

# Header string.h

Header	Opis zawartych funkcji	Przykładowe funkcje
string.h	Operacje na łańcuchach znakowych	strlen(), strcat(), strncat(), strcpy(), strncpy(), strcmp(), strncmp(), strcoll(), strchr(), strrchr(), strspn(), strcspn(), strpbrk(), strstr(), strlwr(), strupr(), strrev()

## #include<string.h>

### Różne operacje na łańcuchach znakowych

- **strlen(s)** - Podaje długość (**length**) łańcucha znaków zmiennej 's' z **wyłączeniem znaku NULL**.
- **strcat(s,t)** - Łączy dwa ciągi (**concatenation**) - dopisuje łańcuch znaków zmiennej 't' na koniec łańcucha znaków zmiennej 's'.
- **strncat(s,t,n)** - Dopisuje 'n' pierwszych znaków łańcucha znaków zmiennej 't' na koniec łańcucha znaków zmiennej 's'.
- **strcpy(s,t)** - Kopiuje (**copy**) łańcuch znaków zmiennej 't' do łańcucha znaków zmiennej 's' powielając poprzednią zawartość łańcucha znaków 's'.
- **strncpy(s,t,n)** - Kopiuje 'n' pierwszych znaków łańcucha znaków zmiennej 't' do łańcucha znaków zmiennej 's' powielając poprzednią zawartość pierwszych 'n' znaków łańcucha 's';
- **strcmp(s,t)** - Porównuje (**compare**) łańcuchy znakowe 's' i 't'. Język 'C' porównuje ich numery ASCII i ustawia je w kolejności jak hasła w encyklopedii:
  - jeżeli  $s < t$  to *strcmp* będzie ujemne;
  - jeżeli  $s == t$  to *strcmp* będzie równe zero;
  - jeżeli  $s > t$  to *strcmp* będzie dodatnie;
- **strncmp(s,t,n)** - Robi to samo co *strcmp*, ale tylko dla pierwszych 'n' znaków;
- **strcoll(s,t)** - Porównuje dwa ciągi znaków zgodnie z bieżącymi ustawieniami regionalnymi.
- **strchr(s,znak)** - Podaje pointer do pierwszego (może jedyne) pojedynczego znaku 'znak' w łańcuchu znaków 's', a więc umożliwia znalezienie pozycji tego znaku w łańcuchu znakowym a tym samym odczyt łańcucha znaków od tego znaku do końca łańcucha znaków 's'.  
Jeżeli go nie znajdzie, to podaje 'NULL'.
- **strrchr(s,znak)** - Podaje pointer do ostatniego (może jedyne) pojedynczego znaku 'znak' w łańcuchu znaków 's', a więc umożliwia znalezienie pozycji tego znaku w łańcuchu znakowym a tym samym odczyt łańcucha znaków od tego znaku do końca łańcucha znaków 's'.  
Jeżeli go nie znajdzie, to podaje 'NULL'.
- **strspn(s,t)** - Podaje długość podciągu w łańcuchu znaków 's', który składa się tylko ze znaków znalezionych w łańcuchu znaków 't'
- **strcspn(s,t)** - Podaje długość podciągu w łańcuchu znaków 's', który składa się tylko ze znaków **nie** znalezionych w łańcuchu znaków 't'
- **strpbrk(s,t)** - Podaje pointer do pierwszego (może jedyne) pojedynczego znaku w łańcuchu znaków 's', który jest jednym ze znaków z łańcucha 't'. Nie obejmuje to znaku NULL. Pamiętaj: Pierwszy znak jest na pozycji 0.  
Jeżeli żadnego znaku z 't' nie znajdzie w 's', to podaje 'NULL'.
- **strstr(s,t)** - Podaje, na której pozycji w łańcuchu znaków 's' zaczyna się występowanie po raz pierwszy (może jedyny) łańcucha znaków 't'. Pamiętaj: Pierwszy znak jest na pozycji 0.
- **strlwr(s)** - Zamienia litery wielkie na małe
- **strupr(s)** - Zamienia litery małe na wielkie
- **strrev(s)** - Odwraca ciąg znaków

```
// Operacje na łańcuchach znakowych

#include<stdio.h>
#include<string.h>           // ten temat
#include<locale.h>           // dla 'setlocale()'

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

    char s[30] = "abra";
    char t[30] = "cadabra";

    const char *u = "abracadabra";
    char znak1 = 'r';
    char *ptr_strchr1 = strchr(u, znak1);
    char *ptr_strrchr1 = strrchr(u, znak1);
    char znak2 = 'Z';
    char *ptr_strchr2 = strchr(u, znak2);
    char *ptr_strrchr2 = strrchr(u, znak2);

    const char *v = "Alex";
    const char *w = "Edwin";
    char *ptr_strpbrk1 = strpbrk(u, v);
    char *ptr_strpbrk2 = strpbrk(u, w);

    char *ptr_strstr1 = strstr(t, v);
    char *ptr_strstr2 = strstr(t, s);

    char z[40] = "Sezame Otwórz Się. To ja, J-23/007";
    char zz[30] = "roma";

