

1. Wprowadzanie danych z klawiatury – funkcja scanf

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

Biblioteka `stdio.h`

Działanie Funkcja `scanf` wczytuje kolejne pola (ciągi znaków), ograniczone znakiem spacji ' ' lub nowej linii '\n'. Obrazy wczytywanych znaków są wyświetlane na ekranie monitora. Liczba wczytywanych pól i sposób ich przetwarzania są zadawane za pomocą pierwszego argumentu funkcji `scanf`, oznaczonego identyfikatorem format, którego wartością jest ciąg znaków (teskt). Ciąg ten składa się z pewnej liczby wzorców konwersji – liczba wzorców występujących w formacie określa liczbę pól wczytywanych ze strumienia wejściowego (klawiatury) i liczbę wskaźników będących dalszymi argumentami funkcji `scanf`. Wzorzec konwersji ma następującą postać:

`% [*] [szerokość] [prefiks] znak_konwersji`

Każdy wzorzec rozpoczyna się zawsze od znaku procenta %, składniki umieszczone w nawiasach prostokątnych nie muszą występować. Znak gwiazdki * użyty we wzorcu nakazuje pominąć kolejne pole znaków ze strumienia wejściowego. *Szerokość* określa maksymalną liczbę znaków, które mają być pobrane z kolejnego pola. Pole to może być krótsze niż zadana szerokość – w takim przypadku przetworzone będą jedynie znaki znajdujące się przed znakiem spacji lub nowej linii. *Prefiks* i *znak_konwersji* określają sposób przetworzenia znaków pola – wynikiem tego przetwarzania jest najczęściej wartość binarna liczby określonego typu. Kolejne wzorce konwersji są powiązane z kolejnymi wskaźnikami będącymi dalszymi argumentami funkcji `scanf` – obliczona wartość binarna liczby jest wprowadzana do obszaru pamięci wskazanego przez związany z danym wzorcem wskaźnik. Zakłada się przy tym, że typ wskaźnika związanego z wzorcem jest taki sam, jak typ konwersji zadanej w tym wzorcu. W tabelach podano znaczenie poszczególnych znaków konwersji i prefiksów

znak k.	wejściowe pole znaków	typ wskaźnika
d	liczba całkowita dziesiętna	int*
D	liczba całkowita dziesiętna	long*
o	liczba całkowita oktalna	int*
O	liczba całkowita oktalna	long*
i	liczba całkowita dziesiętna, oktalna lub heksadecymalna	int*
l	liczba całkowita dziesiętna, oktalna lub heksadecymalna	long*
u	liczba całkowita dziesiętna bez znaku	unsigned int*
U	liczba całkowita dziesiętna bez znaku	unsigned long*
x	liczba całkowita heksadecymalna	int*
X	liczba całkowita heksadecymalna	long*
e, E	liczba zmiennopozycyjna	float*
f	liczba zmiennopozycyjna	float*
g, G	liczba zmiennopozycyjna	float*
s	ciąg znaków	char*
c	znak	char*

prefiks	znaki konwersji	typ wskaźnika
h	d, i, o, u, x	short*
l	d, i, o, u, x	long*
	e, f, g	double*
L	e, f, g	long double*

Pola reprezentujące liczby (poza konwersją u, U) mogą rozpoczynać się od znaków plus + lub minus -. W przypadku wczytywania ciągu znaków (konwersja s) kody kolejnych znaków są lokowane w kolejnych bajtach pamięci, począwszy od bajtu wskazanego przez wskaźnik związany z danym wzorcem konwersji. Konieczne więc jest zarezerwowanie odpowiednio dużego obszaru pamięci, w którym znaki te będą umieszczane. Po ostatnim wczytanym znaku (różnym od znaku spacji lub nowej linii) jest umieszczany znak końca ciągu o wartości 0. Za pomocą konwersji s nie można wczytać tekstu złożonego z kilku słów oddzielonych spacjami. W takim przypadku można skorzystać z konwersji wyliczeniowej o postaci:

[--]

W nawiasach klamrowych podano przedział znaków ASCII (od znaku spacji do znaku tyldy), który będzie akceptowany i wprowadzany. Pierwszy znak spoza tego przedziału (czyli np. znak nowej linii) kończy wprowadzany tekst. Podczas wprowadzania

znaków z klawiatury można używać klawisza Backspace do kasowania poprzednio wprowadzonych znaków. *Prefiks* umieszczony przed znakiem konwersji redefiniuje znaczenie tego znaku.

