```
Script started on Thu 11 Oct 2012 08:16:09 AM CDT
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ C
PP --version
This is CPP version 1.219 executing under perl v5.12.4 and compiling with:

g++ (Ubuntu/Linaro 4.6.1-9ubuntu3) 4.6.1
Copyright (C) 2011 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.


\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ p
wd
/home/georgia/cplusplus
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at rolodex.info
********************************************************************************

Name: Jakob Hansen
Class: CSC122
Project: Rolodex
Level: 4
    Total Attempted: 4 (more to come later hopefully!)

Program description:
    A program that stores name, address, and phone # information about people.
    Users may add, delete, search, edit, or print the information stored in the
    rolodex. Users may edit one or all of the attributes about a given
    "person" in the rolodex.

********************************************************************************
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at rolodex.cpp
#include <iostream>
#include <cctype>
#include <climits>
#include <cstring>
#include "person.h"
#include "strextra.h"

const short MAX_RECORDS = 10;

using namespace std;

// prints all entries containing stuff, with labeling
void print_all(person persarr[], short size = MAX_RECORDS);

/* prints entries of persarr[] that contain searchstr[] in the member data
specified by element */
void search(person persarr[], char searchstr[], char element);

// returns the first empty index in persarr[] or MAX_RECORDS if full.
short first_empty(person persarr[], short size = MAX_RECORDS);

int main(void)
{
    short index, recordid, first;
    person rolodex[MAX_RECORDS];
    person entry;
    char yesno, choice;
    char data[20], searchstr[20];

    cout << "\n\tWelcome to the Rolodex thing!";

    do
    {
        yesno = 'n';
        cout << "\nHow would you like to proceed?"
             << "\n1. Add Entry"
             << "\n2. Edit Entry"
             << "\n3. Delete Entry"
             << "\n4. Find an Entry."
             << "\n5. Print all entries."
             << "\n6. (Q)uit\n";

        cin >> choice;
        cin.ignore(INT_MAX, '\n');

        if ((toupper(choice) == '1' || toupper(choice) == 'A')
         && (first_empty(rolodex) < MAX_RECORDS))
        {


            cout << "\nPlease enter the name: ";
            get_line(data, MAX_NAME);
            entry.set_name(data);

            cout << "\nPlease enter the street: ";
            get_line(data, MAX_STREET);
            entry.set_street(data);


            cout << "\nPlease enter the city: ";
            get_line(data, MAX_CITY);
            entry.set_city(data);

            cout << "\nPlease enter the state: ";
            get_line(data, MAX_STATE);
            entry.set_state(data);

            cout << "\nPlease enter the phone: ";
            get_line(data, MAX_PHONE);
            entry.set_phone(data);

            rolodex[first_empty(rolodex)] = entry; // if so fill it up

            cout << "\nAdding record......successful!";
        }
        else if ((toupper(choice) == '1' || toupper(choice) == 'A')
         && (first_empty(rolodex) == MAX_RECORDS))
        {
            cout << "\nThe Rolodex is full! You'll have to make room!\n";
        }
        else if (toupper(choice) == '2' || toupper(choice) == 'E')
        {
            index = 0;
            print_all(rolodex);

            cout << "\nPlease enter the ID of the record you'd like to edit:  ";

            cin >> recordid;

            while (recordid >= MAX_RECORDS || rolodex[recordid].is_empty())
            {
                cout << "\nThat's a totally invalid ID. Try again:  ";
                cin >> recordid;
            }

            cout << "\nAlright, would you like to enter a single part of an "
                 << "entry or the entire thing? Choose from the menu.\n"
                 << "\n1. Edit name."
                 << "\n2. Edit street."
                 << "\n3. Edit city."
                 << "\n4. Edit state."
                 << "\n5. Edit phone."
                 << "\n6. Edit entire entry."
