

```
c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#define LINE_BUF 1024

void usage(const char *prog) {
    printf("Usage: %s input.md output.html\n", prog);
}

int main(int argc, char *argv[]) {
    if (argc < 3) {
        usage(argv[0]);
        return 1;
    }
    FILE *in = fopen(argv[1], "r");
    if (!in) {
        printf("Failed to open input.\n");
        return 1;
    }
    FILE *out = fopen(argv[2], "w");
    if (!out) {
        printf("Failed to open output.\n");
        fclose(in);
        return 1;
    }
    fprintf(out, "<html><body>\n");
    char line[LINE_BUF];
    bool in_list = false;
    while (fgets(line, sizeof(line), in)) {
        size_t len = strlen(line);
        while (len > 0 && (line[len-1] == '\n' || line[len-1] == '\r')) {
            line[--len] = '\0';
        }
        if (len == 0) {
            if (in_list) {
                fprintf(out, "</ul>\n");
                in_list = false;
            }
            continue;
        }
        if (strncmp(line, "# ", 2) == 0) {
            if (in_list) { fprintf(out, "</ul>\n"); in_list = false; }
            fprintf(out, "<h1>%s</h1>\n", line + 2);
        } else if (strncmp(line, "## ", 3) == 0) {
            if (in_list) { fprintf(out, "</ul>\n"); in_list = false; }
            fprintf(out, "<h2>%s</h2>\n", line + 3);
        } else if (strncmp(line, "### ", 4) == 0) {
            if (in_list) { fprintf(out, "</ul>\n"); in_list = false; }
            fprintf(out, "<h3>%s</h3>\n", line + 4);
        } else if (line[0] == '-' || line[0] == '*') {
            const char *text = line + 1;
            while (*text == ' ' || *text == '\t') text++;
            if (!in_list) {
                fprintf(out, "<ul>\n");
                in_list = true;
            }
            fprintf(out, "<li>%s</li>\n", text);
        } else {
            if (in_list) { fprintf(out, "</ul>\n"); in_list = false; }
            fprintf(out, "<p>%s</p>\n", line);
        }
    }
    if (in_list) fprintf(out, "</ul>\n");
    fprintf(out, "</body></html>\n");
    fclose(in);
    fclose(out);
    return 0;
}
```