c
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_TITLE 100
#define MAX_CONTENT 1000
#define DATA_FILE "memos.txt"
typedef struct {
int id;
char title[MAX_TITLE];
char content[MAX_CONTENT];
} Memo;
void print_menu() {
printf("\n==== Simple Text Notepad ====\n");
printf("1. List memos\n");
printf("2. Add memo\n");
printf("3. View memo\n");
printf("4. Delete memo\n");
printf("0. Exit\n");
printf("Select: ");
}
void list_memos() {
FILE *fp = fopen(DATA_FILE, "r");
if (!fp) {
printf("No memos yet.\n");
return;
}
Memo m;
int found = 0;
printf("\n[Memo List]\n");
while (fread(&m, sizeof(Memo), 1, fp) == 1) {
printf("ID: %d | Title: %s\n", m.id, m.title);
found = 1;
}
if (!found) {
printf("No memos saved.\n");
}
fclose(fp);
}
int get_next_id() {
FILE *fp = fopen(DATA_FILE, "r");
if (!fp) return 1;
Memo m;
int max_id = 0;
while (fread(&m, sizeof(Memo), 1, fp) == 1) {
if (m.id > max_id) max_id = m.id;
}
fclose(fp);
return max_id + 1;
}
void add_memo() {
Memo m;
m.id = get_next_id();
printf("Title: ");
fgets(m.title, MAX_TITLE, stdin);
if (m.title[strlen(m.title) - 1] == '\n')
m.title[strlen(m.title) - 1] = '\0';
printf("Content (end with a single '.' on a line):\n");
char line[256];
m.content[0] = '\0';
while (1) {
if (!fgets(line, sizeof(line), stdin)) break;
if (strcmp(line, ".\n") == 0 || strcmp(line, ".\r\n") == 0) break;
if (strlen(m.content) + strlen(line) < MAX_CONTENT - 1) {
strcat(m.content, line);
} else {
printf("Content too long, truncating.\n");
break;
}
}
FILE *fp = fopen(DATA_FILE, "ab");
if (!fp) {
printf("Failed to open data file.\n");
return;
}
fwrite(&m, sizeof(Memo), 1, fp);
fclose(fp);
printf("Memo saved with ID %d.\n", m.id);
}
void view_memo() {
int id;
printf("Enter memo ID: ");
if (scanf("%d", &id) != 1) {
printf("Invalid input.\n");
while (getchar() != '\n');
return;
}
while (getchar() != '\n');
FILE *fp = fopen(DATA_FILE, "r");
if (!fp) {
printf("No memos.\n");
return;
}
Memo m;
int found = 0;
while (fread(&m, sizeof(Memo), 1, fp) == 1) {
if (m.id == id) {
printf("\n[Memo %d]\nTitle: %s\nContent:\n%s\n", m.id, m.title, m.content);
found = 1;
break;
}
}
if (!found) printf("Memo not found.\n");
fclose(fp);
}
void delete_memo() {
int id;
printf("Enter memo ID to delete: ");
if (scanf("%d", &id) != 1) {
printf("Invalid input.\n");
while (getchar() != '\n');
return;
}
while (getchar() != '\n');
FILE *fp = fopen(DATA_FILE, "r");
if (!fp) {
printf("No memos.\n");
return;
}
FILE *temp = fopen("memos_tmp.txt", "wb");
if (!temp) {
printf("Failed to open temp file.\n");
fclose(fp);
return;
}
Memo m;
int deleted = 0;
while (fread(&m, sizeof(Memo), 1, fp) == 1) {
if (m.id == id) {
deleted = 1;
continue;
}
fwrite(&m, sizeof(Memo), 1, temp);
}
fclose(fp);
fclose(temp);
remove(DATA_FILE);
rename("memos_tmp.txt", DATA_FILE);
if (deleted) printf("Deleted memo %d.\n", id);
else printf("Memo not found.\n");
}
int main() {
int choice;
while (1) {
print_menu();
if (scanf("%d", &choice) != 1) {
printf("Invalid input.\n");
while (getchar() != '\n');
continue;
}
while (getchar() != '\n'); // clear buffer
switch (choice) {
case 1: list_memos(); break;
case 2: add_memo(); break;
case 3: view_memo(); break;
case 4: delete_memo(); break;
case 0: printf("Bye.\n"); return 0;
default: printf("Unknown menu.\n");
}
}
return 0;
}
```