

Przetwarzanie danych tekstowych

Definicje i inicjalizacje zmiennych tekstowych – przykłady

```
char tekst[ ] = {'J', 'e', 's', 't', 'e', 'm', ' ', 't', 'e', 'k', 's', 't', 'e', 'm', '\0'};
char tekst2[ ] = {"Jestem tekstem"};
char tekst3[ ] = "Jestem tekstem";
char tekst4[100] = "Jestem tekstem";
```

Przykładowe operacje na pojedynczych literach tekstu

```
char tekst[ ] = {'J', 'e', 's', 't', 'e', 'm', ' ', 't', 'e', 'k', 's', 't', 'e', 'm', '\0'};
printf("%s\n", tekst);
tekst[1] = 'E'; // zamiana drugiej litery na dużą
printf("%s\n", tekst);
tekst[2] = toupper(tekst[2]); // zamiana trzeciej litery na dużą (biblioteka: ctype.h)
printf("%s\n", tekst);
for( int i=3; i<10; i++)
    tekst[ i ] = toupper(tekst[i]); // zamiana kolejnych siedmiu liter na duże
printf("%s\n", tekst);
tekst[6] = '\0'; // skrócenie tekstu do 6 liter
printf("%s\n", tekst);
for( int i=0; i<3; i++)
    tekst[i] = tolower(tekst[i]); // zamiana 3 początkowych liter na małe
printf("%s\n", tekst);
```

Przykłady przetwarzania tekstu

```
// poszukiwanie ostatniej kropki w łańcuchu
char txt[] = "Ala ma kota. Kot Ali to Mruczek.";
int i, poz_kropki = -1;
for(i=0; txt[i] != '\0'; i++)
    if( txt[i] == '.')
        poz_kropki=i;
printf("pozycja ostatniej kropki: %d", i);

char string[1024];
printf("Podaj tekst do analizy:\n");
gets(string); // pobranie tekstu z konsoli
// poszukiwanie określonych znaków
int i;
for(i=0; string[i] != '\0'; i++)
    if(string[i] == 'A' || string[i] == 'a')
        printf("Litera %c na pozycji: %d\n", string[i], i);
```

Wybrane funkcje operujące na łańcuchach znaków

Poniższe funkcje wymagają dołączenia biblioteki *string.h*.

- Funkcja kopiowania zawartości jednej tablicy znakowej do drugiej (ang. „string copy”).
 - Prototyp:

```
char *strcpy(char *dest, const char *src);
```

- Przykład użycia:

```
char txt1[] = "podstawy programowania w języku C";
char txt2[100];
// kopiowanie str1 do str2
strcpy(txt2, txt1);
printf(txt2);
```

- Funkcja porównująca teksty (ang. „string compare”):

- Funkcja zwraca wartość:
 - < 0 gdy tekst_1 < tekst_2
 - = 0 gdy tekst_1 == tekst_2
 - > 0 gdy tekst_1 > tekst_2

- Prototyp:

```
int strcmp(char *tekst_1, char *tekst_2);
```

- Przykład użycia:

```
char a[] = "Ola";
char b[] = "Ola";
// porównywanie tekstów
int porownaj = strcmp(b, a);
printf("%d\n", porownaj);
```

- Funkcja wyznacza i zwraca długość (ilość znaków) łańcucha s (bez znaku ‘\0’) - od ang. „string length”

- Prototyp:

```
size_t strlen(const char *s)
```

- Przykład użycia:

```
//długosc tekstu
char txt1[] = "Ola";
printf("%d\n", strlen(txt1));
```

- Funkcja dodaje łańcuch src (ang. source) do łańcucha dest (ang. destination) i zwraca wskaźnik na połączony łańcuch (dest) - od ang. „string concatenate”

- Prototyp:

```
char *strcat(char *dest, const char *src)
```

- Przykład użycia:

```
char dest[] = "Ala ma kota";
char src[] = " i dwa psy.";
// konkatencja tekstów (łączenie)
strcat(dest, src); // zwraca wskaźnik na połączone teksty
printf("%d\n", strlen(dest));
printf(dest);
```

- Funkcja szuka pierwszego wystąpienia znaku c w podanym łańcuchu s. Zwraca wskaźnik na znalezioną pozycję wystąpienia lub adres NULL. (od ang. „string char”)

- Prototyp:

```
char *strchr(const char *s, int c)
```

- Przykład użycia:

```
// szukanie pierwszego wystąpienia znaku w podanym lancuchu (tekście)
char str[] = "Jakis tam tekst do przeanalizowania";
char ch = 't';
char *p;
p = strchr(str, ch); // zwraca wskaznik na znaleziona pozycje wystapienia lub adres NULL
printf("string rozpoczynajacy sie od pierwszego wystapienia %c to: %s\n", ch, p);
```

- Funkcja szuka ostatniego wystąpienia znaku c w podanym łańcuchu s. Zwraca wskaźnik na znalezioną pozycję wystąpienia lub adres NULL. (od ang. „string right char”)
 - Prototyp:

```
char *strrchr(char *s, int c)
```

- Przykład użycia:

```
// szukanie ostatniego wystapienia znaku w podanym lancuchu (tekście)
char str2[] = "Jakis tam tekst do przeanalizowania";
char ch2 = 't';
char *p2;
p2 = strrchr(str2, ch2); // zwraca wskaznik na znaleziona pozycje wystapienia lub adres NULL
printf("string rozpoczynajacy sie od ostatniego wystapienia %c to: %s\n", ch2, p2);
```

- Funkcja szuka pierwszego wystąpienia łańcucha sub w podanym łańcuchu s. Zwraca wskaźnik na znalezioną pozycję wystąpienia lub adres NULL. (od ang. „scans string for substring”)
 - Prototyp:

```
char *strstr(char *s, const char *sub)
```

- Przykład użycia:

```
// szukanie pierwszego wystapienia jednego lancucha w drugim lancuchu
char strX[] = "To jest bardzo dlugi tekst, w ktorym bedzie szukany inny tekst.";
char searchStr[] = "tekst";
char *result;
result = strstr(strX, searchStr); // zwraca wskaznik na znaleziona pozycje wystapienia lub adres NULL.
printf("string rozpoczynajacy sie od szukanego wyrazenia: %s\n", result);
```

- Funkcja zamienia zawartość łańcucha s na duże/małe litery (od ang. „string upper” / „string lower”)
 - Prototypy:

```
char*strupr(char*s)
char*strlwr(char*s)
```

- Przykład użycia:

```
// zamiana na duze i male litery
char strY[] = "tekst pisany malymi literami";
strupr(strY); // zamiana na duze
printf("po zamianie na duze litery: %s\n", strY);
strlwr(strY); // zamiana na male
printf("po zamianie z powrotem na male litery: %s\n", strY);
```

Zadania

1. Przepisz i przeanalizuj działanie wszystkich przykładów (kodów) zawartych w tej instrukcji.
2. Napisz program, który będzie zliczał ile razy dana litera wystąpiła w analizowanym tekście. Znaki inne niż litery nie będą zliczane. Do budowy rozwiązania wykorzystaj tablice.
3. Napisz program, który będzie umożliwiał wypisywanie poszczególnych wyrazów zawartych w tekście – każdy wyraz ma być wypisany w osobnej linii.