

Programowanie strukturalne - Wykład 8

Napisy (łańcuchy znaków)

Napisy

Napis - ciąg składający się z co najmniej jednego znaku.

Znaki cudzysłowu nie są częścią łańcucha.

Język C nie posiada typu `string`/łańcuchowego. Wszystkie napisy traktowane są jako tablice typu `char`. Ostatnim znakiem w tablicy jest znak `\0`.

Znak a napis

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     char a = 'q';
7     char b[] = "q";
8     return 0;
9 }
```

srtlen a sizeof

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main()
6 {
7     char nap1[] = "Hello World";
8     char nap2[50] = "Hello World";
9     printf("%Iu\n", sizeof nap1);
10    printf("%Iu\n", strlen(nap1));
11    printf("%Iu\n", sizeof nap2);
12    printf("%Iu\n", strlen(nap2));
13    return 0;
14 }
```

Tablica a wskaźnik

```
1  #include <stdio.h>
2  #define NAPIS "jakiś tekst"
3
4  int main()
5  {
6      char tab[] = NAPIS;
7      const char *wsk = NAPIS;
8      printf("adres napisu %p\n", "jakiś tekst");
9      printf("adres tab: %p\n", tab);
10     printf("adres wsk: %p\n", wsk);
11     printf("adres NAPIS-u: %p\n", NAPIS);
12     return 0;
13 }
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char nap1[] = "absddfvjskjf";
6     char *nap2 = "oijefj";
7     nap1[4] = 'M';
8     *(nap1 + 7) = 'M';
9     nap2[2]='3'; // czy to zawsze możliwe?
10    return 0;
11 }
```

Kopiowanie napisu

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char * napis = "ab6sWR";
6      char * kopia;
7      kopia=napis;
8      printf("%s\n", napis);
9      printf("%p\n", napis);
10     printf("%p\n", &napis);
11     printf("%s\n", kopia);
12     printf("%p\n", kopia);
13     printf("%p\n", &kopia);
14     return 0;
15 }
```

czy można to zrobić notacją tablicową?

Wczytywanie napisów

- scanf

<https://pl.wikibooks.org/wiki/C/scanf>

<https://en.cppreference.com/w/c/io/scanf>

- gets

<https://pl.wikibooks.org/wiki/C/gets>

<https://en.cppreference.com/w/c/io/gets>

- fgets

<https://pl.wikibooks.org/wiki/C/fgets>

<https://en.cppreference.com/w/c/io/fgets>

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int a;
7     scanf("%i", &a);
8     printf("%i\n", a);
9     return 0;
10 }
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int n,m;
7     scanf("%d, %d", &n, &m);
8     printf("%d %d\n", n, m);
9     return 0;
10 }
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     char znak;
7     scanf(" %c",&znak);
8     printf("%c\n",znak);
9     return 0;
10 }
```

Wskaźnik czy tablica?

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char * slowo;
6     scanf("%s", slowo);
7     printf("%s\n", slowo);
8     return 0;
9 }
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char slowo[20];
6     scanf("%s", slowo);
7     printf("%s\n", slowo);
8     return 0;
9 }
```

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char slowo[5];
6      gets(slowo);
7      printf("%s\n", slowo);
8      puts(slowo);
9      return 0;
10 }
```

- wpisz różnej długości napisy (niektóre ze spacjami)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char slowo[5];
6      gets_s(slowo, 4*sizeof(char));
7      printf("%s\n", slowo);
8      puts(slowo);
9      return 0;
10 }
```

- nie działa w każdej konfiguracji


```
1 #include <stdio.h>
2
3 int main()
4 {
5     char slowo[5];
6     fgets(slowo, 5, stdin);
7     printf("%s\n", slowo);
8     puts(slowo);
9     fputs(slowo, stdout);
10    return 0;
11 }
```

Różnice?

- `scanf` - do znaku niedrukowanego, reszta do końca linii
- `gets` - mało bezpieczna przy przepełnieniu
- `fgets` - dodaje koniec linii na końcu napisu

Wyświetlanie napisów

- `printf`

<https://pl.wikibooks.org/wiki/C/printf>

<https://en.cppreference.com/w/c/io/fprintf>

- `puts`

<https://pl.wikibooks.org/wiki/C/puts>

<https://en.cppreference.com/w/c/io/puts>

- `fputs`

<https://pl.wikibooks.org/wiki/C/fputs>

<https://en.cppreference.com/w/c/io/fputs>

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char tekst1[]="abc";
6      char tekst2[]= {'a','b','c'};
7      char tekst3[]="xyz";
8      printf("%p\n", tekst1);
9      printf("%p\n", tekst2);
10     printf("%p\n", tekst3);
11     puts(tekst1);
12     puts(tekst2);
13     puts(tekst3);
14     return 0;
15 }
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char tekst1[]="abc";
6     char tekst2[]= {'a', 'b', 'c'};
7     char tekst3[]="xyz";
8     puts(tekst1);
9     puts(tekst2);
10    puts(tekst3);
11    return 0;
12 }
```

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char tekst1[]="abc";
6     char tekst2[]= {'a', 'b', 'c'};
7     char tekst3[]="xyz";
8     fputs(tekst1, stdout);
9     fputs(tekst2, stdout);
10    fputs(tekst3, stdout);
11    return 0;
12 }
```

```
1  #include <stdio.h>
2
3  int main()
4  {
5      char tekst1[]="abc";
6      char tekst2[]= {'a', 'b', 'c'};
7      char tekst3[]="xyz";
8      printf("%s", tekst1);
9      printf("%s", tekst2);
10     printf("%s", tekst3);
11     return 0;
12 }
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      char buffer[20];
7      int a=5;
8      int b=7;
9      sprintf(buffer, "%5d+%5d=%5d", a, b, a+b);
10     printf("%s", buffer);
11     return 0;
12 }
```



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     char buffer[20];
7     int a=5;
8     int b=7;
9     snprintf(buffer,20*sizeof(char), "%5d+%5d=%5d", a,b, a+b);
10    printf("%s",buffer);
11    return 0;
12 }
```

Bibliografia

- Richard Reese, Wskaźniki w języku C, Wydawnictwo Helion 2014.

