

```
1 // W6_1.cpp : Defines the entry point for the console application.
2 //      Przekazanie tablic w argumentach funkcji
3 //      Przykład: obliczenie ilosci slow w wiersze tekstu
4 //UWAGA! Nie jest konieczne przekazywac tablicy od poczatku !!!
5 //      Do funkcji mozna przekazac tylko czesc tablicy
6
7 #include "stdafx.h"
8 #include <stdio.h>
9 #include <string.h>
10 #define MAX_LINE 512
11
12 //Steruje trybem pracy funkcji fun2.
13 //PRACUJU_ZE_WSKAZNIKIEM - fun2 realizuje dostep do elementow tablicy      ↗
14 //      przez wskaznik do tablicy.
15 //      Jest uzywana arytmetyka wskaznikow
16 // !PRACUJU_ZE_WSKAZNIKIEM - fun2 realizuje dostep do elementow tablicy      ↗
17 //      przez identyfikator tablicy
18 //define PRACUJU_ZE_WSKAZNIKIEM
19
20 static int fun1(char tt[]);
21 static int fun2(char *ptr_str);
22
23 int _tmain(int argc, _TCHAR* argv[])
24 {
25     char str[MAX_LINE] = "no";
26     int no_slow = -1, no_slow1 = -1;
27
28     //przekazanie tablicy jako argumentu faktycznego, poczynajac z adresu      ↗
29     //3 pozycji
30     char str1[] = "abcdef";
31     fun1(&str1[3]);
32     fun2(&str1[3]);
33
34     //liczymy, ile slow w wierszu tekstowym
35     while(strcmp(str, "yes") != 0)
36     {
37         printf_s("podaj weirsz tekstowy\n");
38         gets_s(str);
39
40         no_slow = fun1(str); //przekazanie tablicy przez referencje
41         no_slow1 = fun2(str); //przekazanie tablicy przez wskaznik
42         printf_s("ilosc slow w wiersze: %d\n", no_slow);
43         printf_s("ilosc slow w wiersze: %d\n", no_slow1);
44
45         printf_s("przejsc do nastepnego wiersza - ENTER, skonczyc - yes      ↗
46             \n");
47         gets_s(str);
48     }
49
50     return 0;
51 }
52
53 int fun1(char tt[])
```

```
50 /
    *=====
    =====
51 Liczy ilosc slow w wiersze tt
52 Metoda: poczatek kazdego nowego slowa jest pierwszy symbol, który sie
    rozni
53     od spacji. Mamy: prev - symbol poprzedni, symb - symbol biezacy.
    Wiec
54     poczatek nowego slowa - to jest "PRAWDA" wyrazenia logicznego:
55     symb != ' ' ^ prev == ' '.
56 Przekazanie tablicy przez referencje
57 IN: wiersz znakow tt
58 RetVal: ilosc slow w tt
59 =====
    =====*/
60 {
61     int it = 0;
62     char prev = ' ';
63     char symb;
64     int pos = 0;
65     while((symb = tt[pos++]) != '\0')
66     {
67         if(symb != ' ' && prev == ' ')
68             it++;
69         prev = symb; //przestaw symbol prev do nastepnej iteracji
70     }
71     return it;
72 }
73 }//fun1
74
75 int fun2(char *ptr_str)
76 /
    *=====
    =====
77 Liczy ilosc slow w wiersze tt
78 Metoda: poczatek kazdego nowego slowa jest pierwszy symbol, który sie
    rozni
79     od spacji. Mamy: prev - symbol poprzedni, symb - symbol biezacy.
    Wiec
80     poczatek nowego slowa - to jest "PRAWDA" wyrazenia logicznego:
81     symb != ' ' ^ prev == ' '.
82 Przekazanie tablicy po wskazniku
83 IN: wskaznik na wiersz znakow ptr_str
84 RetVal: ilosc slow w wiersze
85 =====
    =====*/
86 {
87     int it = 0;
88     char prev = ' ';
89
90 #ifdef PRACUJU_ZE_WSKAZNIKIEM
91     while (*ptr_str != '\0')
92     {
```

```
93     if (*ptr_str != ' ' && prev == ' ')
94         it++;
95
96     prev = *ptr_str; //przestaw symbol prev do nastepnej iteracji
97     ptr_str++;       //przestaw wskaznik do nastepnego symbolu      ↗
                       biezacego
98 }
99 #else
100     int pos = 0;
101     while (ptr_str[pos] != '\0')
102     {
103         if (ptr_str[pos] != ' ' && prev == ' ')
104             it++;
105
106         prev = ptr_str[pos++]; //przestaw symbol prev do nastepnej      ↗
                                iteracji
107     }
108 #endif
109     return it;
110 }//fun1
111
112
113
114
```