

## T: Łańcuchy w C++.

1. **String** - to uogólniona klasa łańcuchów/napisów - typ obiektowy.

<https://en.cppreference.com/w/cpp/header/cstring>

<https://en.cppreference.com/w/cpp/header/string>

2. Deklaracja lub inicjalizacja zmiennej typu string

```
string s;  
s="telefon"; // wynik s="telefon"  
string s("telefon"); // wynik s="telefon"  
string s="telefon"; // wynik s="telefon"  
string s(8, '*'); // wynik s="*****"  
string w=s; // wynik w="*****"
```

3. **Operator +** łączy dwa łańcuchy tworząc nowy łańcuch,

```
string a="jeden", b="dwa";  
cout<<a+b<<endl;  
cout<<b+a<<endl;
```

Ważne ! operator + nie może łączyć literałów

4. Deklaracja lub inicjalizacja zmiennej typu string

```
string napis;  
napis="Ala"+"Ola"; //blad !!!  
napis=napis+"Jola";  
napis="Ewa"+napis;  
cout<<napis<<endl;
```

5. **Operator +=** operator łączy dwa łańcuchy tworząc nowy łańcuch,

```
string a="jeden", b="dwa";  
a+=b;  
cout<<a<<endl;
```

6. **Operator indeksowy []** - dostęp do poszczególnych znaków jak w tablicy

```
cout<<a[2]<<endl;
```

7. **Wczytywanie całego łańcucha**

```
string wczytaj;  
cin>>wczytaj;  
cout<<wczytaj<<endl;
```

Ważne ! jeśli wpisujemy więcej wyrazów rozdzielonych spacją + enter => wczyta nam tylko pierwszy wyraz - pozostałe zignoruje

>> - operator pomija wszystkie białe znaki i odczytuje liczby, łańcuchy

**Białymi znakami** są nazywane te symbole, które są używane w tekście i nie posiadają swojej reprezentacji graficznej np.: **spacja**, **tabulacja** czy też **znak przejścia do nowej linii (enter)**.

```
#include <cctype>
```

```
int isspace( int ch );
```

Funkcja `isspace` zwraca wartość różną od zera gdy argument, który został przekazany do funkcji jest białym znakiem. W przeciwnym wypadku funkcja zwraca wartość zero.

Wykaz białych znaków

kod ASCII (hex) opis		zapis formalny
0x09	tabulacja pozioma	'\t'
0x0A	przejście karetki do nowego wiersza	'\n'
0x0B	tabulacja pionowa	'\v'

0x0C	przesuwa karetkę na początek nowej strony '\f'
0x0D	powrót karetki na początek wiersza '\r'
0x20	spacja ' '

```
#include <cctype>
int isalnum( int ch );
```

Funkcja **isalnum** zwraca wartość różną od zera gdy argument, który został przekazany do funkcji jest **znakiem alfanumerycznym**. W przeciwnym wypadku funkcja zwraca wartość zero.

#### nazwa argumentu opis

ch	Kod znaku ASCII
----	-----------------

```
#include <cctype>
int isalpha( int ch );
```

Funkcja **isalpha** zwraca wartość różną od zera gdy argument, który został przekazany do funkcji jest **literą alfabetu**. W przeciwnym wypadku funkcja zwraca wartość zero.

#### nazwa argumentu opis

ch	Kod znaku ASCII
----	-----------------

```
#include <cctype>
int isdigit( int ch );
```

Funkcja **isdigit** zwraca wartość różną od zera gdy argument, który został przekazany do funkcji jest **cyfrą**. W przeciwnym wypadku funkcja zwraca wartość zero.

#### nazwa argumentu opis

ch	Kod znaku ASCII
----	-----------------

8. **get()** - pobiera następny znak ze strumienia

```
char a;
a=cin.get();
lub
cin.get(a);
wczytanie z klawiatury wartości zmiennej a, równoważne cin>>a;
```

```
cin.get();
wymuszenie naciśnięcia klawisza ENTER
```

```
char s[50];
cin.get(s, sizeof(s), 'a')
wczytanie z klawiatury do zmiennej s znaków, aż do pojawienia się znaku 'a', odczytania 49 znaków lub odczytania wszystkich znaków
```

```
cin.get(s, 24)
wczytanie z klawiatury do zmiennej s znaków, aż do pojawienia się znaku specjalnego '\n' (domyślnie), odczytania 23 znaków lub odczytania wszystkich znaków
```

9. `getline(char * tekst, int dlg, char k='\n')` - odczytuje znaki ze strumienia (również spacje) do zmiennej `tekst` do momentu osiągnięcia znaku `k` (znak ten jest pobierany ze strumienia, ale nie jest dopisywany do zmiennej `tekst`), odczytania `dlg-1` znaków lub odczytania wszystkich znaków

```
string wczytaj;  
getline(cin, wczytaj);  
cout<<wczytaj<<endl;
```

```
char s[50];
```

```
cin.getline(s, 40, 'c');
```

wczytanie z klawiatury do zmiennej `s` znaków, aż do pojawienia się znaku `'c'` lub odczytania 39 znaków

```
cin.getline(s, sizeof(s));
```

wczytanie z klawiatury do zmiennej `s` **wszystkich znaków** (maksymalna długość łańcucha wynosi 256 znaków)

```
cin.getline(s, 50)
```

wczytanie z klawiatury do zmiennej `s` **znaków, aż do odczytania 49 znaków**

```
string s1;
```

```
getline(cin, s1);
```

wczytanie z klawiatury do zmiennej `s1` **wszystkich znaków** (maksymalna długość łańcucha wynosi 256 znaków)

