```c
/* lab2.c
 * Christopher Brant
 * cbrant
 * ECE 2230
 * Section 001
 * Spring 2017
 * Programming Assignment #2
 * Due on 2/15/17 at 11:59 PM
 * Professor Walt Ligon
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "list.h"
#include "guitar.h"
#include "guitardb.h"


#define MAXLINE 50
#define MAXCOMSIZE 8

int main(int argc, char *argv[])
{
	char line[MAXLINE];                                          //
Input buffer
	char command[MAXCOMSIZE];                                  // Command
string
	int cont = 1;                                                //
Continue variable
	int reterr, keylook;                                       // The retu
rn error variable
	guitar_t match, new_guitar;

	// The following variable is the database pointer
	guitardb_t dbpoint = guitardb_init();

	printf("\nCommands:\n> ADD\n> DELETE key-data\n> LOOKUP key-data\n");
	printf("> LIST\n> HELP \n> QUIT\n\n");

	/* Continue to wait for input and execute commands until
	   the QUIT command is entered */
	while (cont == 1)
	{
		keylook = 0;
		reterr = 0;
		fgets(line, sizeof(line), stdin);       // Gather user input from t
he keyboard
		sscanf(line, "%s", command);

		// Conditions for the ADD command and follow up questions
		if (strcmp(command, "ADD") == 0)
		{
			new_guitar = guitar_init();
			reterr = guitardb_add(dbpoint, new_guitar);

			if (reterr == 1)
				printf("\nData added.\n\n");
			else
				printf("Error in adding data.\n");
		}

		// Conditions for the LOOKUP command and follow up print statements
		else if (strcmp(command, "LOOKUP") == 0)
		{
			while (keylook < 1)
			{
				printf("\nEnter item key to search for, greater tha
n 0: ");
				fgets(line, sizeof(line), stdin);
				sscanf(line, "%d", &keylook);
				printf("\n");
			}

			match = guitardb_lookup(dbpoint, keylook);

			if (match == NULL)
				printf("\nData not found with item key: %d\n\n", ke
ylook);
			else
				guitar_print(match);
		}

		// Conditions for the DELETE command and follow up print statements
		else if (strcmp(command, "DELETE") == 0)
		{
			while (keylook < 1)
			{
				printf("Enter item key to find and delete, greater
than 0: ");
				fgets(line, sizeof(line), stdin);
				sscanf(line, "%d", &keylook);
			}

			match = guitardb_delete(dbpoint, keylook);

			if (match != NULL)
			{
				free(match);
				printf("\nData deleted.\n\n");
			}
			else
				printf("\nData not found.\n\n");
		}

		// Conditions for the LIST command and continual print statements
		else if (strcmp(command, "LIST") == 0)
			guitardb_report(dbpoint);

		// Conditions for the HELP command and corresponding print statemen
ts
		else if (strcmp(command, "HELP") == 0)
		{
			printf("\nCommands:\n> ADD\n> DELETE key-data\n> LOOKUP key
-data\n");
			printf("> LIST\n> HELP \n> QUIT\n\n");
		}

		// Conditions for the QUIT command and the possible errors
		else if (strcmp(command, "QUIT") == 0)
		{
			cont = 0;

			guitardb_finalize(dbpoint);
		}
		// Conditions for incorrect inputs
		else
		{
			printf("Incorrect command. Try again or type 'QUIT' to end
all processes.\n");
			printf("Remember no spaces after single word commands.\n\n"
);
		}
```

```
                command[0] = '\0';
        }

        return 0;
}
```