    // Zmienne pomocnicze trzymające niezmienną wartość łańcucha znaków 's'
    char a[30] = "abra";
    char b[30] = "abra";
    char c[30] = "abra";
    char d[30] = "abra";
    char e[30] = "abra";

    printf("\n - - - - - Operacje na łańcuchach znakowych - - - - - \n\n");

    printf("\n Pierwszy łańcuch znaków 's', 'a', 'b', 'c', 'd' i 'e' : %s", s);
    printf("\n Drugi łańcuch znaków 't' : %s", t);
    printf("\n\n\n");

    printf("strlen(s) - Podaje długość (length) łańcucha znaków zmiennej 's' z \
wyłączeniem znaku NULL.\n");
    printf("Wynik strlen(s): Długość łańcucha znaków 'a' = %d \n\n", strlen(s) );

    printf("strcat(s,t) - łączy (concatenation) dwa ciągi: dopisuje łańcuch znaków \
zmiennej 't' na koniec łańcucha znaków zmiennej 's'. \n");
    printf("Wynik strcat(a, t): Teraz wartość łańcucha znaków 'a' = %s \n\n", strcat(a,
t));

    printf("strncat(s,t,n) - Dopisuje 'n' pierwszych znaków łańcucha znaków zmiennej \
't' na koniec łańcucha znaków zmiennej 's'. \n");
}
```

```

printf("Wynik strncat(b, t, 4): Teraz wartość łańcucha znaków 'b' = %s \n\n",
strncat(b, t, 4));

printf("strcpy(s,t) - Kopiuje (copy) łańcuch znaków zmiennej 't' do łańcucha \
znaków zmiennej 's' powielając poprzednią zawartość łańcucha znaków 's'. \n");
printf("Wynik strcpy(c, t): Teraz wartość łańcucha znaków 'c' = %s \n\n", strcpy(c,
t));

printf("strncpy(s,t,n) - Kopiuje 'n' pierwszych znaków łańcucha znaków zmiennej \
't' do łańcucha znaków zmiennej 's' powielając poprzednią zawartość pierwszych 'n' \
znaków łańcucha 's'. \n");
printf("Wynik strncpy(e, t, 2): Teraz wartość łańcucha znaków 'e' = %s \n\n",
strncpy(e,t,2));

printf("strcmp(s,t) - Porównuje (compare) łańcuchy znakowe 's' i 't'. Język 'C' \
porównuje ich numery ASCII i ustawia je w kolejności jak hasła w encyklopedii: \n");
printf("      - jeżeli s < t to strcmp będzie ujemne, \n");
printf("      - jeżeli s == t to strcmp będzie równe zeru, \n");
printf("      - jeżeli s > t to strcmp będzie dodatnie. \n");
printf("Wynik strcmp(s, t) = %d \n", strcmp(s, t) );
printf("Poniżej jest dowód na to, że w 30-o bajtowym łańcuchu 's' zaraz po \
\"abra\" wstawiany jest automatycznie znak zamknięcia łańcucha znakowego (NULL). \n");
printf("Wynik strcmp(s, \"abra\") = %d \n", strcmp(s, "abra") );
printf("Wynik strcmp(t, s) = %d \n\n", strcmp(t, s) );

printf("strncmp(s,t,n) - Robi to samo co strcmp, ale tylko dla pierwszych 'n' \
znaków. \n");
printf("Wynik strncmp(\"Al Pacino\", \"Alibaba\", 3) = %d \n", strncmp("Al Pacino",
"Alibaba", 3) );
printf("Wynik strncmp(\"Alicja\", \"Alibaba\", 3) = %d \n", strncmp("Ali",
"Alibaba", 3) );
printf("Wynik strncmp(\"Alibaba\", \"Al Pacino\", 3) = %d \n\n", strncmp("Alibaba",
"Al Pacino", 3) );

printf("strchr(s,znak) - Podaje pointer do pierwszego (może jedyne) pojedynczego \
znaku 'znak' w łańcuchu znaków 's', a więc umożliwia znalezienie pozycji tego znaku \
w łańcuchu znakowym a tym samym odczyt łańcucha znaków od tego znaku do końca \
łańcucha znaków 's'. \n");
printf("      Jeżeli go nie znajdzie, to podaje 'NULL'. \n");

if (ptr_strchr1 == NULL)
    printf("Łańcuch znaków \"%s\" nie zawiera znaku '%c'. \n", a, znak1);
else
    printf("Pierwsze wystąpienie znaku '%c' w łańcuchu znaków \"%s\" jest na \
pozycji %d. \n", znak1, a, ptr_strchr1 - u );

if (ptr_strchr2 == NULL)
    printf("Łańcuch znaków \"%s\" nie zawiera znaku '%c'. \n", a, znak2);
else
    printf("Pierwsze wystąpienie znaku '%c' w łańcuchu znaków \"%s\" jest na \
pozycji %d. \n", znak2, a, ptr_strchr2 - u );

printf("\n");

printf("strrchr(s,znak) - Podaje pointer do ostatniego (może jedyne) \
pojedynczego znaku 'znak' w łańcuchu znaków 's', a więc umożliwia znalezienie \
pozycji tego znaku w łańcuchu znakowym a tym samym odczyt łańcucha znaków od tego \
znaku do końca łańcucha znaków 's'. \n");
printf("      Jeżeli go nie znajdzie, to podaje 'NULL'.\n");

