

# Header ctype.h

Header	Opis zawartych funkcji	Przykładowe funkcje
ctype.h	Operacje na typach znakowych	isalnum(), isalpha(), iscntrl(), isdigit(), isxdigit(), isgraph(), islower(), isupper(), isprint(), ispunct(), isspace(), isblank(), tolower(), toupper()

## #include<ctype.h>

- **isalnum(znak)** - sprawdza czy znak jest alfanumeryczny (A-Z,a-z,0-9)
- **isalpha(znak)** - sprawdza czy znak jest znakiem alfabetu (A-Z,a-z)
- **iscntrl(znak)** - sprawdza czy znak jest znakiem kontrolnym, np. [Enter]
- **isdigit(znak)** - sprawdza czy znak jest znakiem numerycznym (cyfrą)
- **isxdigit(znak)** - sprawdza czy znak może być cyfrą w zapisie systemu szesnastkowego
- **isgraph(znak)** - sprawdza czy znak jest znakiem graficznym, np. 7, 7, ll, T, litery, cyfry, itd.
- **islower(znak)** - sprawdza czy znak jest małą literą alfabetu
- **isupper(znak)** - sprawdza czy znak jest wielką literą alfabetu
- **isprint(znak)** - sprawdza czy znak jest możliwy do wydruku
- **ispunct(znak)** - sprawdza czy znak jest znakiem interpunkcyjnym
- **isspace(znak)** - sprawdza czy znak jest 'białym' znakiem (spacja, enter, tab, ...)
- **isblank(znak)** - sprawdza czy znak jest spacją lub tab'em
- **tolower(znak)** - zmienia znak na małą literę jeżeli był on wielką literą
- **toupper(znak)** - zmienia znak na wielką literę jeżeli był on małą literą

