosowanie

Funkcja służy do określania liczby sprzężonej dla zadanej liczby zespolonej.

### Składnia

```
IMCONJUGATE(liczba)
```

### Opis argumentów

*liczba*

liczba zespolona, dla której chcemy wyznaczyć liczbę sprzężoną. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

```
=IMCONJUGATE("1+2i")
```

zwróci wartość 1-2i.

## IMCOS

Funkcja służy do wyznaczania cosinusa liczby zespolonej.

## Składnia

`IMCOS(liczba)`

## Opis argumentów

*liczba*

liczba zespolona, której cosinus chcemy wyznaczyć. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

`=IMCOS("5+5i")`

zwróci wartość 21.0505561816548+71.1552598809822i.

## IMDIV

### Zastosowanie

Funkcja służy do wyznaczania ilorazu dwóch liczb zespolonych.

## Składnia

`IMDIV(dzielna; dzielnik)`

## Opis argumentów

*dzielna*

liczba zespolona stanowiąca dzielną ilorazu. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

*dzielnik*

liczba zespolona stanowiąca dzielnik ilorazu. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

```
IMDIV("2+3i";"1+2i")  
zwróci wartość 1.6-0.2i.
```

## IMEXP

### Zastosowanie

Funkcja służy do wyznaczania postaci wykładniczej liczby zespolonej.

### Składnia

```
IMEXP(liczba)
```

### Opis argumentów

*liczba*

liczba zespolona, której postać wykładniczą chcemy otrzymać. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

```
IMEXP("2+3i")  
zwróci wartość -7.3151100949011+1.0427436562359i.
```

## IMLN

Funkcja służy do wyznaczania logarytmu naturalnego liczby zespolonej.

### Składnia

```
IMLN(liczba)
```

## Opis argumentów

*liczba*

liczba zespolona, której logarytm naturalny chcemy wyznaczyć. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

```
=IMLN("1+1i")
```

zwróci wartość 0.346573590279973+0.785398163397448i.

## IMLOG10

Funkcja służy do wyznaczania logarytmu dziesiętnego liczby zespolonej.

## Składnia

IMLOG10(*liczba*)

## Opis argumentów

*liczba*

liczba zespolona, której logarytm dziesiętny chcemy wyznaczyć. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

```
=IMLOG10("3+5i")
```

zwróci wartość 0.765739458521128+0.447486970040493i.

## IMLOG2

Funkcja służy do wyznaczania logarytmu o podstawie równej 2 liczby zespolonej.

## Składnia

`IMLOG2(liczba)`

## Opis argumentów

*liczba*

liczba zespolona, której logarytm o podstawie 2 chcemy wyznaczyć. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

`=IMLOG2("1+4j")`

zwróci wartość 2.04373142062517+1.91275056849686j.

## IMPOWER

Funkcja służy do wyznaczania potęgi liczby zespolonej.

## Składnia

`IMLOG2(liczba; wykładnik)`

## Opis argumentów

*liczba*

liczba zespolona, której potęgę chcemy wyznaczyć. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

*wykładnik*

wykładnik potęgi; może być dowolną liczbą rzeczywistą.

## Uwagi

Przykładowa funkcja

`=IMPOWER("1+2i";3)`

zwróci wartość -11-2i.

# IMPRODUCT

Funkcja służy do wyznaczania iloczynu liczb zespolonych.

## Składnia

`IMPRODUCT(liczba1;liczba2;...;liczba30)`

## Opis argumentów

*liczba1;liczba2;...;liczba30*

liczby zespolone (maksymalnie 30), których iloczyn chcemy wyznaczyć. Liczby powinny być podane w postaci  $x+yi$  lub  $x+yj$  i ujęte w cudzysłów. Argumenty mogą być odwołaniami do komórek lub zakresów komórek zawierających liczby zespolone.

## Uwagi

Przykładowa funkcja

`IMPRODUCT("1+2i";"2+3i" )`  
zwróci wartość  $-4+7i$ .

# IMREAL

## Zastosowanie

Funkcja służy do wyodrębniania części rzeczywistej liczby zespolonej.