```

```

if (ptr_strchr1 == NULL)
    printf("łańcuch znaków \"%s\" nie zawiera znaku '%c'. \n", a, znak1);
else
    printf("Ostatnie wystąpienie znaku '%c' w łańcuchu znaków \"%s\" jest na pozycji \
%d. \n", znak1, a, ptr_strchr1 - u );

if (ptr_strchr2 == NULL)
    printf("łańcuch znaków \"%s\" nie zawiera znaku '%c'. \n", a, znak2);
else
    printf("Ostatnie wystąpienie znaku '%c' w łańcuchu znaków \"%s\" jest na pozycji \
%d. \n", znak2, a, ptr_strchr2 - u );

printf("\n");

printf("strspn(s,t) - Podaje długość podciągu od początku łańcucha znaków 's', \
który jest kopią całości lub części łańcucha znaków 't'. Przykład \"Pacino \" da \
liczbę 7 \n");
printf("Wynik strspn(\"Pacino jest aktorem\", \"Pamiętam Al Pacino z wielu \
filmów\") = %d \n\n", strspn("Pacino jest aktorem", "Pamiętam Al Pacino z wielu \
filmów") );

printf("strcspn(s,t) - Podaje długość podciągu od początku (pierwszy znak ma nr 0) \
łańcucha znaków 's', który składa się tylko ze znaków nie będących żadnym znakiem w \
łańcuchu znaków 't'. \n");
printf("Wynik strcspn(\"Pacino ma na imię Al\", \"xyAz\") = %d \n",
    strcspn("Pacino ma na imię Al", "xyAz") );
printf("Wynik strcspn(\"Pacino ma na imię Al\", \"Alex\") = %d \n",
    strcspn("Pacino ma na imię Al", "Alex") );
printf("Wynik strcspn(\"Pacino ma na imię Al\", \"lex\") = %d \n",
    strcspn("Pacino ma na imię Al", "lex") );
printf("Wynik strcspn(\"Pacino ma na imię Al\", \"ex\") = %d \n\n",
    strcspn("Pacino ma na imię Al", "ex") );

printf("strpbrk(s,t) - Podaje pointer do pierwszego (może jedyne) pojedynczego \
znaku w łańcuchu znaków 's', który jest jednym ze znaków z łańcucha 't'. \
Nie obejmuje to znaku NULL. Pamiętaj: Pierwszy znak jest na pozycji 0. \n");
printf("    Jeżeli żadnego znaku z 't' nie znajdzie w 's', to podaje 'NULL'. \n");

if (ptr_strpbrk1 == NULL)
    printf("łańcuch znaków \"%s\" nie zawiera żadnego ze znaków \"%s\". \n", u, v);
else
    printf("Pierwsze wystąpienie znaku w \"%s\", który jest jednym ze znaków \
łańcucha \"%s\" to '%c' i jest na pozycji %d. \n", u, v, *strpbrk(u, v), strpbrk(u, v)
- u );

if (ptr_strpbrk2 == NULL)
    printf("łańcuch znaków \"%s\" nie zawiera żadnego ze znaków \"%s\". \n", u, w);
else
    printf("Pierwsze wystąpienie znaku w \"%s\", który jest jednym ze znaków \
łańcucha \"%s\" to '%c' i jest na pozycji %d. \n", u, w, *strpbrk(u, w), strpbrk(u, w)
- u );

printf("\n");

printf("strstr(s,t) - Podaje, na której pozycji w łańcuchu znaków 's' zaczyna się \
występowanie po raz pierwszy (może jedyne) łańcucha znaków 't'. Pamiętaj: Pierwszy \
znak jest na pozycji 0. \n");