Wynik Wynikiem funkcji `scanf` jest liczba pól znaków, które zostały poprawnie wczytane, poddane konwersji, a obliczone wartości zostały zapamiętane.

Przykład

```
#include <stdio.h>

void main ( void )
{
    .....
    char cZnak ;
    int nElement, nSkładnik, *pnAdres = &nSkładnik ;
    int anTablica [ 20 ] ;
    long lDługi, lPodłużny ;
    float flUłamek ;
    double dbDokładny ;
    long double ldPrecyzyjny ;
    char *szFormat = "%f%lf%Lf" ;
    char *szSłowo = new char [ 10 ] ;
    char acZnaki [ 16 ] ;
    char acTekst [ 65 ] ;
    //
    scanf ( "%c%d", &cZnak, &nElement); // identyfikatory
    scanf ( "%i", pnAdres );           // wskaźnik
    scanf ( "%d%d", &anTablica [ 0 ], &anTablica [ 19 ] ) ;
    scanf ( "%D%ld", &lDługi, &lPodłużny ) ;
    scanf ( szFormat, &flUłamek, &dbDokładny, &ldPrecyzyjny ) ;
    scanf( "%s%s", szSłowo, acZnaki ) ;
    scanf ( "[%--]", acTekst ) ;
    .....
}
```

Uwaga Funkcja `scanf` przerywa działanie w przypadku napotkania znaku niedozwolonego (np. znaku litery w liczbie całkowitej). Znak ten nie jest usuwany ze strumienia wejściowego, co niekiedy powoduje zapętlenie programu. Sytuacji tej można uniknąć, sprawdzając liczbę przetworzonych poprawnie pól znakowych, przekazywaną jako wynik funkcji `scanf`.

2. Wyprowadzanie danych na monitor ekranowy – funkcja printf

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

Biblioteka `stdio.h`

Działanie Funkcja `printf` umożliwia wyprowadzanie ciągów znaków reprezentujących wartości wyrażeń, które są jej argumentami. Sposób wyprowadzania określa argument `format` będący ciągiem znaków zawierającym:

- znaki przesyłane bezpośrednio na ekran monitora,
- wzorce konwersji definiujące sposób przetwarzania wartości wyrażeń na ciągi znaków.

Liczba wzorców konwersji powinna być równa liczbie wyrażeń, kolejne wzorce określają sposób przetwarzania wartości kolejnych wyrażeń. Wzorzec konwersji ma postać:

`% [opis] [szerokość] [. precyzja] [prefiks] znak_konwersji`

Wzorzec konwersji rozpoczyna się od znaku procentu %, składniki zapisane w nawiasach kwadratowych mogą nie występować. *Prefiks* i *znak_konwersji* określają sposób przekształcania wartości wyrażenia odpowiadającego danemu wzorcowi konwersji na ciąg znaków – składniki te muszą być dostosowane do typu wartości wyrażenia.

