```c
#include <stdio.h>
#include <stdlib.h>

// struct per memorizzare ogni riga
struct data
{
    int n;
    float f;
    char s[6 + 1]; // ogni stringa deve avere spazio per \0
};

int main()
{
    // tmp struttura di supporto. vett puntatore a vettore di strutture Data
    // i variabile di ciclo, N numero di righe lette, size capacità del vettore
    // fp puntatore al file, prima di input e poi di output
    struct data tmp, *vett = NULL;
    int i, N = 0, size = 1;
    FILE *fp;

    // Apro il file di input in lettura
    fp = fopen("data_es2_input", "r");
    if (fp == NULL)
        return (EXIT_FAILURE);
    // Memorizzo la riga sulla variabile temporanea fino alla fine del file
    while (fscanf(fp, " %d %f %6s", &(tmp.n), &(tmp.f), tmp.s) != EOF)
    {
        // Controllo se necessaria duplicazione della capacità vettore
        if (size == N + 1)
        {
            size *= 2;
            vett = (struct data *)realloc(vett, size * sizeof(struct data));
        }
        // Copio la variabile temporanea nel vettore ed incremento il numero di righe
lette
        vett[N] = tmp;
        N++;
    }
    // Chiudo il file di input e aptro il file di output in scrittura
    fclose(fp);
    fp = fopen("data_es2_output", "w");
    if (fp == NULL)
        return (EXIT_FAILURE);
    // Scrivo il vettore su disco iterando in senso inverso
    for (i = N - 1; i >= 0; i--)
        fprintf(fp, " %d %f %6s\n", vett[i].n, vett[i].f, vett[i].s);

    // Chiudo il file di output. Libero memoria e termino
    fclose(fp);
    free(vett);
    return 0;
}
```

52