

```

1  /*
2  * -- M.G. Maffucci --
3  * 4. In un vettore alfanumerico trovare la stringa (o le
4  * stringhe) di lunghezza maggiore, precisandone la posizione
5  * occupata all'interno del vettore.
6  */
7
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <string.h>
11
12 int main()
13 {
14     //costanti
15     const int NMAX = 10;    //dimensione massima del vettore
16                             // contenente le stringhe
17     const int LMAX = 30;    //lunghezza massima delle stringhe
18                             // nel vettore
19
20     //input
21     char stringhe[NMAX][LMAX]; //vettore di stringhe
22     int n;                      //numero di stringhe da
23                                 // inserire nel vettore
24     int lungMax;                //lunghezza massima delle
25                                 // stringhe
26     int i;                      //posizione nel vettore
27     int daButtare;             //variabile usata per ripulire
28                                 // lo stdin
29
30     /*
31     * Inserisco il numero di stringhe da inserire controllando
32     * che sia compreso fra 1 e 10
33     */
34     do{
35         printf("Quante stringhe vuoi inserire (1-10)?: ");
36         scanf("%d", &n);
37     } while((n < 1) || (n > NMAX));
38     /*
39     * Questo while vuoto (notate il ; subito dopo) mi permette
40     * di ripulire lo stdin senza usare la fflush() che non e'
41     * piu' consigliato utilizzare dall'ultima versione del C.
42     * Quella proposta e' una delle possibili soluzioni: la
43     * funzione getchar() legge un carattere per volta dallo
44     * stdin, memorizzandolo nella variabile daButtare,
45     * fino a quando non trova l'invio che ne provoca la
46     * conclusione.
47     */
48     while( (daButtare = getchar()) != '\n');
49     /*
50     * Inserisco nel vettore le stringhe.
51     */
52     for(i = 0; i < n; i++){
53         printf("Stringa %d: ", (i + 1));
54         fgets(stringhe[i], LMAX, stdin);
55         //la seguente istruzione permette di eliminare l'invio
56         // dalla stringa appena caricata nel vettore
57         stringhe[i][(strlen(stringhe[i]) - 1)] = '\0';
58     }
59     /*
60     * Eseguo la ricerca della lunghezza maggiore delle stringhe.
61     */
62     //inizializzo la variabile lungMax alla lunghezza della
63     // prima stringa del vettore
64     lungMax = strlen(stringhe[0]);
65     for(i = 1; i < n; i++){
66         if( lungMax <= strlen(stringhe[i]) ){
67             lungMax = strlen(stringhe[i]);
68         }
69     }

```

```
68     }
69     /*
70     * Ciclo di visualizzazione delle stringhe di lunghezza
71     * maggiore e della relativa posizione.
72     */
73     for(i = 0; i < n; i++){
74         if( strlen(stringhe[i]) == lungMax ){
75             printf("La stringa \"%s\" che si trova nella posizione %d ha
la dimensione massima\n", stringhe[i], (i + 1));
76         }
77     }
78     return 0;
79 }
80
```