znak k.	typ wyrażenia	wynik konwersji
d	int	liczba dziesiętna ze znakiem
i	int	liczba dziesiętna ze znakiem
o	unsigned	liczba oktalna bez znaku
u	unsigned	liczba dziesiętna bez znaku
x	unsigned	liczba heksadecymalna bez znaku (a ... f)
X	unsigned	liczba heksadecymalna bez znaku (A ... F)
f	float	liczba rzeczywista ze znakiem o postaci ddd.ddd
e	float	liczba rzeczywista ze znakiem o postaci d.ddde[+/-]ddd
g	float	konwersja f lub e, zależnie od wartości argumentu
E	float	jak konwersja e z użyciem litery E do oznaczenia wykładnika
G	float	jak konwersja g z użyciem litery E do oznaczenia wykładnika
s	char*	ciąg znaków
c	char	pojedynczy znak

prefiks	znaki konwersji	typ wyrażenia
h	d, i, o, u, x, X	short
l	d, i, o, u, x, X e, E, f, g, G	long double
L	e, E, f, g, G	long double

W przypadku konwersji s, przeznaczonej dla ciągów znaków, wyprowadzanych jest co najwyżej tyle znaków, ile wskazano za pomocą składnika szerokość. Gdy składnik ten nie występuje we wzorcu, wyprowadzane są wszystkie znaki ciągu. Znak nowej linii '\n' jest zamieniany przez funkcję printf na parę znaków CR i LF, co powoduje przesunięcie kursora na początek następnego wiersza (z ewentualnym przewinięciem ekranu). Poprawnie są interpretowane również znaki sterujące: '\t' (tabulacja pozioma HT), '\r' (powrót kursora CR), '\b' (cofniecie kursora ze zmazaniem poprzedniego znaku BS) i '\a' (dzwonek BELL). Składnik szerokość określa

minimalną liczbę znaków, które zostaną wyprowadzone. Większa liczba znaków zostanie wyprowadzona, gdy w wyniku konwersji otrzymano ciąg znaków dłuższy od podanej szerokości. Jeżeli natomiast wynik konwersji zawiera mniej znaków, to zostanie on uzupełniony znakami spacji. Składnik opis jest jednym znakiem:

- uzupełnianie znakami spacji z prawej strony; gdy opis nie występuje, uzupełnia się z lewej strony,
- + wyprowadzanie znaku liczby (plus albo minus); gdy opis nie występuje, wyprowadzany jest jedynie minus,

spacja wyprowadzanie znaku spacji zamiast znaku plus liczby.

Składnik *precyzja* rozpoczyna się od znaku kropki i jest wartością całkowitą określającą liczbę miejsc po kropce dziesiętnej wyprowadzanej wartości zmiennopozycyjnej lub liczbę znaków podczas wyprowadzania ciągu znaków (konwersja s). Gdy precyzja nie jest określona, wyprowadzanych jest 6 miejsc po kropce dziesiętnej.

Wynik Wynikiem funkcji printf jest łączna liczba wyprowadzonych znaków.

Przykład

```
#include <stdio.h>

void main(void)
{
    .....
    int nBok = 15 ;
    printf ( "Wartość zmiennej nBok wynosi %d", nBok ) ;
    printf ("%d", nBok * nBok + 2 ) ;
    int anSzyk [ 5 ] = { 1, 2, 3, 4, 5 } ;
    for ( int nKolejny = 0 ; nKolejny < 5 ; nKolejny++ )
        printf ( "\nSzyk [%d]=%d", nKolejny, anSzyk [ nKolejny ] ) ;
    //
    float flSkok = 151.37825;
    printf ( "%+12.3f", flSkok ) ;    // +151.378
    printf ( "%E", sin ( 1 - 1 / flSkok ) * cos ( 1 / exp( flSkok ) ) ) ;
    //
    long lCzas = 3782505 ;
    double dbKsiężyc = 27E38 ;
    long double ldMikrus = -5E-127 ;
    printf ( "%ld \t %le \t %Le", lCzas, dbKsiężyc, ldMikrus ) ;
    //
```

```
char *szNapis = "raz dwa trzy" ;  
printf( "%s", szNapis ) ;      // wszystkie znaki  
printf ( "%.7s", szNapis ) ;  // 7 początkowych znaków  
printf ( "\n" ) ;            // nowa linia  
...  
}
```