#### 10. Porównanie łańcuchów:

`>`, `<`, `>=`, `<=`, `!=` - operatory porównują wg porządku alfabetycznego - np.: `'a'>'A'`, `'z'>'a'`

`==` - operator porównuje czy łańcuchy są takie same

```
string haslo = "admin";
```

```
string password;
```

```
cout << "Podaj haslo:" << endl;
```

```
getline(cin, password);
```

```
if(haslo == password)
```

```
cout << "OK" << endl;
```

```
else
```

```
cout << "bledne haslo";
```

```
//=====
```

```
string s1, s2("Ala"), s3="Ola";
```

```
s1="Ala";
```

```
if (s1==s2) cout<<"teksty "<<s1<<" i "<<s2<<" sa takie  
same"<<endl;
```

```
else if (s1<s2) cout<<"tekst "<<s1<<" jest wczesniej w slowniku  
niz tekst "<<s2<<endl;
```

```
else cout<<"tekst "<<s1<<" nie jest pozniej w slowniku niz tekst  
"<<s2<<endl;
```

```
s2+='n';
```

```
if (s1==s2) cout<<"teksty "<<s1<<" i "<<s2<<" sa takie  
same"<<endl;
```

```
else if (s1<s2) cout<<"tekst "<<s1<<" jest wczesniej w slowniku  
niz tekst "<<s2<<endl;
```

```
else cout<<"tekst "<<s1<<" jest pozniej w slowniku niz tekst  
"<<s2<<endl;
```

```
s3+="!!!";
```

```
cout<<s3<<endl;
```

```
s3[1]='*';  
cout<<s3<<endl;
```

11. Przydatne funkcje:

```
string s = "abedef";
```

- **bool empty(void)** - zwraca wartość **true**, jeśli napis jest pusty, w przeciwnym razie **false**  
`bool a = s.empty(); // a=0`
- **int size(void)** - zwraca liczbę znaków w napisie  
`int a = s.size(); // a=6`
- **char at(int i)** - zwraca znak o podanym indeksie **i**, zapobiega wyjściu poza przedział  
`char a = s.at(3); // a='d'`
- **void clear(void)** - usuwa wszystkie znaki z łańcucha  
`s.clear(); // s=""`
- **char\* c\_str(void)** - konwertuje typ **string** na **char[]**  
`char a[s.size()+1];  
strcpy(a, s.c_str()); // a="abcdef"`
- **string substr(int i, int j)** - zwraca podciąg znaków napisu zaczynający się na pozycji **i**, o długości **j** znaków  
`string a=s.substr(2,3); // a="cde"`
- **int find(string b)** - wyszukuje w napisie podciąg znaków **b**, zwraca pozycję rozpoczęcia szukanego podciągu w tekście lub wartość większą od długości przeszukiwanego napisu  
`string b="cde";  
int a=s.find(b); // a=2`

12. Konwersje łańcuchów:

(1) konwersji typu **string** na **char[]**:

```
string s1 = "abcdef";  
char s2[50];  
strcpy(s2, s1.c_str());
```

(2) konwersji **char[ ]** na **string**:

```
string s1;  
char s2[] = "abcdef";  
s1 = s2;
```

**W DOMU** (czas 2 tygodnie)

**Zad.1** Napisz program wyznaczający długość wprowadzonego z klawiatury imienia i nazwiska.

**Zad.2** Napisz program wypisujący wprowadzony z klawiatury tekst jako rozstrzelony w następujący sposób:

a) 1 znak + 1 spacja,

b) 1 znak + 1 spacja (tekst czytany od końca),

**Zad.3** Napisz program wyznaczający liczbę podanych poniżej znaków w tekście zadeklarowanym w programie (zastosuj inicjalizację łańcucha):

a) liczba znaków 'a',

b) liczba znaków różnych od 'B', które mają numer parzysty.

**Zad.4** Napisz program wypisujący określone znaki tekstu wprowadzonego z klawiatury:

a) wypisywanie znaków różnych od 'h',

b) wypisywanie znaków równych 'a' lub 'A' lub 'o' lub 'O'.

**Zad.5** Napisz program wczytujący z klawiatury tekst ze spacjami i wypisujący go w poniższy sposób:

a) wszystkie wyrazy wypisane w kolumnie,

b) wszystkie wyrazy, których pierwszy znak jest równy ostatniemu, wypisane w kolumnie.

**Zad.6** Napisz program wczytujący dany tekst (zastosuj inicjalizację łańcucha) i wypisujący z niego wszystkie wyrazy o nieparzystej liczbie liter zaczynające się na literę 'p'.