                 << "\n7. Go Back to Main Menu.";
```

```
        cin >> choice;

        if (choice == '1')
        {
            cout << "\nYou have chosen to edit the name. The name of the "
                 << "current entry is ";
            rolodex[recordid].get_name(data);
            cout << data << " \n";
            cout << "\nPlease enter the new name: ";
            get_line(data, MAX_NAME);
            rolodex[recordid].set_name(data);
        }
        else if (choice == '2')
        {
            cout << "\nYou have chosen to edit the street. The street of the "
                 << "current entry is ";
            rolodex[recordid].get_street(data);
            cout << data << " \n";
            cout << "\nPlease enter the new street: ";
            get_line(data, MAX_STREET);
            rolodex[recordid].set_street(data);
        }
        else if (choice == '3')
        {
            cout << "\nYou have chosen to edit the city. The city of the "
                 << "current entry is ";
            rolodex[recordid].get_city(data);
            cout << data << " \n";
            cout << "\nPlease enter the new city: ";
            get_line(data, MAX_CITY);
            rolodex[recordid].set_city(data);
        }
        else if (choice == '4')
        {
            cout << "\nYou have chosen to edit the state. The state of the "
                 << "current entry is ";
            rolodex[recordid].get_state(data);
            cout << data << " \n";
            cout << "\nPlease enter the new state: ";
            get_line(data, MAX_STATE);
            rolodex[recordid].set_state(data);
        }
        else if (choice == '5')
        {
            cout << "\nYou have chosen to edit the phone. The phone of the "
                 << "current entry is ";
            rolodex[recordid].get_phone(data);
            cout << data << " \n";
            cout << "\nPlease enter the new state: ";
            get_line(data, MAX_PHONE);
            rolodex[recordid].set_phone(data);
        }
        else if (choice == '6')
        {
            cout << "\nPlease enter the name: ";
            get_line(data, MAX_NAME);
            entry.set_name(data);

            cout << "\nPlease enter the street: ";
            get_line(data, MAX_STREET);
            entry.set_street(data);


            cout << "\nPlease enter the city: ";
            get_line(data, MAX_CITY);
            entry.set_city(data);

            cout << "\nPlease enter the state: ";
            get_line(data, MAX_STATE);
            entry.set_state(data);

            cout << "\nPlease enter the phone: ";
            get_line(data, MAX_PHONE);
            entry.set_phone(data);

            rolodex[recordid] = entry;
        }
        else if (choice != '7')
        {
            cout << "\nEditing record.....successful!";
        }
        else
        {
            cout << "\nReturning to main menu!";
        }

    }
    else if (toupper(choice) == '3' || toupper(choice) == 'D')
    {
        index = 0;
        // check for the first empty spot so we can go back and clean up
        first = first_empty(rolodex);
        print_all(rolodex);

        cout << "\nPlease enter the ID of the record to delete:  ";

        cin >> recordid;


        // check for valid id
        while (recordid >= MAX_RECORDS || rolodex[recordid].is_empty())
        {
            cout << "\nThat's a totally invalid ID. Try again:  ";
            cin >> recordid;
        }

        // empty out that spot with a default construction
        rolodex[recordid] = person();

        // move all the other doggies down cause our first_empty() depends
        // on the empty spots being at the end

        for (index = recordid+1; index < MAX_RECORDS; index++)
        {
            rolodex[index-1] = rolodex[index];
        }

        // make sure all the spots after the good data are clear.

        for (index = first-1; index < MAX_RECORDS; index++)
        {
            rolodex[index] = person();
        }

        cout << "\nDeleting....Complete!";

    }
    else if (toupper(choice) == '4' || toupper(choice) == 'F')
    {
        cout << "\nYou've chosen to find an entry!"
             << "\nPlease Choose from the list:"
             << "\n\n1.Find by (n)ame."
             << "\n2.Find by (s)treet."
             << "\n3.Find by (c)ity."
             << "\n4.Find by s(t)ate."