// Operacje na pojedynczych znakach

```
#include<stdio.h>
#include<ctype.h>           // ten temat
#include<string.h>          // dla strlen()
#include<stdlib.h>          // dla atoi()
#include<locale.h>          // dla 'setlocale()'

char znak_a = 'a';          // test na isalnum(), isalpha(), islower(), isprint()
char znak_A = 'A';          // test na isalnum(), isalpha(), isupper(), isprint()
char znak_Enter = '\x0D';   // test na iscntrl() czyli: czy to znak kontrolny?
                             // ASCII od 0 do 31 i 127 (delete)

char znak_5 = '5';          // test na isdigit() czyli: czy to cyfra?
char znak_f = 'f';          // test na isxdigit() czyli: czy to cyfra zapisu
                             // szesnastkowego? (jedna z 0123456789abcdefABCDEF)

char znak_gr = 'x';         // test na isgraph() czyli: czy to znak graficzny?
// zestaw znaków graficznych: ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9
//                               : ; ? @ A B C D E F G H I J K L M N O P Q R S T
//                               U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l
//                               m n o p q r s t u v w x y z { | } ~ (spacja)

char znak_przecinek = ',';  // test na ispunct() czyli: czy to znak interpunkcyjny, jeden z:
                             // ! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~

char znak_spacja = ' ';     // test na isspace(), isblank()
char znaki_liczba[] = "12345"; // test na numeryczność łańcucha znakowego
int flaga = 0;              // flaga dla tego łańcucha znakowego
char nazwa[] = "James Bond, agent 007/J23"; // test na tolower() i toupper()
char nazwa_1[25];           // łańcuch pomocniczy dla tolower() i toupper()

int main(void)
```

```

{ setlocale(LC_CTYPE, "Polish"); //polskie znaki

printf("\n ***** Operacje na pojedynczych znakach *****\n");
printf("\n ----- Test na pojedynczych znakach ----- \n");

printf("\n - - - isalnum(), isalpha(), islower(), isupper(), isprint() - - - \n");
printf("isalnum(znak) - sprawdza czy 'znak' jest alfanumeryczny (A-Z,a-z,0-9).\n");
printf("isalpha(znak) - sprawdza czy 'znak' jest znakiem alfabetu (A-Z,a-z).\n");
printf("islower(znak) - sprawdza czy 'znak' jest małą literą alfabetu.\n");
printf("isupper(znak) - sprawdza czy 'znak' jest wielką literą alfabetu.\n");
printf("isprint(znak) - sprawdza czy 'znak' jest możliwy do wydruku.\n\n");

if( isalnum(znak_a) && isalpha(znak_a) && islower(znak_a) && isprint(znak_a) )
    printf("Znak '%c' jest alfanumeryczny, alfabetyczny i małą literą, i jest możliwy do \
wydruku.\n", znak_a);
else
    printf("Znak %c nie przeszedł wymaganych testów.\n", znak_a);

if(isupper(znak_a) ) // nieprawda, bo to małe 'a'
    printf("Znak '%c' jest wielką literą.\n", znak_a);
else
    printf("Znak '%c' nie przeszedł wymaganych testów bo nie jest wielką literą.\n", znak_a);

printf("\n");

if( isalnum(znak_A) && isalpha(znak_A) && isupper(znak_A) && isprint(znak_A) )
    printf("Znak '%c' jest alfanumeryczny, alfabetyczny i wielką literą, i jest możliwy do \
wydruku.\n", znak_A);
else
    printf("Znak '%c' nie przeszedł wymaganych testów.\n", znak_A);

if(islower(znak_A) ) // nieprawda, bo to wielkie 'A'
    printf("Znak '%c' jest małą literą.\n", znak_A);
else
    printf("Znak '%c' nie przeszedł wymaganych testów bo nie jest małą literą.\n", znak_A);

printf("\n - - - - - iscntrl() - - - - - \n");
printf("iscntrl(znak) - sprawdza czy 'znak' jest znakiem kontrolnym, np. [Enter].\n\n");

if( iscntrl(znak_Enter) )
    printf("[Enter] - Badany znak jest znakiem sterującym - ASCII od 0 do 31 włącznie i \
127.\n");
else
    printf("[Enter] - Badany znak nie przeszedł wymaganych testów.\n");

if( iscntrl(znak_a) )
    printf("Znak '%c' jest znakiem sterującym - ASCII od 0 do 31 włącznie.\n", znak_a);
else
    printf("Znak '%a' nie przeszedł wymaganych testów bo nie jest znakiem sterującym.\n",
znak_a);

printf("\n - - - - - isdigit() - - - - - \n");
printf("isdigit(znak) - sprawdza czy 'znak' jest znakiem numerycznym (cyfrą, 0-9).\n\n");

if( isdigit(znak_5) )
    printf("Znak '%c' jest cyfrą.\n", znak_5);
else
    printf("Znak '%c' nie przeszedł wymaganych testów bo nie jest cyfrą.\n", znak_5);

if( isdigit(znak_a) )
    printf("Znak '%c' jest cyfrą.\n", znak_a);
else
    printf("Znak '%c' nie przeszedł wymaganych testów bo nie jest cyfrą.\n", znak_a);

printf("\nSprawdzenie czy łańcuch znakowy da się zamienić na liczbę naturalną:\n");

