

# Podstawy Programowania - standardowy strumień wejścia/wyjścia

Dawid Jabłoński

24 stycznia 2023

# Spis treści

- 1 Funkcja: printf
- 2 Kody formatujące
- 3 Przykład: printf
- 4 Kody formatujące długość
- 5 Funkcja: scanf
- 6 Przykład: scanf

# Funkcja: printf()

## Blok

```
int printf( tekst_sterujący , argument_1 , argument_2 , ... ) ;
```

tekst sterujący - jest to stała łańcuchowa (w cudzysłowach) zawierająca:

- zwykłe znaki (które są po prostu kopiowane na ekran)
- kody formatujące kolejnych argumentów

# Kody formatujące

- `%g` - liczba zmiennoprzecinkowa (krótszy z formatów `%f` `%e`)

# Kody formatujące

- `%g` - liczba zmiennoprzecinkowa (krótszy z formatów `%f` `%e`)
- `%u` - liczba dziesiętna bez znaku

# Kody formatujące

- `%g` - liczba zmiennoprzecinkowa (krótszy z formatów `%f` `%e`)
- `%u` - liczba dziesiętna bez znaku
- `%x` - liczba w kodzie szesnastkowym (bez znaku)

# Kody formatujące

- `%g` - liczba zmiennoprzecinkowa (krótszy z formatów `%f` `%e`)
- `%u` - liczba dziesiętna bez znaku
- `%x` - liczba w kodzie szesnastkowym (bez znaku)
- `%o` - liczba w kodzie ósemkowym (bez znaku)

# Kody formatujące

- `%g` - liczba zmiennoprzecinkowa (krótszy z formatów `%f` `%e`)
- `%u` - liczba dziesiętna bez znaku
- `%x` - liczba w kodzie szesnastkowym (bez znaku)
- `%o` - liczba w kodzie ósemkowym (bez znaku)
- `l` - przedrostek (long) stosowany przed: `d` `u` `x` `o`



# Kody formatujące cd.

- `%c` - pojedynczy znak

# Kody formatujące cd.

- %c - pojedynczy znak
- %s - łańcuch znaków

# Kody formatujące cd.

- %c - pojedynczy znak
- %s - łańcuch znaków
- %d - liczba dziesiętna ze znakiem

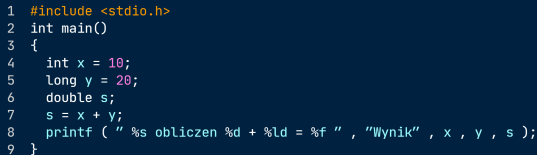
# Kody formatujące cd.

- %c - pojedynczy znak
- %s - łańcuch znaków
- %d - liczba dziesiętna ze znakiem
- %f - liczba zmiennoprzecinkowa (notacja dziesiętna)

# Kody formatujące cd.

- `%c` - pojedynczy znak
- `%s` - łańcuch znaków
- `%d` - liczba dziesiętna ze znakiem
- `%f` - liczba zmiennoprzecinkowa (notacja dziesiętna)
- `%e` - liczba zmiennoprzecinkowa (notacja wykładnicza)

# Przykład

A screenshot of a terminal window with a dark blue background and light green text. The window has three colored window control buttons (red, yellow, green) in the top-left corner. The code is a C program that includes <stdio.h>, defines a main function, and declares variables x (int), y (long), and s (double). It assigns x = 10 and y = 20, calculates s = x + y, and prints the result using printf with format specifiers %d, %ld, and %f. The output is "Wynik 10 20 30.000000".

```
1 #include <stdio.h>
2 int main()
3 {
4     int x = 10;
5     long y = 20;
6     double s;
7     s = x + y;
8     printf ( " %s obliczen %d + %ld = %f " , "Wynik" , x , y , s );
9 }
```

Rysunek 1: Efekt na ekranie: Wynik obliczen  $10 + 20 = 30.000000$

# Kody formatujące długość

Aby określić ilość drukowanych cyfr do kodu formatującego można dodać kody długości: %Xd %X.Xf

Kod	Działanie
%4d:	liczba dziesiętna na czterech pozycjach
%10f:	liczba rzeczywista na 10 pozycjach
%10.2f:	liczba rzeczywista na 10 pozycjach, 2 cyfry po przecinku
%.3f:	liczba rzeczywista z dokładnością do 3 cyfr po przecinku

# Funkcja: scanf()

## Blok

```
int scanf( tekst_sterujący , adres_1 , adres_2 , ... ) ;
```

& - operator adresowania (zwraca adres zmiennej podanej po operatorze)

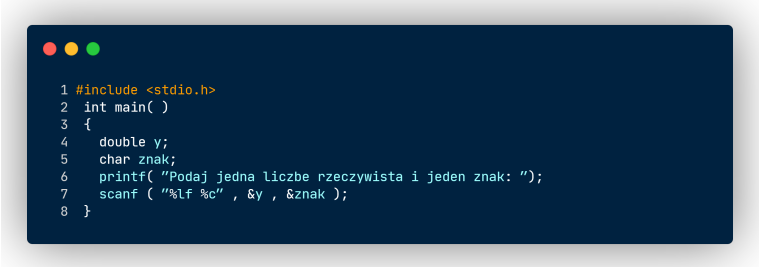


# Przykład

Aby uzyskać w terminalu komunikację z użytkownikiem

Wydruk - Podaj jedna liczbę rzeczywista i jeden znak:

Odczyt - 456.789 a



```
1 #include <stdio.h>
2 int main( )
3 {
4     double y;
5     char znak;
6     printf( "Podaj jedna liczbe rzeczywista i jeden znak: ");
7     scanf ( "%lf %c" , &y , &znak );
8 }
```

Rysunek 2: Wynik wczytywania:  $y == 456.789$ ,  $\text{znak} == 'a'$

- Marek Piasecki - Wykład 2