## Składnia

`IMREAL(liczba)`

## Opis argumentów

*liczba*

liczba zespolona, której część rzeczywistą chcemy wyodrębnić. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

```
=IMREAL("4+3i")  
zwróci wartość 4.
```

## IMSIN

Funkcja służy do wyznaczania sinusa liczby zespolonej.

### Składnia

```
IMSIN(liczba)
```

### Opis argumentów

*liczba*

liczba zespolona, której sinus chcemy wyznaczyć. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

```
=IMSIN("5+5i")  
zwróci wartość -71.161721061921+21.0486448808835i.
```

## IMSQRT

Funkcja służy do wyznaczania pierwiastka kwadratowego liczby zespolonej.

### Składnia

```
IMSQRT(liczba)
```

## Opis argumentów

*liczba*

liczba zespolona, której pierwiastek kwadratowy chcemy wyznaczyć. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

```
=IMSQRT("4+6i")
```

zwróci wartość 2.36760454372431+1.26710349832363i.

## IMSUB

### Zastosowanie

Funkcja służy do wyznaczania różnicy dwóch liczb zespolonych.

### Składnia

```
IMSUB(liczba1; liczba2)
```

## Opis argumentów

*liczba1*

liczba zespolona, od której ma zostać odjęta liczba *liczba2*. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

*liczba2*

liczba zespolona, która ma zostać odjęta od liczby *liczba1*. Liczba powinna być podana w postaci  $x+yi$  lub  $x+yj$  i ujęta w cudzysłów.

## Uwagi

Przykładowa funkcja

```
=IMSUB("5-2i"; "2+3i")
```

zwróci wartość 3-5i.



## IMSUM

Funkcja służy do wyznaczania sumy liczb zespolonych.

### Składnia

```
IMSUM(liczba1; liczba2; ...; liczba30)
```

### Opis argumentów

*liczba1*; *liczba2*; ...; *liczba30*

liczby zespolone (maksymalnie 30), których sumę chcemy wyznaczyć. Liczby powinny być podane w postaci  $x+yi$  lub  $x+yj$  i ujęte w cudzysłów. Argumenty mogą być odwołaniami do komórek lub zakresów komórek zawierających liczby zespolone.

### Uwagi

Przykładowa funkcja

```
=IMSUM("-1-2i"; "-3-4i")  
zwróci wartość -4-6i.
```

## KONWERTUJ\_ADD

### Zastosowanie

Funkcja służy do konwersji wartości wyrażonej w jednym systemie miar do innego systemu miar.

### Składnia

```
KONWERTUJ_ADD(wartość; jednostka_wejściowa;  
jednostka_docelowa)
```

### Opis argumentów

*wartość*

wartość, którą chcemy konwertować, wyrażona w jednostkach określonych argumentem *jednostka\_wejściowa*.

### *jednostka\_wejsciowa*

argument określający jednostkę wartości *wartość*. Może przyjmować wartości symboli jednostek wymienionych w tabeli 11.1. Symbole jednostek należy ująć w cudzysłów. Dodatkowo przed symbolami jednostek z tabeli 11.1 można umieszczać wymienione w tabeli 11.2 przedrostki odpowiadające poszczególnym mnożnikom.

### *jednostka\_docelowa*

jednostka, w jakiej ma być podany wynik funkcji. Argument może przyjmować wartości symboli jednostek wymienione w tabeli 11.1. Symbole jednostek należy ująć w cudzysłów. Dodatkowo przed symbolami jednostek z tabeli 11.1 można umieszczać wymienione w tabeli 11.2 przedrostki odpowiadające poszczególnym mnożnikom.