             << "\n5.Find by (p)hone.\n";
        cin >> choice;
```

```
            if (toupper(choice) == '1' || toupper(choice) == 'N')
            {
                cout << "\nEnter the name to search for:  ";
                get_line(searchstr, MAX_NAME);
                cout << "\nOk, searching for " << searchstr << '\n';
                search(rolodex, searchstr, choice);
            }
            else if (toupper(choice) == '2' || toupper(choice) == 'S')
            {
                cout << "\nEnter the street to search for:  ";
                get_line(searchstr, MAX_STREET);
                cout << "\nOk, searching for " << searchstr << '\n';
                search(rolodex, searchstr, choice);
            }
            else if (toupper(choice) == '3' || toupper(choice) == 'C')
            {
                cout << "\nEnter the city to search for:  ";
                get_line(searchstr, MAX_CITY);
                cout << "\nOk, searching for " << searchstr << '\n';
                search(rolodex, searchstr, choice);
            }
            else if (toupper(choice) == '4' || toupper(choice) == 'T')
            {
                cout << "\nEnter the state to search for:  ";
                get_line(searchstr, MAX_STATE);
                cout << "\nOk, searching for " << searchstr << '\n';
                search(rolodex, searchstr, choice);
            }
            else if (toupper(choice) == '5' || toupper(choice) == 'P')
            {
                cout << "\nEnter the phone to search for:  ";
                get_line(searchstr, MAX_PHONE);
                cout << "\nOk, searching for " << searchstr << '\n';
                search(rolodex, searchstr, choice);
            }
            else
            {
                cout << "\nAborting Search!\n";
            }
        }
        else if (toupper(choice) == '5' || toupper(choice) == 'P')
        {
            print_all(rolodex);
        }

        cout << "\nWould you like to continue?  ";
        cin >> yesno;
    } while (toupper(yesno) == 'Y');

    cout << "\nGoodbye!\n\n";

    return 0;
}

void print_all(person persarr[], short size)
{
    short index = 0;

    cout << '|' << left << setw(2)  << "ID"
         << '|' << setw(17) << "Name"
         << '|' << setw(18) << "Street"
         << '|' << setw(18) << "City"
         << '|' << setw(5)  << "State"
         << '|' << setw(13) << "Phone"
         << "|\n"
         << "----------------------------------------"
         << "----------------------------------------\n";

    while (!persarr[index].is_empty() && index < size) // print all the records
    {
        cout << '|' << setw(2)  << index;
        persarr[index].print();
        index++;
    }

    return;
}


void search(person persarr[], char searchstr[], char element)
{
    short index, pos=0;
    char data[20];
    for (index = 0; index < MAX_RECORDS; index++)
    {
        if (toupper(element) == '1' || toupper(element) == 'N')
        {
            persarr[index].get_name(data);
        }
        else if (toupper(element) == '2' || toupper(element) == 'S')
        {
            persarr[index].get_street(data);
        }
        else if (toupper(element) == '3' || toupper(element) == 'C')
        {
            persarr[index].get_city(data);
        }
        else if (toupper(element) == '4' || toupper(element) == 'T')
        {
            persarr[index].get_state(data);
        }
        else if (toupper(element) == '5' || toupper(element) == 'P')
        {
            persarr[index].get_phone(data);
        }

        if (find(data, searchstr, pos) != -1)
        {
            //cout << "found match:\n";
            persarr[index].print();
        }
        else
        {
            //cout << "Didn't find a match :(\n";
        }
    }

    return;
}

short first_empty(person persarr[], short size)
{
    short count = 0;

    while (count < size && !persarr[count].is_empty())
    {
        count++;
    }

    return count;
}



\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at person.h
#ifndef PERSON_H_INC
#define PERSON_H_INC
```

```
#include <iostream>
#include <cctype>
#include <cstring>
#include <iomanip>

using namespace std;

const long MAX_NAME = 18,
               MAX_STREET = 18,
               MAX_CITY = 18,
               MAX_STATE = 3,
               MAX_PHONE = 14;

class person
{
    char name[MAX_NAME],
       street[MAX_STREET],
         city[MAX_CITY],
        state[MAX_STATE],
        phone[MAX_PHONE];

public:
    // constructors
    person(void);
    person(const char name[], const char street[], const char city[],
           const char state[], const char phone[]);
    person(const person & p);

    // printing & reading
    void print(void) const;
    bool read(void);

    // accessors
    void get_name(char newname[], const short len = 0) const;
    void get_street(char newstreet[], const short len = 0) const;
    void get_city(char newcity[], const short len = 0) const;
    void get_state(char newstate[], const short len = 0) const;
    void get_phone(char newphone[], const short len = 0) const;


    // mutators
    bool set_name(const char newname[]);
    bool set_street(const char newstreet[]);
    bool set_city(const char newcity[]);
