

```

1 //NOTO: il problema può essere risolto usando una pila implementata come lista
  semplice, senza aumentare il costo computazionale.
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 // struct per memorizzare ogni riga
7 typedef struct
8 {
9     int n;
10    float f;
11    char s[6 + 1];    // ogni stringa deve avere spazio per \0
12    struct data *succ; // Puntatore all'elemento successivo della lista
13    struct data *prec;
14 } data;
15
16 int main()
17 {
18     FILE *fp;                // Puntatore al file prima di input, poi di output
19     data *head, *tail, *tmp; // HEAD: inizio lista, TAIL: fine lista, TMP: variabile
    di supporto
20     // Inizializzazione lista vuota
21     head = NULL;
22     tail = NULL;
23
24     // Apro file input
25     fp = fopen("data_es2_input", "r");
26
27     // Alloco il primo elemento della lista
28     tmp = (data *)malloc(sizeof(data));
29     tmp->prec = NULL;
30     tmp->succ = NULL;
31     head = tmp;
32     tail = tmp;
33
34     // Inserimento in fondo alla lista fino a quando la lista non è vuota
35     while (fscanf(fp, " %d %f %6s", &(tmp->n), &(tmp->f), tmp->s) != EOF)
36     {
37         tmp = (data *)malloc(sizeof(data));
38         tmp->prec = tail;
39         tmp->succ = NULL;
40         tail->succ = tmp;
41         tail = tmp;
42     }
43     // il ciclo while viene eseguito una volta di troppo, quindi libero tmp
    aggiornando la coda
44     // Ho preferito usare questo espediente in modo da poter usare fscanf
    direttamente sull'elemento
45     // della lista e non su una variabile d'appoggio. Questo permette di non dover
    copiare ogni volta la struct
46     tail = tmp->prec;
47     free(tmp);
48

```

```

49 // Chiudo il file di input e apro il file di output
50 fclose(fp);
51 fp = fopen("data_es2_output", "w");
52
53 // Visita dalla coda. Scrivo su file e libero memoria. Poi termino
54 while (tail != NULL)
55 {
56     fprintf(fp, "%d %f %6s\n", tail->n, tail->f, tail->s);
57     tmp = tail->prec;
58     free(tail);
59     tail = tmp;
60 }
61 return 0;
62 }
63
64 // Implementazione lista semplice
65 /*int main()
66 {
67     data *tmp, *head = NULL;
68     FILE *fp;
69
70     fp = fopen("data_es2_input", "r");
71     tmp = (data *)malloc(sizeof( data));
72     while (fscanf(fp, " %d %f %6s", &(tmp->n), &(tmp->f), tmp->s) != EOF)
73     {
74         tmp->next = head;
75         head = tmp;
76         tmp = (data *)malloc(sizeof(data));
77     }
78     free(tmp);
79     fclose(fp);
80
81     fp = fopen("data_es2_output", "w");
82     while (head != NULL)
83     {
84         fprintf(fp, " %d %f %6s\n", head->n, head->f, head->s);
85         tmp = head;
86         free(head);
87         head = tmp->next;
88     }
89     fclose(fp);
90
91     free(head);
92     return 0;
93 }*/
94

```