

```

1      /*
      File:      delete_s.cpp

      КРВ:      Как в C++ удалить какой - либо символ из строки?

      Задача.   Дана строка
                "0123456789".
                Удалить из нее символ '6'

      Решение.  MSP, 25.10.07, 22:31-23:34

      Site:     http://www.Best-Listing.ru/
*/

10     # include <stdio.h>
11     # include <conio.h>
12     # include <string.h>

13     int main ()
14     {
15         clrscr ();

16         char * T = "0123456789";
17         // на удивление, так писать можно!
18         // Что такое
19         // "1234567890"?
20         // Это строка, занимающая 11 байт в памяти,
21         // адрес этой строки задается OS один раз, он является
22         // постоянным, неизменяемым, назовем его A1, и всегда
23         // по этому адресу находится начало строки "1234567890".
24         //
25         // Итак, еще раз.
26         // Что такое строка
27         // "0123456789"?
28         // это АДРЕС A1
29         //
30         // Что такое
31         // char * T
32         // ?
33         // Это указатель на строку.
34         // Что является значением указателя?
35         // - Значением указателя является АДРЕС ПАМЯТИ.
36         // Получается, что в наших обозначениях в T записан адрес A1.
37         //
38         // Что такое инициализация?
39         // Это задание ПЕРВОНАЧАЛЬНОГО значения переменной.
40         // Таким образом УКАЗАТЕЛЬ T инициализирован.
41         // Первоначальное значение T равно A1.

42         // дана строка

```

```

43 puts ("ПЕРВЫЙ способ:");
44 printf ("данная T = %s\n", T);
45 // чтобы напечатать строку, нужно указать в printf АДРЕС начала
46 // строки. Печать идет с указанного адреса до нуля-символа

47 // ПЕРВЫЙ способ

48 // Мы хотим вырезать из строки символ '6'.
49 // Строка T заканчивается нуля-символом '\0'.
50 // Чтобы обрезать строку нужно записать
51 // после ее КОНЦА нуля-символ.
52 // Так и сделаем.
53 * (T + 6) = '\0';

54 // Функция printf () печатает строки до тех пор, пока не
55 // достигнет нуля-символа. Как только обнаружен нуля-символ,
56 // так сразу печать строки прекращается.
57 printf ("обрезанная T = %s\n", T);

58 // Сколько байт останется на остаток строки T?
59 // -- 4 байта: "890" + '\0'
60 char OST [4];

61 // Как работает f. strcpy ()?
62 // Эту функцию можно понимать так:
63 // strcpy (A2, A3);
64 // С указанного адреса A3 переписываются байт за байтом
65 // начиная с адреса A2 символы, пока не прочтется нуля-символ
66 // в конце строки A3.
67 strcpy (OST, T + 7);
68 // теперь в OST находится "890"

69 // Склеим строки
70 // Это делается с помощью f. strcat ()
71 // Ее можно представить так:
72 // strcat (str1, str2);
73 // Мы хотим получить str1 + str2.
74 // Но это ОЧЕНЬ ОПАСНАЯ ФУНКЦИЯ, потому что после склеивания
75 // мы можем случайно зайти за рамки разрешенной памяти для
76 // хранения str1.
77 // Например, может случиться так:
78 // OS разрешила с адреса str1 расположить не более 101 символа.
79 // Сейчас, например, нуля-символ находится на str1 + 27.
80 // Это означает, что приклеить к str1 можно не более 83 символов!

81 // Сейчас T уже обрезана, и осталось всего 4 байта, причем
82 // всегда нужно помнить, что последний байт - нуля-байт.
83 // Длина T сейчас 6 + 1 байт
84 // приклеим OST, но приклеиваться OST начнет с нуля-байта
85 // строки T
86 strcat (T, OST); // места хватило, теперь, даже в конце строки

```

```
87             // ДВА нуль-байта

88     // вот и вырезали ОДИН символ из строки, как требовалось в задаче!
89     printf ("после склейки, ответ: T = %s\n", T);

90     // Если вы поняли, что написано выше, то вот, ВТОРОЙ СПОСОБ:
91     char * S = "0123456789";
92     strcpy (S + 6, S + 7);

93     printf ("\nВТОРОЙ способ: S = %s\n", S);

94     getch ();

95     return 0;
96 }
```

Sergey Mitrofanov, 17.09.13

E-mail: infostar@mail.ru

© <http://www.Best-Listing.ru>, 2006–2013