    bool set_state(const char newstate[]);
    bool set_phone(const char newphone[]);
    bool is_empty(void);
};

#endif



\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at person.cpp
#include <iostream>
#include <cctype>
#include <cstring>
#include <iomanip>
#include "strextra.h"
#include "person.h"



using namespace std;

    // constructors

    person::person(void)
    {
        name[0] = '\0';
        street[0] = '\0';
        city[0] = '\0';
        state[0] = '\0';
        phone[0] = '\0';
    }

    person::person(const char name[], const char street[], const char city[],
                                const char state[], const char phone[])
    {
        set_name(name);
        set_street(street);
        set_city(city);
        set_state(state);
        set_phone(phone);
    }

    person::person(const person & p)
    {
        strcpy(name, p.name);
        strcpy(street, p.street);
        strcpy(city, p.city);
        strcpy(state, p.state);
        strcpy(phone, p.phone);
    }

    // printing & reading
    void person::print(void) const
    {
        cout << '|' << setw(17) << name
             << '|' << setw(18) << street
             << '|' << setw(18) << city
             << '|' << setw(5)  << state
             << '|' << setw(13) << phone
             << "|\n";
    }

    bool person::read(void)
    {
        bool success = !cin.fail();


        get_line(name, MAX_NAME);

        get_line(street, MAX_STREET);

        get_line(city, MAX_CITY);

        get_line(state, MAX_STATE);

        get_line(phone, MAX_PHONE);

        return success;
    }

    // accessors
    void person::get_name(char newname[], const short len) const
    {
        if (len > 0)
        {
            strncpy(newname, name, len-1);
            newname[len-1] = '\0';
        }
        else
        {
            strcpy(newname, name);
        }
```

```
        return;
    }

    void person::get_street(char newstreet[], const short len) const
    {
        if (len > 0)
        {
            strncpy(newstreet, street, len-1);
            newstreet[len-1] = '\0';
        }
        else
        {
            strcpy(newstreet, street);
        }
        return;
    }

    void person::get_city(char newcity[], const short len) const
    {
        if (len > 0)
        {
            strncpy(newcity, city, len-1);
            newcity[len-1] = '\0';
        }
        else
        {
            strcpy(newcity, city);
        }
        return;
    }

    void person::get_state(char newstate[], const short len) const
    {
        if (len > 0)
        {
            strncpy(newstate, state, len-1);
            newstate[len-1] = '\0';
        }
        else
        {
            strcpy(newstate, state);
        }
        return;
    }
    void person::get_phone(char newphone[], const short len) const
    {
        if (len > 0)
        {
            strncpy(newphone, phone, len-1);
            newphone[len-1] = '\0';
        }
        else
        {
            strcpy(newphone, phone);
        }
        return;
    }

    //mutators

    bool person::set_name(const char newname[])
    {
        strncpy(name, newname, MAX_NAME-1);
        name[MAX_NAME-1] = '\0';
        return true;
    }
    bool person::set_street(const char newstreet[])
    {
        strncpy(street, newstreet, MAX_STREET-1);
        street[MAX_STREET-1] = '\0';
        return true;
    }
    bool person::set_city(const char newcity[])
    {
        strncpy(city, newcity, MAX_CITY-1);
        city[MAX_CITY-1] = '\0';
        return true;
    }
    bool person::set_state(const char newstate[])
    {
        strncpy(state, newstate, MAX_STATE-1);
        state[MAX_STATE-1] = '\0';
        return true;
    }
    bool person::set_phone(const char newphone[])
    {
        strncpy(phone, newphone, MAX_PHONE-1);
        phone[MAX_PHONE-1] = '\0';
        return true;
    }


    bool person::is_empty(void)
    {
        return (name[0] == '\0' && street[0] == '\0' && city[0] == '\0' &&
            state[0] == '\0' && phone[0] == '\0');
    }
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at strextra.h
#ifndef STREXTRA_H_INC
#define STREXTRA_H_INC

#include <cstring>
#include <iostream>
#include <climits>

using namespace std;

// char version
short find(const char str[], char c, short index = 0);

// string version
short find(const char str[], char substr[], short index = 0,
                                        bool caseSense = false);


// nice input
void get_line(char s[], const long max);

#endif
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at strextra.cpp
#include "strextra.h"
#include <cstring>
#include <iostream>
#include <climits>

using namespace std;

// char
short find(const char str[], char c, short index)
{
    while (str[index] != c && str[index] != '\0')
    {
        index++;
    }
    if (str[index] == '\0')
    {
```

```
        index = -1;
    }
    return index;
}

// string
short find(const char str[], char substr[], short index, bool caseSense)
{
    short matchpos = 0, subindx = 0, wildcount = 0;
    bool beginmatch, endmatch;

    while (substr[subindx] != '\0' && str[index] != '\0' && matchpos != -1)
    {
        // found escape character
        if (substr[subindx] == '/' && substr[subindx-1] != '/')