Tabela 11.1. Dopuszczalne symbole jednostek wielkości fizycznych

Wielkość	Jednostka
Masa	g, sg, lbm, u, ozm, stone, ton, grain, pweight, hweight, shweight
Długość	m, mi, Nmi, in, ft, yd, ang, Pica, ell, parsec
Czas	yr, day, hr, mn, sec
Ciśnienie	Pa, atm, mmHg, Torr, psi
Siła	N, dyn, pond
Energia	J, e, c, cal, eV, HPh, Wh, BTU
Moc	W, HP, PS
Siła pola	T, ga
Temperatura	C, F, K, Reau, Rank
Objętość	l, tsp, tbs, oz, cup, pt, qt, gal, m3, mi3, Nmi3, in3, ft3, yd3, ang3, Pica3, barrel, bushel, regton, Schooner, Middy, Glass
Powierzchnia	m2, mi2, Nmi2, in2, ft2, yd2, ang2, Pica2, Morgen, ar, acre, ha
Prędkość	m/s, m/h, mph, kn, admkn

## Uwagi

### Przykładowa funkcja

=KONWERTUJ\_ADD(1; "atm"; "hPa"),

zwróci wynik 1013,25, co oznacza że jedna atmosfera odpowiada 1013,25 hektopaskalom.

Tabela 11.2. Przedrostki i odpowiadające im mnożniki

Przedrostek	Mnożnik
Y	$10^{24}$
Z	$10^{21}$
E	$10^{18}$
P	$10^{15}$
T	$10^{12}$
G	$10^9$
M	$10^6$
k	$10^3$
h	$10^2$
e	$10^1$
d	$10^{-1}$
c	$10^{-2}$
m	$10^{-3}$
u	$10^{-6}$
n	$10^{-9}$
p	$10^{-12}$
f	$10^{-15}$
a	$10^{-18}$
z	$10^{-21}$
y	$10^{-24}$

## OCT2BIN

### Zastosowanie

Funkcja służy do konwersji liczby zapisanej w postaci ósemkowej do zapisu binarnego.

### Składnia

`OCT2BIN(liczba;miejsca)`

## Opis argumentów

*liczba*

liczba ósemkowa, którą chcemy konwertować.

*miejsca*

liczba znaków, którą ma zwrócić funkcja.

## Uwagi

Jeżeli pominiemy argument *miejsca*, liczba zostanie przedstawiona za pomocą najmniejszej liczby znaków. Jeżeli argument *miejsca* określa liczbę znaków jako większą od najmniejszej koniecznej, liczba zostanie uzupełniona zerami wiodącymi na początku. Liczby ujemne są reprezentowane w zapisie dopełnienia do dwóch. Jeśli argument *liczba* będzie liczbą ujemną, funkcja zwróci 10-bitową liczbę, w której najbardziej znaczący bit jest bitem znaku, a pozostałe bity reprezentują wartość. W takim przypadku argument *miejsca* jest ignorowany.

## OCT2DEC

### Zastosowanie

Funkcja służy do konwersji liczby zapisanej w postaci ósemkowej do zapisu dziesiętnego.

### Składnia

OCT2DEC(*liczba*)

## Opis argumentów

*liczba*

liczba ósemkowa, którą chcemy konwertować.

## Uwagi

Liczby ujemne są wprowadzane w zapisie dopełnienia do dwóch.

## OCT2HEX

Funkcja służy do konwersji liczby zapisanej w postaci ósemkowej do zapisu szesnastkowego.

### Składnia

```
DEC2HEX(liczba; miejsca)
```

### Opis argumentów

*liczba*

liczba ósemkowa, którą chcemy konwertować.

*miejsca*

liczba znaków, którą ma zwrócić funkcja.

### Uwagi

Jeżeli pominiemy argument *miejsca*, liczba zostanie przedstawiona za pomocą najmniejszej liczby znaków. Jeżeli argument *miejsca* określa liczbę znaków jako większą od najmniejszej koniecznej, liczba zostanie uzupełniona zerami (wstawianymi na początku). Liczby ujemne są reprezentowane w zapisie dopełnienia do dwóch. Jeśli argument *liczba* będzie liczbą ujemną, funkcja zwróci 10-znakową liczbę, w której najbardziej znaczący bit jest bitem znaku, a pozostałe bity reprezentują wartość. W takim przypadku argument *miejsca* jest ignorowany.