

Wprowadzanie i wyprowadzanie danych

getchar

Definicja: `char getchar () ;`

Wczytuje kolejny opis klawisza z klawiatury (oczekiwanie), bez echa.

Biblioteka: `stdio.h`

Wyjście: Kod ASCII lub 0

Przykład:

```
char nowy;  
nowy = getchar ( ) ;
```

scanf

Definicja:

```
int scanf ( const char *format,  wskaźnik,  
            wskaźnik, ... );
```

Wczytuje kolejne pola (ciągi znaków) ze stdin do napotkania białego znaku lub znaku końca linii, z echem.

format - tekst będący ciągiem wzorców konwersji.

Wzorzec konwersji:

% [*] [szerokość] [modyfikator] typ

szerokość - maksymalna liczba znaków (opcjonalnie)

modyfikator - jeden z: **hh**, **h**, **l**, **ll**, **j**, **z**, **t**, **L** (opcjonalnie)

typ - definiuje typ argumentu, np. **i**, **d**, **f**, **s**

Biblioteka: **stdio.h**

Wyjście: Liczba wczytanych pól znaków

Przykład:

```
int liczba_sztuk;  
double szerokosc;
```

```
scanf( " %d", &liczba_sztuk );  
scanf( " %lf", &szerokosc );  
scanf( " %lf%d", &szerokosc, &liczba_sztuk );
```

```
int lampy, krzesla, *wsk = &krzesla;  
float temp;  
double cena;  
char opcja;
```

```
scanf( " %d%d%f%lf", &lampy, wsk, &temp, &cena );
```

```
// 1 5 SP 3 4 7 Enter - 2 5 . 4 Enter  
// 3 . 9 9 Enter
```

```
// lampy == 15   krzesla == 347  
// temp == -25.4   cena == 3.99
```

Znak / ciąg znaków

wczytywanie pojedynczych znaków:

```
scanf ( "%c" , &znak ) ;
```

wczytywanie ciągów znaków:

```
char tekst [16];    // tablica 16 elementowa  
scanf ( "%s" , tekst ) ;
```

putchar

Definicja: `int putchar (int c) ;`

Wysyła znak do stdout.

Biblioteka: `stdio.h`

Wyjście: Znak (argument funkcji) c lub EOF (end-of-line)

Przykład:

```
char cc = 'R' ;  
putchar ( cc ) ; // 'R' pojawi się na ekranie
```

puts

Definicja: `int puts (char *napis);`

Wysyła do stdout ciąg znaków oraz znak końca linii.

Biblioteka: `stdio.h`

Wyjście: Ciąg znaków oraz znak końca linii lub EOF

Przykład:

```
#include <stdio.h>

void main( )
{
    char *nn = "Ciąg znakow.";
    puts(nn);
}
```

printf

Definicja:

```
int printf ( const char *format,  wyrażenie,  
            wyrażenie, ... );
```

Wysyła do stdout ciągi znaków reprezentujących wartości wyrażeń.

format zawiera:

- znaki przesyłane bezpośrednio do stdout
- oraz wzorce konwersji

Biblioteka: `stdio.h`

Przykład:

```
int liczba_kolorow = 256;  
printf( "%d", liczba_kolorow );
```

```
double objetosc = 15.72;  
printf( "%lf", objetosc );
```

```
char *tekst = "Dokumentacja.";   
printf ( "%s", tekst );
```

Wzorzec konwersji

`% [flaga] [szerokość] [precyzja] [modyfikator] typ`

flaga :

- uzupełnianie znakami spacji z prawej strony
- + wyprowadzanie znaku liczby
- SP znak spacji zamiast znaku plus

szerokość :

- minimalna liczba znaków dla liczb
- maksymalna liczba znaków dla ciągów znaków

precyzja :

. liczba_miejsc_po_kropce

modyfikator, typ - wyjaśnione wcześniej

```
int Alfa = 5;
float Beta = 12.45;

printf("Wynik: \n Alfa = %d,\t Beta = %f\n",
        Alfa, Beta + 500);

// Wynik:
// Alfa = 5,   Beta = 512.450000

char opcja = 'X';
char *Napis = "Opis programu.";

printf("Wybrano opcję %c : %31s", opcja, Napis);
// Wybrano opcję X : Opis programu.
```

```
int koty = 2, *wsk_k = &koty;  
float test = 23.345678;  
double wynik = -0.01234567;
```

```
printf("Liczba kotów : %d", *wsk_k );  
printf("\nWynik testu = %12.3f\n"  
       "Razem = %.5lf\n", test + 5, wynik );
```

```
// Liczba kotów : 2  
// Wynik testu =          28.345  
// Razem = -0.01234
```