        {
            subindx++;
        }
        // if the first char in the search string is * then beginmatch is true
        else if (!beginmatch && subindx == 0 && substr[subindx] == '*')
        {
            beginmatch = true;
            matchpos = 0;
        }
        // found unescaped '*' that's not the last char in search string
        else if (substr[subindx] == '*' && substr[subindx-1] != '/'
                                        && substr[subindx + 1] != '\0')
        {
            beginmatch = true;
            wildcount = index;
            // pause at '*' and look forward for next match
            while ((caseSense ? str[wildcount] : toupper(str[wildcount])) !=
                (caseSense ? substr[subindx+1] : toupper(substr[subindx+1])) &&
                                        str[wildcount] != '\0')
            {
                wildcount++;
            }
            // found next match
            if (str[wildcount] != '\0')
            {
                index = wildcount;
                subindx++;
            }
            // didn't find it - abort
            else
            {
                matchpos = -1;
            }
        }
        // found an unescaped '?' that's not the last char
        else if (substr[subindx] == '?' && substr[subindx - 1] != '/'
                                        && substr[subindx + 1] != '\0')
        {
               index++;
               subindx++;
        }
        // found a match, throw begin marker and advance to check next pair
        else if (!beginmatch &&
            (caseSense ? substr[subindx] : toupper(substr[subindx])) ==
            (caseSense ? str[index] : toupper(str[index])))
        {
            matchpos = index; //store the match
            beginmatch = true;
            index++;
            subindx++;
        }
        // in the middle of potential matching string, keep checking
        else if (beginmatch &&
            (caseSense ? substr[subindx] : toupper(substr[subindx])) ==
            (caseSense ? str[index] : toupper(str[index]))
                            && substr[subindx + 1] != '\0')
        {
            index++;
            subindx++;
        }
        // check if the last character matches too
        else if (beginmatch &&
            (caseSense ? substr[subindx] : toupper(substr[subindx])) ==
            (caseSense ? str[index] : toupper(str[index]))
                           && substr[subindx + 1] == '\0')
        {
            endmatch = true;
            index++;
            subindx++;
        }
        else if (substr[subindx] == '*' && substr[subindx + 1] == '\0')
        {
            endmatch = true;
            subindx++;
        }
        else if (beginmatch && substr[subindx + 1] == '\0'
                            && substr[subindx] == '?')
        {
            endmatch = true;
            index++;
            subindx++;
        }
        /* didn't find a match, keep going in "searched" string, start over in
        substring */
        else
        {
            index++;
            subindx = 0;
            beginmatch = false;
        }
    }
    if (!beginmatch || !endmatch)
    {
        matchpos = -1;
    }

    return matchpos;
}

//nice input
void get_line(char s[], const long max)
{
    cout.flush();
    if (cin.peek() == '\n')
    {
        cin.ignore();
    }
    cin.getline(s, max);
    if (cin.fail())
    {
        cin.clear();
        cin.ignore(INT_MAX, '\n');
    }
    return;
}
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ C
PP rolodex person strextra
person.cpp...
rolodex.cpp***
strextra.cpp...
rolodex.cpp: In function â\200\230int main()â\200\231:
rolodex.cpp:218:35: warning: conversion to â\200\230short intâ\200\231 from
â\200\230intâ\200\231 may alter its value [-Wconversion]
```

```
rolodex.cpp:225:32: warning: conversion to â\200\230short intâ\200\231 from
â\200\230intâ\200\231 may alter its value [-Wconversion]


\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ .
/rolodex.out

        Welcome to the Rolodex thing!
How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
1

Please enter the name: George Washington

Please enter the street: Adams St.

Please enter the city: Lexington

Please enter the state: VA

Please enter the phone: 430 178 9238

Adding record......successful!
Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
1

Please enter the name: John Adams

Please enter the street: Jefferson St.

Please enter the city: Boston

Please enter the state: MA

Please enter the phone: 359 247 2340

Adding record......successful!
Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
1

Please enter the name: Thomas Jefferson

Please enter the street: Burr St.

Please enter the city: Williamsburg

Please enter the state: PA

Please enter the phone: 235 24863468

Adding record......successful!
Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
1

Please enter the name: James Madison

Please enter the street: Clinton St.

Please enter the city: Saalem

Please enter the state: MA

Please enter the phone: 34683836 3456

Adding record......successful!
Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
1

Please enter the name: James Monroe

Please enter the street: Tompkins Ave.

