

LABORATORIUM 12. ŁAŃCUCHY ZNAKOWE I FUNKCJE ŁAŃCUCHOWE.

Cel laboratorium:

Zapoznanie z metodami inicjalizacji łańcuchów znakowych i funkcjami działającymi na łańcuchach znakowych.

Zakres tematyczny zajęć:

- *pojęcie łańcucha znaków,*
- *stałe i zmienne łańcuchowe,*
- *wczytywanie i wyprowadzanie łańcuchów,*
- *funkcje standardowe na łańcuchach z biblioteki string.h.*

Kompendium wiedzy:

Łańcuch znakowy: tablica typu char zakończona znakiem zerowym (`\0`).

Definiowanie /inicjacja łańcuchów:

- **Stała łańcuchowa** (literał łańcuchowy) - sekwencja znaków w `""`.
- **Zmienna łańcuchowa** - inicjacja tylko w definicji.

```
#define FIRMA "ABC"
#define DL 20

char imie[10];
char nazwisko[DL];
char dyrektor[]="Kowalski"; //char *dyrektor="Kowalski";
char prezes[]={ 'K', 'o', 's', '\0' };
printf("Dyrektorem firmy \" %s\" jest %s a prezesem %s\n",
      FIRMA, dyrektor, prezes);
```

Wczytywanie łańcuchów - funkcje: `scanf()`, `gets()`, `fgets()`:

- Funkcja **`scanf()`** służy do wczytywania różnych zmiennych, w tym łańcuchów, pobierając znaki ze standardowego urządzenia wejścia do chwili napotkania znaku niedrukowanego (nowej linii `'\n'`, tabulatora, spacji), który pozostaje. Funkcja ta dodaje na końcu łańcucha znak `'\0'`.

```
scanf("%s", tablica_znakowa);
```

```
scanf("%s", nazwisko);
```

- Funkcja **`gets()`** służy do wczytywania łańcuchów pobierając znaki (w tym spacje) ze standardowego urządzenia wejścia do chwili napotkania znaku nowej linii `'\n'` (ale bez niego), dodaje na końcu znak `'\0'` i przekazuje łańcuch do programu. Nie sprawdza liczby wprowadzanych znaków z zadeklarowanym rozmiarem tablicy.

```
gets(tablica_znakowa);
```

```
gets(imie);
```



<ul style="list-style-type: none"> Funkcja fgets() umożliwia podanie liczby znaków, które zostaną wprowadzone. Po odczytaniu znaku '\n' pozostawia go w łańcuchu.
fgets(nazwa_tablicy, liczba_znakow, strumien_wej);
<pre>fgets(nazwisko, DL, stdin);</pre> <p>Wyprowadzanie łańcuchów - funkcje: printf(), puts() fputs():</p> <ul style="list-style-type: none"> Funkcja printf() służy do wyprowadzania wartości różnych zmiennych, w tym łańcuchów, na standardowe urządzenie wyjścia i można ją użyć w następujących postaciach (znak '\n' należy dodać do łańcucha formatującego, by przejść do nowej linii):
<pre>printf(stala_lancuchowa); printf("%s\n", tablica_znakowa); printf("%ps\n", tablica_znakowa);</pre>
<p>gdzie: p - minimalna liczba znaków wyprowadzanego pola, jeżeli p jest ze znakiem minus, zmienna w polu wyrównywana jest do lewej.</p> <pre>printf("Witaj w swiecie programowania "); printf("%s\n", nazwisko); printf("Witaj %20s\n", nazwisko);</pre> <ul style="list-style-type: none"> Funkcja puts() wyprowadza łańcuch zakończony znakiem '\0' i do wyprowadzanego łańcucha dodaje znak końca linii:
<pre>puts(stala_lancuchowa); puts(tablica_znakowa);</pre>
<pre>puts(FIRMA); puts(nazwisko);</pre> <ul style="list-style-type: none"> Funkcja fputs() wyprowadza łańcuch na wskazane wyjście (plik lub ekran - stdout). Przy wyświetleniu fputs() nie dodaje znaku przejścia do nowego wiersza
fputs(tablica_znakowa, stdout);
<pre>fputs(nazwisko, stdout);</pre> <p>Funkcje łańcuchowe z biblioteki string.h:</p> <ul style="list-style-type: none"> Operacja podstawienia (kopiowania): Poza inicjalizacją łańcuchów w chwili definiowania ich typów, napisów nie można podstawiać do tablic przez zastosowanie znaku '='. Należy użyć kopiowania łańcuchów:
<pre>strcpy(dokad, skad); strncpy(dokad, skad, maxliczba);</pre>
<p>gdzie: dokad – tablica znakowa lub napis, skad – napis lub stała napisowa, maxliczba - maksymalna liczba znaków do skopiowania.</p> <pre>strcpy(nazwisko, "Nowak"); strncpy(nazwisko, "Nowak", DL-1)</pre>