```

```

if (ptr_strstr1 == NULL)
    printf("łańcuch znaków \"%s\" nie zawiera łańcucha znaków \"%s\". \n", t, v);
else
    printf("Pierwsze wystąpienie łańcucha znaków \"%s\" w łańcuchu znaków \"%s\" \
jest na pozycji %d. \n", v, t, strstr(t, v) - t );

if (ptr_strstr2 == NULL)
    printf("łańcuch znaków \"%s\" nie zawiera łańcucha znaków \"%s\". \n", t, s);
else
    printf("Pierwsze wystąpienie łańcucha znaków \"%s\" w łańcuchu znaków \"%s\" \
jest na pozycji %d. \n", s, t, strstr(t, s) - t );

printf("\n");

printf("strlwr(s) - Zamienia litery wielkie na małe. \n");
printf("Wynik zamiany wielkich liter na małe w łańcuchu znaków \"%s\" --> ", z);
printf("%s \n\n", strlwr(z) );

printf("strupr() - Zamienia litery małe na wielkie. \n");
printf("Wynik zamiany małych liter na wielkie w łańcuchu znaków \"%s\" --> ", z);
printf("%s \n\n", strupr(z) );

printf("strrev() - Odwraca ciąg znaków. \n");
printf("Wynik odwrócenia łańcucha znaków \"%s\" --> ", zz);
printf("%s \n\n", strrev(zz) );

return 0;
}

```

## Wynik działania programu:

```

- - - - - Operacje na łańcuchach znakowych - - - - -

Pierwszy łańcuch znaków 's', 'a', 'b', 'c', 'd' i 'e' : abra
Drugi łańcuch znaków 't' : cadabra

strlen(s) - Podaje długość (length) łańcucha znaków zmiennej 's' z wyłączeniem znaku NULL.
Wynik strlen(s): Długość łańcucha znaków 'a' = 4

strcat(s,t) - Łączy (concatenation) dwa ciągi: dopisuje łańcuch znaków zmiennej 't' na koniec łańcucha znaków zmiennej 's'.
Wynik strcat(a, t): Teraz wartość łańcucha znaków 'a' = abracadabra

strncat(s,t,n) - Dopisuje 'n' pierwszych znaków łańcucha znaków zmiennej 't' na koniec łańcucha znaków zmiennej 's'.
Wynik strncat(b, t, 4): Teraz wartość łańcucha znaków 'b' = abracada

strcpy(s,t) - Kopiuje (copy) łańcuch znaków zmiennej 't' do łańcucha znaków zmiennej 's' powielając poprzednią zawartość łańcucha znaków 's'.
Wynik strcpy(c, t): Teraz wartość łańcucha znaków 'c' = cadabra

strncpy(s,t,n) - Kopiuje 'n' pierwszych znaków łańcucha znaków zmiennej 't' do łańcucha znaków zmiennej 's' powielając poprzednią zawartość pierwszych 'n' znaków łańcucha 's'.
Wynik strncpy(e, t, 2): Teraz wartość łańcucha znaków 'e' = cara

strcmp(s,t) - Porównuje (compare) łańcuchy znakowe 's' i 't'. Język 'C' porównuje ich numery ASCII i ustawia je w kolejności jak hasła w encyklopedii:
    - jeżeli s < t to strcmp będzie ujemne,
    - jeżeli s == t to strcmp będzie równe zero,
    - jeżeli s > t to strcmp będzie dodatnie.
Wynik strcmp(s, t) = -1
Poniżej jest dowód na to, że w 30-o bajtowym łańcuchu 's' zaraz po "abra" wstawiany jest automatycznie znak zamknięcia łańcucha znakowego (NULL).
Wynik strcmp(s, "abra") = 0