Please enter the city: Trenton

Please enter the state: NJ

Please enter the phone: 6953349 3474

Adding record......successful!
Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
1

Please enter the name: John Q. Adams

Please enter the street: Calhoun Blvd

Please enter the city: Lexington

Please enter the state: MA

Please enter the phone: 249 34693596

Adding record......successful!
```

```
Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
1

Please enter the name: Andrew Jackkson

Please enter the street: Van Buren Dr.

Please enter the city: Yorktown VA

Please enter the state: VA

Please enter the phone: 3468346 3569

Adding record......successful!
Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
1

Please enter the name: Martin Van Buren

Please enter the street: Johnson Lane

Please enter the city: Charleston

Please enter the state: SC

Please enter the phone: 246834682457

Adding record......successful!
Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
1

Please enter the name: William Henty Harrison

Please enter the street: Tyler Ave.

Please enter the city: Bennington

Please enter the state: NY

Please enter the phone: 684 340 2587

Adding record......successful!
Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
1

Please enter the name: John Tyler

Please enter the street: Polk St.

Please enter the city: Monmouth

Please enter the state: NJ

Please enter the phone: 486 306 3868

Adding record......successful!
Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
1

The Rolodex is full! You'll have to make room!

Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
2
|ID|Name             |Street            |City              |State|Phone        |
--------------------------------------------------------------------------------
|0 |George Washington|Adams St.         |Lexington         |VA   |430 178 9238 |
|1 |John Adams       |Jefferson St.     |Boston            |MA   |359 247 2340 |
|2 |Thomas Jefferson |Burr St.          |Williamsburg      |PA   |235 248 3468 |
|3 |James Madison    |Clinton St.       |Salem             |MA   |346 346 3456 |
|4 |James Monroe     |Tompkins Ave.     |Trenton           |NJ   |695 349 3474 |
|5 |John Q. Adams    |Calhoun Blvd      |Lexington         |MA   |249 346 3596 |
|6 |Andrew Jackson   |Van Buren Dr.     |Yorktown          |VA   |346 346 3569 |
|7 |Martin Van Buren |Johnson Lane      |Charleston        |SC   |246 346 2457 |
|8 |William Henry Har|Tyler Ave.        |Bennington        |NY   |684 340 2587 |
|9 |John Tyler       |Polk St.          |Monmouth          |NJ   |486 306 3868 |

Please enter the ID of the record you'd like to edit:  5

Alright, would you like to enter a single part of an entry or the entire thing? Cho
ose from the menu.

1. Edit name.
2. Edit street.
3. Edit city.
4. Edit state.
5. Edit phone.
6. Edit entire entry.
7. Go Back to Main Menu.3

You have chosen to edit the city. The city of the current entry is Lexington
```

```
Please enter the new city: New Lexington

Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
p
|ID|Name             |Street            |City              |State|Phone        |
--------------------------------------------------------------------------------
|0 |George Washington|Adams St.         |Lexington         |VA   |430 178 9238 |
|1 |John Adams       |Jefferson St.     |Boston            |MA   |359 247 2340 |
|2 |Thomas Jefferson |Burr St.          |Williamsburg      |PA   |235 248 3468 |
|3 |James Madison    |Clinton St.       |Salem             |MA   |346 346 3456 |
|4 |James Monroe     |Tompkins Ave.     |Trenton           |NJ   |695 349 3474 |
|5 |John Q. Adams    |Calhoun Blvd      |New Lexington     |MA   |249 346 3596 |
|6 |Andrew Jackson   |Van Buren Dr.     |Yorktown          |VA   |346 346 3569 |
|7 |Martin Van Buren |Johnson Lane      |Charleston        |SC   |246 346 2457 |
|8 |William Henry Har|Tyler Ave.        |Bennington        |NY   |684 340 2587 |
|9 |John Tyler       |Polk St.          |Monmouth          |NJ   |486 306 3868 |

Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
3
|ID|Name             |Street            |City              |State|Phone        |
--------------------------------------------------------------------------------
|0 |George Washington|Adams St.         |Lexington         |VA   |430 178 9238 |
|1 |John Adams       |Jefferson St.     |Boston            |MA   |359 247 2340 |
|2 |Thomas Jefferson |Burr St.          |Williamsburg      |PA   |235 248 3468 |
|3 |James Madison    |Clinton St.       |Salem             |MA   |346 346 3456 |
|4 |James Monroe     |Tompkins Ave.     |Trenton           |NJ   |695 349 3474 |
|5 |John Q. Adams    |Calhoun Blvd      |New Lexington     |MA   |249 346 3596 |
|6 |Andrew Jackson   |Van Buren Dr.     |Yorktown          |VA   |346 346 3569 |
|7 |Martin Van Buren |Johnson Lane      |Charleston        |SC   |246 346 2457 |
|8 |William Henry Har|Tyler Ave.        |Bennington        |NY   |684 340 2587 |
|9 |John Tyler       |Polk St.          |Monmouth          |NJ   |486 306 3868 |

Please enter the ID of the record to delete:  7

Deleting....Complete!
Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
5
|ID|Name             |Street            |City              |State|Phone        |
--------------------------------------------------------------------------------
|0 |George Washington|Adams St.         |Lexington         |VA   |430 178 9238 |
|1 |John Adams       |Jefferson St.     |Boston            |MA   |359 247 2340 |
|2 |Thomas Jefferson |Burr St.          |Williamsburg      |PA   |235 248 3468 |
|3 |James Madison    |Clinton St.       |Salem             |MA   |346 346 3456 |
|4 |James Monroe     |Tompkins Ave.     |Trenton           |NJ   |695 349 3474 |
|5 |John Q. Adams    |Calhoun Blvd      |New Lexington     |MA   |249 346 3596 |
|6 |Andrew Jackson   |Van Buren Dr.     |Yorktown          |VA   |346 346 3569 |
|7 |William Henry Har|Tyler Ave.        |Bennington        |NY   |684 340 2587 |
|8 |John Tyler       |Polk St.          |Monmouth          |NJ   |486 306 3868 |

Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
f

You've chosen to find an entry!
Please Choose from the list:

1.Find by (n)ame.
2.Find by (s)treet.
3.Find by (c)ity.
4.Find by s(t)ate.
5.Find by (p)hone.
n

Enter the name to search for:  hen

Ok, searching for hen
|William Henry Har|Tyler Ave.        |Bennington        |NY   |684 340 2587 |

Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
f

You've chosen to find an entry!
Please Choose from the list:

1.Find by (n)ame.
2.Find by (s)treet.
3.Find by (c)ity.
4.Find by s(t)ate.
5.Find by (p)hone.
n

Enter the name to search for:  GEORGE

Ok, searching for GEORGE
|George Washington|Adams St.         |Lexington         |VA   |430 178 9238 |

Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
f

You've chosen to find an entry!
```

```
Please Choose from the list:

1.Find by (n)ame.
2.Find by (s)treet.
3.Find by (c)ity.
4.Find by s(t)ate.
5.Find by (p)hone.
s

Enter the street to search for:  POLK

Ok, searching for POLK
|John Tyler       |Polk St.          |Monmouth          |NJ   |486 306 3868 |

Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
f

You've chosen to find an entry!
Please Choose from the list:

1.Find by (n)ame.
2.Find by (s)treet.
3.Find by (c)ity.
4.Find by s(t)ate.
5.Find by (p)hone.
c

Enter the city to search for:  lexInG

Ok, searching for lexInG
|George Washington|Adams St.         |Lexington         |VA   |430 178 9238 |
|John Q. Adams    |Calhoun Blvd      |New Lexington     |MA   |249 346 3596 |

Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
4

You've chosen to find an entry!
Please Choose from the list:

1.Find by (n)ame.
2.Find by (s)treet.
3.Find by (c)ity.
4.Find by s(t)ate.
5.Find by (p)hone.
t

Enter the state to search for:  v

Ok, searching for v
|George Washington|Adams St.         |Lexington         |VA   |430 178 9238 |

Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
f

You've chosen to find an entry!
Please Choose from the list:

1.Find by (n)ame.
2.Find by (s)treet.
3.Find by (c)ity.
4.Find by s(t)ate.
5.Find by (p)hone.
t

Enter the state to search for:  Ny

Ok, searching for Ny
|William Henry Har|Tyler Ave.        |Bennington        |NY   |684 340 2587 |

Would you like to continue?  y

How would you like to proceed?
1. Add Entry
2. Edit Entry
3. Delete Entry
4. Find an Entry.
5. Print all entries.
6. (Q)uit
q

Would you like to continue?  6

Goodbye!

\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ e
xit
exit

Script done on Thu 11 Oct 2012 08:33:03 AM CDT
```