<ul style="list-style-type: none"> Operacja łączenia (konkatenacja): Funkcja dołącza do dotychczasowej zawartości napis1 zawartość napis2, pierwszy argument zmienia się, drugi pozostaje bez zmian.
<pre>strcat (napis1, napis2); strncat (napis1, napis2, maxliczba);</pre>
<pre>strcat(nazwisko, imie); max= DL-strlen(imie)-1 strncat(nazwisko, imie,);max);</pre> <ul style="list-style-type: none"> Porównywanie napisów: Funkcja zwraca wartość 0, gdy napisy są równe leksykograficznie (słownikowo), wartość dodatnią, gdy napis1 jest słownikowo większy (leży dalej w słowniku), wartość ujemną, gdy napis1 jest mniejszy (leży bliżej początku słownika). Przy porównywaniu funkcja korzysta z kodów ASCII (cyfry < duże litery < małe litery).
<pre>strcmp (napis1, napis2); strncmp (napis1, napis2, maxliczba);</pre>
<pre>strcmp(dyrektor, nazwisko); strncmp(dyrektor, nazwisko,5);</pre> <ul style="list-style-type: none"> Obliczanie długości łańcucha znaków: Funkcja oblicza długość łańcucha znaków od pierwszego znaku do znaku terminalnego '\0'.
<pre>strlen (napis1);</pre>
<ul style="list-style-type: none"> Konwersja tekstu na liczbę:
<pre>atoi(napis); //int atol(napis); //long atof(napis); //double</pre>
<pre>int i=atoi("1234"); double x=atof("56.78"); printf("%d\n",i); printf("%f\n",x);</pre>

Pytania kontrolne:

1. Co to jest łańcuch? Czym różni się zapis 'x' od "x"?
2. Jak zdefiniować stałą, a jak zmienną łańcuchową?
3. Podaj funkcje do wczytywania łańcuchów i ich zastosowania.
4. Podaj funkcje do wyświetlania łańcuchów i ich zastosowania.
5. Jak obliczyć długość łańcucha?
6. Jak skopiować łańcuchy?
7. Jak połączyć dwa łańcuchy w jeden łańcuch?
8. Jak porównać łańcuchy?



Zadania do analizy

Zadanie 12.1. Operacje na łańcuchach

- Przeanalizuj przykład programu wykorzystującego łańcuchy.
- Podaj tekst w komentarzach.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h> //???
4  #include <locale.h>
5  #define FIRMA "ABC"
6
7  int main(int argc, char *argv[]) {
8  setlocale(LC_ALL, ""); //???
9  char dyrektor[]="Kowalski"; //char *dyrektor="Kowalski";
10 char prezes[]={ 'K', 'o', 's', '\0' }; //???
11 printf("Dyrektorem firmy \" %s\" jest %s a prezesem %s\n",
FIRMA, dyrektor, prezes); //???
12 char str1[5], str2[20], str3[7], str4[20], str5[20],
str6[50];
13 printf("Kto jest autorem Pana Tadeusza\n");
14 gets(str1); //???
15 puts (str1); //???
16 printf("Kto jest autorem Akademii Pana Kleksa\n");
17 scanf("%s",str2); //???
18 printf ("Autor: %s znany i lubiany\n",str2); //???
19 printf ("Autor: %20s znany i lubiany\n",str2); //???
20 printf ("Autor: %-20s znany i lubiany\n",str2); //???
21 printf("=====\n");
22 printf("Kto jest autorem Pana Tadeusza ");
23 fgets(str3,7,stdin); //???
24 fputs(str3,stdout); //???
25 printf("=====\n");
26 strcpy(str4, "Witam "); //???
27 printf ("%s\n", str4);
28 strcpy(str5, "tutaj"); //???
29 printf("%s\n",str5);
30 printf ("Konkatenacja str4 i str5:\n");
31 strcat(str4, str5); //???
32 printf ("%s\n", str4);
33 printf("Kopiowanie str4 do str6:\n");
34 strcpy(str6, str4);
35 printf ("%s\n", str6);
36 printf("Dlugosc str6:%d\n",strlen(str6)); //???
37 return 0;
38 }
```



Zadanie 12.2. Kopiowanie łańcuchów

- Przeanalizuj przykład programu wykorzystującego łańcuchy.
- Podaj tekst w komentarzach.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>          //???
4  #define DL 30
5  #define MAXDL 6
6  #define N 3
7
8  int main(int argc, char *argv[]) {
9  char slowa[3][MAXDL];        //???
10 char temp[DL];               //???
11 int i=0;
12 printf("Podaj %d slow zaczynajacych sie na a:\n",N);
13 while(i<N && gets(temp))     //???
14 { if(temp[0] != 'a')          //???
15     printf("%s nie zaczyna sie na a!\n", temp);
16     else
17     {strncpy(slowa[i], temp, MAXDL-1); //???
18      slowa[i][MAXDL-1]='\0';         //???
19      i++;
20     }
21 }
22 printf("Przyjete slowa:\n")    ;
23 for(i=0;i<N;i++)
24     puts(slowa[i]);            //???
25
26 printf("Podaj %d slowa do skrocenia\n",N);
27 while(i>0 && gets(temp))
28 {
29     if(strlen(temp)>MAXDL)       //???
30     *(temp+MAXDL)='\0';         //???
31     puts(temp);
32     i--;
33 }
34 return 0;
35 }
```

Zadania do wykonania

Zadanie 12.3. Wyśrodkowanie tekstu

Napisz funkcję pozwalającą na wyśrodkowywanie podanego tekstu. Przyjmij, że liczba znaków w linii wynosi 80, zaś tekst jest wprowadzany bez kontrolowania długości wprowadzanych znaków.