```

```

for(int i=0; i<strlen(znaki_liczba); i++)
    if( !isdigit(znaki_liczba[i]) )
    { printf("łańcuch znakowy '%s' nie jest liczbą.\n", znaki_liczba);
      flaga = 1;
      break;
    }
if(flaga==0)
    printf("łańcuch znakowy '%d' dał się zamienić na liczbę naturalną.\n", atoi(znaki_liczba)
);

printf("\n - - - - - isxdigit() - - - - - \n");
printf("isxdigit(znak) - sprawdza czy 'znak' może być cyfrą zapisu szesnastkowego.\n\n");

if( isxdigit(znak_f) )
    printf("Znak '%c' jest cyfrą systemu szesnastkowego.\n", znak_f);
else
    printf("Znak '%c' nie przeszedł wymaganych testów bo nie jest cyfrą systemu \
szesnastkowego. \n", znak_f);

if( isxdigit(znak_przecinek) )
    printf("Znak '%c' jest cyfrą systemu szesnastkowego.\n", znak_przecinek);
else
    printf("Znak '%c' nie przeszedł wymaganych testów bo nie jest cyfrą systemu \
szesnastkowego. \n", znak_przecinek);

printf("\n - - - - - isgraph() - - - - - \n");
printf("isgraph(znak) - sprawdza czy 'znak' jest znakiem graficznym.\n\n");

if( isgraph(znak_gr) )
    printf("Znak '%c' jest znakiem graficznym.\n", znak_gr);
else
    printf("Znak '%c' nie przeszedł wymaganych testów bo nie jest znakiem graficznym.\n",
znak_gr);

if( isgraph(znak_Enter) )
    printf("[Enter] - Badany znak jest znakiem graficznym.\n");
else
    printf("[Enter] - Badany znak nie przeszedł wymaganych testów bo nie jest znakiem \
graficznym.\n");

printf("\n - - - - - ispunct() - - - - - \n");
printf("ispunct(znak) - sprawdza czy 'znak' jest znakiem interpunkcyjnym.\n\n");

if( ispunct(znak_przecinek) )
    printf("Znak '%c' jest znakiem interpunkcyjnym.\n", znak_przecinek);
else
    printf("Znak '%c' nie przeszedł wymaganych testów bo nie jest znakiem interpunkcyjnym.\n",
znak_przecinek);

if( ispunct(znak_5) )
    printf("Znak '%c' jest znakiem interpunkcyjnym.\n", znak_5);
else
    printf("Znak '%c' nie przeszedł wymaganych testów bo nie jest znakiem interpunkcyjnym.\n",
znak_5);

printf("\n - - - - - isspace(), isblank() - - - - - \n");
printf("isspace(znak) - sprawdza czy 'znak' jest 'białym' znakiem (spacja, enter, tab, \
...)\n");
printf("isblank(znak) - sprawdza czy 'znak' jest spacją lub tabem.\n\n");

if( isspace(znak_spacja) && isblank(znak_spacja) )
    printf("Znak '%c' jest 'białym' znakiem.\n", znak_spacja);
else
    printf("Znak '%c' nie przeszedł wymaganych testów bo nie jest 'białym' znakiem.\n",
znak_spacja);

```

```

if( isspace(znak_5) )
    printf("Znak '%c' jest 'białym' znakiem.\n", znak_5);
else
    printf("Znak '%c' nie przeszedł wymaganych testów bo nie jest 'białym' znakiem.\n",
znak_5);

printf("\n ----- Wymiana wielkich liter na małe i odwrotnie ----- \n");
printf("\n - - - - - - - - - - - - tolower() i toupper() - - - - - - - - - \n");
printf("tolower(s) - zmienia 'znak' na małą literę jeżeli był on wielką literą.\n");
printf("toupper(s) - zmienia 'znak' na wielką literę jeżeli był on małą literą.\n");

printf("\nWymiana wielkich liter na małe. Reszta znaków nie ulegnie zmianie:\n");
for(int i=0; i<strlen(nazwa); i++)
    nazwa_1[i] = tolower(nazwa[i]);
printf("W łańcuchu znakowym '%s' nie ma wielkich liter.", nazwa_1);

printf("\nWymiana małych liter na wielkie. Reszta znaków nie ulegnie zmianie:\n");
for(int i=0; i<strlen(nazwa); i++)
    nazwa_1[i] = toupper(nazwa[i]);
printf("W łańcuchu znakowym '%s' nie ma małych liter.", nazwa_1);

return 0;
}

