```c
#include <stdlib.h>
#include <stdio.h>
#define MAX_NAME_LENGTH 100
#define MAX_DATA_BASE_LENGTH 100

// Declaration of person struct
typedef struct person
{
        char firstName[MAX_NAME_LENGTH];
        char lastName[MAX_NAME_LENGTH];
        unsigned int phoneNumber;
} Person;

// Function declarations
void loadScvFile(Person *dataBase[], char *fileName);
void addPersonToCsvFile(char *filename);
void findPersonFromPhoneNumber(char *fileName);

int main() {
        char fileName[MAX_NAME_LENGTH];
        Person *dataBase[MAX_DATA_BASE_LENGTH];

        puts("**********************");
        puts("Welcome to Assignment 9");
        puts("**********************");

        // Get file name from user
        printf("\nEnter data base name: ");
        scanf("%s",fileName);

        // Load SCV file into database
        loadScvFile(dataBase, fileName);

        while(1) {
                int controlVar = 0;

                printf("\n1: Add person to %s.\n2: Search %s on Phone Number.\n3: Close Program.\nPlease choose option: ", fileName, fileName);
                scanf("%1d", &controlVar);
                puts("");

                if (controlVar==1)
                {
                        addPersonToCsvFile(fileName);
                }

                if (controlVar == 2) {
                        findPersonFromPhoneNumber(fileName);
                }

                if (controlVar == 3)
                {
                        break;
                }
        }

        puts("Program closing...");
        return 0;
}

void loadScvFile (Person *dataBase[], char *fileName) {
```

```c
        // Open file
        FILE *fPtr = fopen(fileName, "r+");
        // Creat new file is file does not exists.
        if (fPtr == NULL)
        {
                printf("Creating SCV file %s...\n", fileName);
                fPtr = fopen(fileName, "w+");
        }

        // Load SCV file into database
        char *line = NULL;
        size_t len = 0;
        size_t counter = 0;

        while ((getline(&line, &len, fPtr)) != -1)
        {
                Person person;
                sscanf(line, "%[^,]%*[, ]%[^,]%*[, ]%u", person.firstName, person.lastName, &person.phoneNumber);
                dataBase[counter] = &person;
                counter++;
        }

        // Close file
        free(line);
        fclose(fPtr);
}

void addPersonToCsvFile (char *filename) {
        // Open file in append mode
        FILE *fPtr = fopen(filename, "a+");

        // Creat Person struct
        Person person;

        // Get first name from user.
        printf("Enter First Name: ");
        scanf("%s", person.firstName );

        // Get last name from user.
        printf("Enter Last Name: ");
        scanf("%s", person.lastName );

        // Get phone number from user.
        printf("Enter Phone Number: ");
        scanf("%d", &person.phoneNumber);

        // Append person to SCV file.
        fprintf(fPtr, "\n%s, %s, %d", person.firstName, person.lastName, person.phoneNumber);
        fclose(fPtr);
}

void findPersonFromPhoneNumber (char *fileName) {
        // Get phone number to search for from user.
        unsigned int phoneNumber;
        printf("Enter Phone Number To Search For: ");
        scanf("%d", &phoneNumber);

        // open file
        FILE *fPtr = fopen(fileName, "r");

```

```c
        // Search for phoneNumber
        char *line = NULL;
        size_t len = 0;
        char match = 0;
        Person person;

        while ((getline(&line, &len, fPtr)) != -1)
        {
                sscanf(line, "%[^,]%*[, ]%[^,]%*[, ]%u", person.firstName, person.lastName, &person.phoneNumber);
                if (phoneNumber == person.phoneNumber) {
                        puts("Match found:");
                        printf("%s, %s, %d\n", person.firstName, person.lastName, person.phoneNumber);
                        match = 1;
                        break;
                }
        }
        if (match == 0) {
                puts("No match found.");
        }

        // Close file
        free(line);
        fclose(fPtr);
}
```