Zastosuj tę funkcję do wyświetlenia:

- nazwy przedmiotu, zdefiniowanego jako stała łańcuchowa,
- twoich danych osobowych (imię i nazwisko), zainicjowanych podczas definicji,
- danych kolegi (imię i nazwisko), wczytanych z klawiatury.

Zadanie 12.4. Formatowanie wiersza

Napisz funkcje pozwalające na wyrównywanie do lewej i wyrównanie do prawej tekstu.

Zastosuj te funkcje do wyświetlenia twojego imienia i nazwiska.

Zastosuj te funkcje do wyświetlenia wiersza *Lokomotywa* Juliana Tuwima:

*Stoi na stacji lokomotywa,
Ciężka, ogromna i pot z niej sphywa:
Thusta oliwa.
Stoi i sapie, dyszy i dmucha,
Żar z rozgrzanego jej brzucha bucha:
Buch - jak gorąco!
Uch - jak gorąco!*

Przyjmij, że liczba znaków w linii wynosi 80, zaś tekst jest wprowadzany bez kontrolowania długości wprowadzanych znaków. Wiersz zapisz w jednej tablicy dwuwymiarowej z zainicjowanymi wartościami.

Napisz program, w którym wczytane są dane, wywołana funkcja, wyświetlone wyniki.

Zadanie 12.5. Obliczanie długości napisu

Napisz program obliczający długość zdania powstałego przez konkatencję następujących stałych łańcuchowych: "Biblioteka", "string.h", "pozwala", "na", "zastosowanie", "funkcji", "do", "łańcuchow", "znakowych". Pamiętaj o wprowadzeniu spacji, aby powstałe zdanie wyglądało właściwie.

Zadanie 12.6. Zamiana tekstu na duże litery

Napisz funkcję pozwalającą na zamianę wprowadzonego tekstu na tekst, który będzie wypisany dużymi literami. Wprowadzony tekst może zawierać duże, małe litery oraz inne znaki. Do tego celu zastosuj funkcję kontrolującą liczbę wprowadzonych znaków. Wypisz napis wprowadzony i zmieniony. Wykorzystując funkcję z zadania 12.2 wyśrodkuj zmieniony napis.

Napisz program, w którym wczytane są dane, wywołana funkcja, wyświetlone wyniki.

Zadania dodatkowe

Zadanie 12.7. Zakończenie działania programu

Napisz funkcję, w której zakończenie programu nastąpi po wprowadzeniu napisu "quit". Program pozwala na wprowadzanie wielu napisów (pętla nieskończona).

Napisz program, w którym wczytane są dane, wywołana funkcja, wyświetlone wyniki.



Zadanie 12.8. Słownikowe porządkowanie wyrazów

Wersja podstawowa:

Napisz funkcję, która uporządkuje leksykograficznie (od litery a do z) pięć trzyliterowych wyrazów wprowadzonych z klawiatury, uwzględniając tylko pierwszą literę wyrazów.

Napisz program, w którym wczytane są dane, wywołana funkcja, wyświetlone wyniki.

Wersja zaawansowana:

Napisz funkcję, która uporządkuje leksykograficznie (od litery a do z) pięć trzyliterowych wyrazów wprowadzonych z klawiatury, uwzględniając kolejne litery wyrazów.

Napisz program, w którym wczytane są dane, wywołana funkcja, wyświetlone wyniki.

Zadanie 12.9. Tworzenie napisów w odbiciu lustrzanym

Napisz funkcję, pozwalającą odwrócić kolejność znaków wprowadzonego napisu. W wersji rozbudowanej zastosuj funkcję zamieniającą napis na duże litery (zadanie 12.5), oraz wyśrodkowujący go (zadanie 12.2, liczba znaków w linii 80).

Napisz program, w którym wczytane są dane, wywołane funkcje, wyświetlone wyniki.

Zadanie 12.10. Szyfr Cezara

Napisz funkcję, która pozwoli na szyfrowanie wprowadzonego tekstu stosując szyfr Cezara. Zastosuj przesunięcie znaków o wartość 3. Zakończenie wprowadzania napisów nastąpi po wpisaniu napisu "quit". Wypisz napisy zaszyfrowane.

Napisz drugą funkcję dekodującą teksty zaszyfrowane kodem Cezara.

Zadanie 12.11. Moduł z funkcjami do obsługi łańcuchów

Zgrupuj wszystkie własne funkcje do obsługi łańcuchów w module `lancuchy.h`. Napisz program, w którym przetestujesz funkcje z tego modułu.

