

```

1  /*
2  * -- M.G. Maffucci --
3  * 5. Ordinare un vettore alfanumerico in base alla lunghezza
4  * delle sue componenti (per prima la stringa di lunghezza
5  * minore, e cosi' via fino all'ultima stringa che e' quella di
6  * lunghezza maggiore).
7  *
8  * NOTA: l'algoritmo riguarda l'ordinamento crescente degli
9  * elementi di un vettore.
10 /*
11 #include <stdio.h>
12 #include <stdlib.h>
13 #include <string.h>
14
15 #define NMAX 30      //numero massimo di stringhe caricate nel vettore
16 #define LMAX 50     //lunghezza massima di ogni singola stringa caricata
17 int main()
18 {
19     //input
20     char stringhe[NMAX][LMAX]; //vettore alfanumerico
21     int n; //numero di stringhe da inserire nel vettore
22     //lavoro
23     int i; //indice del vettore
24     int j; //indice del vettore utilizzato per l'ordinamento
25     char comodo[LMAX]; //stringa di supporto da utilizzare nello
26     // scambio di posizioni fra le stringhe
27     // del vettore
28     char daButtare; //variabile usata per ripulire lo stdin
29     /*
30     * Inserisco il numero di stringhe da caricare nel vettore,
31     * controllando la validita' dell'input.
32     */
33     do{
34         printf("Inserisci il numero di parole da caricare (1-30): ");
35         scanf("%d", &n);
36     } while( (n < 1) || (n > NMAX));
37     /*
38     * Questo while vuoto (notate il ; subito dopo) mi permette di ripulire lo stdin
39     * senza usare la fflush() che è consigliato non usare più dall'ultima versione del C.
40     * Quella proposta è una delle possibili soluzioni: la funzione getchar() legge un
41     * carattere per volta dallo stdin, memorizzandolo nella variabile dabuttare,
42     * fino a quando non trova l'invio che ne provoca la conclusione.
43     */
44     while( (daButtare = getchar()) != '\n');
45     /*
46     * Carico il vettore di stringhe.
47     */
48     for(i = 0; i < n; i++){
49         /*
50         * L'inserimento delle stringhe lo eseguo utilizzando la
51         * funzione fgets() che, in questo, caso legge NMAX
52         * caratteri di input, piu' il carattere di invio, dallo
53         * standard input (stdin), memorizzandoli nella posizione
54         * i-sima del vettore.
55         */
56         printf("Inserisci la stringa: ");
57         fgets(stringhe[i], NMAX, stdin);
58         /*
59         * Eliminiamo dalla stringa il carattere di invio che
60         * si trova alla fine e che e' stato memorizzato dalla
61         * fgets().
62         */
63         stringhe[i][strlen(stringhe[i]) - 1] = '\0';
64     }
65     /*
66     * Ciclo di ordinamento in base alla lunghezza delle

```

```
67     * stringhe.
68     */
69     for(i = 0; i < (n - 1); i++){
70         for(j = (i + 1); j < n; j++){
71             if(strlen(stringhe[i]) > strlen(stringhe[j])){
72                 strcpy(comodo, stringhe[i]);
73                 strcpy(stringhe[i], stringhe[j]);
74                 strcpy(stringhe[j], comodo);
75             }
76         }
77     }
78     /*
79     * Visualizzo il vettore ordinato
80     */
81     for(i = 0; i < n; i++){
82         printf("%s\n", stringhe[i]);
83     }
84     return 0;
85 }
```