

# Przetwarzanie tekstów

# Definiowanie tekstów

```
char *Litery = "ABC";
```



A	B	C	
0x41	0x42	0x43	0x00

```
char *Napis = "Waniki."; // tekst stały
```

```
*(Napis + 1) = 'y'; // błąd
```

```
char Tekst[16] = { "Pomołka." };
```

```
// kopiowanie tekstu stałego do tablicy
```

```
Tekst[3] = 'y'; // poprawnie
```

```
*(Tekst + 3) = 'y'; // poprawnie
```

# Przetwarzanie tekstów

```
char Napis [ 16 ] = "Kopytko.";
```

```
int i = 0;
```

```
while( Napis[i] != 0 )  
{  
    if ( Napis[i] == 'o')  
    {  
        Napis[i] = 'a';  
    }  
  
    ++i;  
}
```

```
char Napis [ 16 ] = "Kopytko.";
```

```
char *ptr = Napis;
```

```
while ( *ptr != 0 )  
{  
    if ( *ptr == 'o' )  
    {  
        *ptr = 'a';  
    }  
  
    ++ptr;  
}
```

# Biblioteka `string.h`

```
char* strcat( char* Destination,  
              const char* Source );
```

```
// Dodaje (konkatenuje) łańcuchy Source  
// i Destination. Wynik: wskaźnik  
// łańcucha, do którego dołączany jest tekst
```

```
char* strncat( char* Dest,  
              const char* Source,  
              size_t Count );
```

```
// Analogicznie jak strcat lecz kopiuje  
// nie więcej niż Count znaków
```

```
char* strchr( const char* String, int C );
```

```
// Wyszukuje pierwsze wystąpienie znaku C  
// w łańcuchu String. Wynik: wskaźnik  
// znalezionego znaku bądź NULL  
// jeśli nie znaleziono
```

```
char* strrchr( const char* String, int C );
```

```
// Analogicznie jak strchr lecz od końca
```

```
char* strpbrk( const char* String,  
               const char* CharSet );
```

```
// Wyszukuje pierwsze wystąpienie jednego  
// ze znaków z łańcucha CharSet  
// w łańcuchu String
```

```
size_t strlen( const char* String );
```

```
// Oblicza długość łańcucha String
```



```
int strcmp( const char* String1,  
            const char* String2 );
```

```
// Porównanie leksykograficzne dwóch  
// łańcuchów. Wynik funkcji:  
// < 0 - String1 mniejszy niż String2  
// = 0 - String1 i String2 identyczne  
// > 0 - String1 większy niż String2
```

```
int strncmp( const char* String1,  
             const char* String2,  
             size_t Count );
```

```
// Analogicznie jak strcmp lecz nie więcej  
// niż Count znaków
```

```
char* strcpy( char* Destination,  
              const char* Source );
```

```
// Kopiuje łańcuch Source do Destination  
// usuwając poprzednią wartość Destination.  
// Wynik: wskaźnik łańcucha, do którego  
// kopiowany jest tekst
```

```
char* strncpy( char* Dest,  
               const char* Source,  
               size_t Count );
```

```
// Analogicznie jak strcpy lecz nie więcej  
// jak Count znaków
```

```
char* strstr( char* str1, const char *str2 );
```

```
// Zwraca wskaźnik do pierwszego  
// wystąpienia str2 w str1,  
// lub wskaźnik zerowy jeśli str2  
// nie jest częścią str1.  
// Proces porównywania nie obejmuje  
// znaku końca ciągu znakowego.  
// Jeśli taki wystąpi proces jest przerywany.
```

Więcej funkcji biblioteki `string.h` można znaleźć tutaj:

<http://www.cplusplus.com/reference/cstring/>