```

```

Wynik strcmp(t, s) = 1

strncmp(s,t,n) - Robi to samo co strcmp, ale tylko dla pierwszych 'n' znaków.
Wynik strncmp("Al Pacino", "Alibaba", 3) = -1
Wynik strncmp("Alicja", "Alibaba", 3) = 0
Wynik strncmp("Alibaba", "Al Pacino", 3) = 1

strchr(s,znak) - Podaje pointer do pierwszego (może jedyne) pojedynczego znaku 'znak' w łańcuchu znaków 's', a więc
umożliwia znalezienie pozycji tego znaku w łańcuchu znakowym a tym samym odczyt łańcucha znaków od tego znaku do końca
łańcucha znaków 's'.
    Jeżeli go nie znajdzie, to podaje 'NULL'.
Pierwsze wystąpienie znaku 'r' w łańcuchu znaków "abracadabra" jest na pozycji 2.
łańcuch znaków "abracadabra" nie zawiera znaku 'Z'.

strrchr(s,znak) - Podaje pointer do ostatniego (może jedyne) pojedynczego znaku 'znak' w łańcuchu znaków 's', a więc
umożliwia znalezienie pozycji tego znaku w łańcuchu znakowym a tym samym odczyt łańcucha znaków od tego znaku do końca
łańcucha znaków 's'.
    Jeżeli go nie znajdzie, to podaje 'NULL'.
Ostatnie wystąpienie znaku 'r' w łańcuchu znaków "abracadabra" jest na pozycji 9.
łańcuch znaków "abracadabra" nie zawiera znaku 'Z'.

strspn(s,t) - Podaje długość podciągu od początku łańcucha znaków 's', który jest kopią całości lub części łańcucha
znaków 't'. Przykład "Pacino " da liczbę 7
Wynik strspn("Pacino jest aktorem", "Pamiętam Al Pacino z wielu filmów") = 7

strcspn(s,t) - Podaje długość podciągu od początku (pierwszy znak ma nr 0) łańcucha znaków 's', który składa się tylko
ze znaków nie będących żadnym znakiem w łańcuchu znaków 't'.
Wynik strcspn("Pacino ma na imię Al", "xyAz") = 18
Wynik strcspn("Pacino ma na imię Al", "Alex") = 18
Wynik strcspn("Pacino ma na imię Al", "lex") = 19
Wynik strcspn("Pacino ma na imię Al", "ex") = 20

strpbrk(s,t) - Podaje pointer do pierwszego (może jedyne) pojedynczego znaku w łańcuchu znaków 's', który jest
jednym ze znaków z łańcucha 't'. Nie obejmuje to znaku NULL. Pamiętaj: Pierwszy znak jest na pozycji 0.
    Jeżeli żadnego znaku z 't' nie znajdzie w 's', to podaje 'NULL'.
łańcuch znaków "abracadabra" nie zawiera żadnego ze znaków "Alex".
Pierwsze wystąpienie znaku w "abracadabra", który jest jednym ze znaków łańcucha "Edwin" to 'd' i jest na pozycji 6.

strstr(s,t) - Podaje, na której pozycji w łańcuchu znaków 's' zaczyna się występowanie po raz pierwszy (może jedyny)
łańcucha znaków 't'. Pamiętaj: Pierwszy znak jest na pozycji 0.
łańcuch znaków "cadabra" nie zawiera łańcucha znaków "Alex".
Pierwsze wystąpienie łańcucha znaków "abra" w łańcuchu znaków "cadabra" jest na pozycji 3.

strlwr(s) - Zamienia litery wielkie na małe.
Wynik zamiany wielkich liter na małe w łańcuchu znaków "Sezame Otwórz Się. To ja, J-23/007" --> sezame otwórz się.
to ja, j-23/007

strupr() - Zamienia litery małe na wielkie.
Wynik zamiany małych liter na wielkie w łańcuchu znaków "Sezame Otwórz Się. To ja, J-23/007" --> SEZAMIE OTWÓRZ SIĘ.
TO JA, J-23/007

strrev() - Odwraca ciąg znaków.
Wynik odwrócenia łańcucha znaków "roma" --> amor

-----
Process exited after 13.94 seconds with return value 0
Press any key to continue . . .

```