
Podstawy programowania.

Wykład 6

Tablice znaków.

Tablice znaków i napisy

- Napis jest tablicą znaków z dodanym na zakończenie znakiem końca napisu `\0` :
 - `char s[] = "abc";` // równoważne: `char s[] = { 'a', 'b', 'c', '\0' }`
- Przykład powyżej podaje
 - inicjowanie tablicy bez podania rozmiaru (dotyczy nie tylko znaków)
 - kompilator automatycznie oblicza potrzebny rozmiar
 - specjalne inicjowanie napisu jako tablicy znaków – `"abc"`
 - standardowe inicjowanie tablicy kolejnymi wartościami - `{ 'a', ... }`
 - jeśli liczba wartości jest mniejsza od rozmiaru, uzupełnia się przez 0
- Operacji na napisach można dokonywać:
 - jak na tablicy znaków
 - dostęp do pojedynczych elementów tablicy
 - za pomocą specjalnych funkcji traktujących napis jako całość
 - wiele takich funkcji zawiera standardowa biblioteka C

Przykładowe operacje na tablicach znaków

- obliczenie długości napisu zapisanego w tablicy

```
int i;  
for (i = 0; napis[i] != '\0'; i++) { };  
return i;
```

- skopiowanie napisu - zawartości jednej tablicy do drugiej

```
int i=0; char c;  
while( (c=napis_z[i])!='\0' ) { napis_do[i]=c; i++; }  
napis_do[i] = '\0';
```

- porównanie napisów - zawartości dwóch tablic

```
int i=0; char c; char d; int wynik=0;  
do{  
    c=napis_1[i]; d=napis_2[i]; i++;  
    if(c!=d) { wynik=1; break; }  
} while (c!= '\0' && d!= '\0');
```

Przykład: wczytanie pojedynczej linii z *stdin*

```
#include <stdio.h>
#define MAXLINE 100 // maximum input line size

void main( void )
{
    char linia[MAXLINE];
    int zakres = MAXLINE - 1; // w napisie jako tablicy znaków miejsce na \0
    int znak; // do podstawiania znaków z getchar() - EOF nie jest znakiem
    int i;
    for (i=0; i<zakres && (znak=getchar()) != EOF && znak != '\n'; i++)
        linia[i] = znak;
    if (znak == '\n') { linia[i] = znak; i++; } // koniec linii też do tablicy
    linia[i] = '\0'; // od 0 do i - czyli i+1 elementów tablicy
    int dlugosc_linii = i; // liczymy tylko znaki, bez \0 !!!
    printf("%s",linia); // lub: for(i=0; i<dlugosc_linii; i++) printf("%c", linia[i]);
    return;
}
```

Przykład – zliczanie znaków z *stdin*

```
#include <stdio.h> // USES
void main(void)
{
    char c; int i, nwhite=0, nother=0, ndigit[10]={0}; // definicje i inicjowanie
    while((c = getchar()) != EOF) { // złożone wyrażenie - specjalność C
        switch (c) { // switch porównuje wyrażenie o typie int
            case '0': case '1': case '2': case '3': case '4': // z zestawem stałych wartości
            case '5': case '6': case '7': case '8': case '9': // także o typie int
                ndigit[c-'0']++; // printf("%c, %d, %d\n", c, c-'0', ndigit[c-'0']);
                break;
            case ' ': case '\n': case '\t':
                nwhite++; // po porównaniu wykonuje zestaw instrukcji i przechodzi dalej
                break; // można więc łączyć kilka wartości z tym samym zestawem instrukcji
            default: // etykieta default jest opcjonalna
                nother++; break; // break jest specjalną instrukcją, która przenosi sterowanie
        } // poza zakończenie danej instrukcji sterującej (dotyczy to switch i pętli)
    }
    printf("digits ="); for (i = 0; i < 10; i++) printf(" %d", ndigit[i]);
    printf(", white space = %d, other = %d\n", nwhite, nother);
}
```

Biblioteka standardowa C - <string.h>

→ Podstawowe funkcje obsługi napisów

- `strcat(s,t)` – dopisanie t na koniec s
- `strncat(s,t,n)` – to samo dla n pierwszych znaków t
- `strcmp(s,t)` – porównanie dwóch napisów
- `strncmp(s,t,n)` – to samo dla n pierwszych znaków
- `strcpy(s,t)` – skopiowanie t do s
- `strncpy(s,t,n)` – to samo dla co najwyżej n znaków
- `strlen(s)` – obliczenie długości napisu (bez \0)
- funkcje zaznaczone na czerwono mogą nie być bezpieczne, lepiej używać bezpiecznych wariantów z ograniczeniem liczby znaków na których operuje funkcja
 - niebezpieczeństwo polega na przekroczeniu rozmiaru (przepełnieniu) buforów, w których przechowywane są napisy

→ Przykład

```
char napis[20] = "Hello world!";  
if( strncmp(napis, "Bye bye world!", 15) != 0 ) printf("No bye bye!\n");
```

Przykład: wczytanie linii raz jeszcze

```
#define MAXLINE 60
void main( void )
{
    int i; int nr_linii;
    for(nr_linii=1; nr_linii<=10; nr_linii++){
        char linia[MAXLINE];
        int zakres = MAXLINE - 1; // w napisie jako tablicy znaków miejsce na \0
        int dlugosc_linii=0; // liczymy tylko znaki, bez \0 !!!
        int znak;
        for (i=0; i<zakres; ++i){
            znak = getchar();
            if(znak==EOF) break; // można przerwać także dla znak=='\n' - nie dodawać go do napisu
            linia[dlugosc_linii] = znak; dlugosc_linii++;
            if(znak == '\n') break; // znak nowej linii jest dodawany do napisu
        }
        linia[dlugosc_linii] = '\0'; // od 0 do dlugosc_linii -> dlugosc_linii+1 elementów
        // wykorzystanie funkcji bibliotecznej do umieszczania sformatowanych liczb w napisie:
        char linia_plus[MAXLINE+20]; // miejsce na numer linii
        sprintf(linia_plus, "linia numer %2d: \"%s\"\n", nr_linii, linia);
    }
}
```