

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 // struct per memorizzare ogni riga
5 struct data
6 {
7     int n;
8     float f;
9     char s[6 + 1]; // ogni stringa deve avere spazio per \0
10 };
11
12 int main()
13 {
14     // tmp struttura di supporto. vett puntatore a vettore di strutture Data
15     // i variabile di ciclo, N numero di righe lette, size capacità del vettore
16     // fp puntatore al file, prima di input e poi di output
17     struct data tmp, *vett = NULL;
18     int i, N = 0, size = 1;
19     FILE *fp;
20
21     // Apro il file di input in lettura
22     fp = fopen("data_es2_input", "r");
23     // Memorizzo la riga sulla variabile temporanea fino alla fine del file
24     while (fscanf(fp, " %d %f %6s", &(tmp.n), &(tmp.f), tmp.s) != EOF)
25     {
26         // Controllo se necessaria duplicazione della capacità vettore
27         if (size == N + 1)
28         {
29             size *= 2;
30             vett = (struct data *)realloc(vett, size * sizeof(struct data));
31         }
32         // Copio la variabile temporanea nel vettore ed incremento il numero di righe
33         lette
34         vett[N] = tmp;
35         N++;
36     }
37     // Chiudo il file di input e apro il file di output in scrittura
38     fclose(fp);
39     fp = fopen("data_es2_output", "w");
40
41     // Scrivo il vettore su disco iterando in senso inverso
42     for (i = N - 1; i >= 0; i--)
43         fprintf(fp, " %d %f %6s\n", vett[i].n, vett[i].f, vett[i].s);
44
45     // Chiudo il file di output. Libero memoria e termino
46     fclose(fp);
47     free(vett);
48     return 0;
49 }

```