```

## Wynik działania programu:

```

***** Operacje na pojedynczych znakach *****

----- Test na pojedynczych znakach -----

- - - isalnum(), isalpha(), islower(), isupper(), isprint() - - -
isalnum(znak) - sprawdza czy 'znak' jest alfanumeryczny (A-Z,a-z,0-9).
isalpha(znak) - sprawdza czy 'znak' jest znakiem alfabetu (A-Z,a-z).
islower(znak) - sprawdza czy 'znak' jest małą literą alfabetu.
isupper(znak) - sprawdza czy 'znak' jest wielką literą alfabetu.
isprint(znak) - sprawdza czy 'znak' jest możliwy do wydruku.

Znak 'a' jest alfanumeryczny, alfabetyczny i małą literą, i jest możliwy do wydruku.
Znak 'a' nie przeszedł wymaganych testów bo nie jest wielką literą.

Znak 'A' jest alfanumeryczny, alfabetyczny i wielką literą, i jest możliwy do wydruku.
Znak 'A' nie przeszedł wymaganych testów bo nie jest małą literą.

- - - - - iscntrl() - - - - -
iscntrl(znak) - sprawdza czy 'znak' jest znakiem kontrolnym, np. [Enter].

[Enter] - Badany znak jest znakiem sterującym - ASCII od 0 do 31 włącznie i 127.
Znak '0x0.000000p-1022' nie przeszedł wymaganych testów bo nie jest znakiem sterującym.

- - - - - isdigit() - - - - -
isdigit(znak) - sprawdza czy 'znak' jest znakiem numerycznym (cyfrą, 0-9).

Znak '5' jest cyfrą.
Znak 'a' nie przeszedł wymaganych testów bo nie jest cyfrą.

Sprawdzenie czy łańcuch znakowy da się zamienić na liczbę naturalną:
Łańcuch znakowy '12345' dał się zamienić na liczbę naturalną.

- - - - - isxdigit() - - - - -
isxdigit(znak) - sprawdza czy 'znak' może być cyfrą zapisu szesnastkowego.

Znak 'f' jest cyfrą systemu szesnastkowego.
Znak ',' nie przeszedł wymaganych testów bo nie jest cyfrą systemu szesnastkowego.

```

```

- - - - - isgraph() - - - - -
isgraph(znak) - sprawdza czy 'znak' jest znakiem graficznym.

Znak 'x' jest znakiem graficznym.
[Enter] - Badany znak nie przeszedł wymaganych testów bo nie jest znakiem graficznym.

- - - - - ispunct() - - - - -
ispunct(znak) - sprawdza czy 'znak' jest znakiem interpunkcyjnym.

Znak ',' jest znakiem interpunkcyjnym.
Znak '5' nie przeszedł wymaganych testów bo nie jest znakiem interpunkcyjnym.

- - - - - isspace(), isblank() - - - - -
isspace(znak) - sprawdza czy 'znak' jest 'białym' znakiem (spacja, enter, tab, ...)
isblank(znak) - sprawdza czy 'znak' jest spacją lub tabem.

Znak ' ' jest 'białym' znakiem.
Znak '5' nie przeszedł wymaganych testów bo nie jest 'białym' znakiem.

----- Wymiana wielkich liter na małe i odwrotnie -----

- - - - - tolower() i toupper() - - - - -
tolower(s) - zmienia 'znak' na małą literę jeżeli był on wielką literą.
toupper(s) - zmienia 'znak' na wielką literę jeżeli był on małą literą.

Wymiana wielkich liter na małe. Reszta znaków nie ulegnie zmianie:
W łańcuchu znakowym 'james bond, agent 007/j23' nie ma wielkich liter.

Wymiana małych liter na wielkie. Reszta znaków nie ulegnie zmianie:
W łańcuchu znakowym 'JAMES BOND, AGENT 007/J23' nie ma małych liter.
-----
Process exited after 0.5418 seconds with return value 0
Press any key to continue . . . _

```