

Script started on 2022-03-07 17:22:21-06:00 [TERM="xterm" TTY="/dev/pts/14" COLUMN=

a_vitale7@ares:~\$ pwd

/home/students/a_vitale7

a_vitale7@ares:~\$ cat rolodex_entry.info

```
*****
*
* NAME: Antonino Vitale          CLASS: CSC122-W01      *
*
* Lab: Dont lose that number!    Level: 7              *
*
* Description:                   *
*
* This program allows the user to store, search using strextra *
* library, and delete names, addresses, phone numbers, email  *
* addresses, and zip codes.    *
*
*****/
```

a_vitale7@ares:~\$ cat rolodex_entry.cpp

```
#include <iostream>
```

```
#include <sstream>
```

```
#include <iomanip>
```

```
#include <limits>
```

```
#include <cmath>
```

```
#include <math.h>
```

```
#include <time.h>
```

```
#include <cstdlib>
```

```
#include <array>
```

```
#include <vector>
```

```
#include <string>
```

```
#include <stdlib.h>
```

```
#include <cstring>
```

```
#include "rolodex.h"
```

```
#include "strextra.h"
```

```
using namespace std;
```

```
void disp_mainMenu(void);
```

```
void disp_editMenu(void);
```

```
void disp_findMenu(void);
```

```
void disp_sortMenu(void);
```

```
void debug(void);
```

```
int randomizer(unsigned int min, unsigned int max);
```

```
string str_input;
```

```
unsigned int int_input;
```

```
int main()
```

```
{
```

```
    bool exit_main = false;
```

```
    vector<rolodex> rolodex_entries(0);
```

```
    cout << "\n\t\tRolodex class program" << endl; //program start statement
```

```
    while (!exit_main) {
```

```
        disp_mainMenu();
```

```
        cout << "\n\tChoice: ";
```

```
        cin >> str_input;
```

```
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
```

```
        switch (tolower(str_input[0])) {
```

```
            case '1': { //add rolodex entry
```

```
                rolodex new_entry;
```

```
                cout << "\nName: ";
```

```
                getline(cin, str_input);
```

```
                new_entry.set_name(str_input.c_str());
```

```

        cout << "Address: ";
        getline(cin, str_input);
        new_entry.set_address(str_input.c_str());
        cout << "Phone number: ";
        getline(cin, str_input);
        new_entry.set_phone_number(str_input.c_str());
        cout << "Email: ";
        getline(cin, str_input);
        new_entry.set_email(str_input.c_str());
        cout << "Zipcode: ";
        getline(cin, str_input);
        new_entry.set_zip(str_input.c_str());
        rolodex_entries.push_back(new_entry);
        cout << "\nNew Entry Added!" << endl;
        break;
    } case '2': { //edit entries
        bool exit_edit = false;
        while (!exit_edit) {
            cout << "\nType 0 to cancel.";
            cout << "\nWhich entry would you like to edit: ";
            cin >> int_input;
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            if (int_input <= 0) {
                exit_edit = true;
            }
            else if (int_input - 1 > rolodex_entries.size()) {
                cout << "Invalid Input.";

```

```

        }
        else {
            rolodex_entries[int_input - 1].output();
            cout << "\ntype exit to cancel.";
            cout << "\nIs this the rolodex you would like?";
            getline(cin, str_input);
            if (tolower(str_input[0]) == 'y') {
                unsigned int hold_pos = int_input - 1;
                while (!exit_edit) {
                    disp_editMenu();
                    cout << "\n\tChoice: ";
                    cin >> str_input;
                    cin.ignore(numeric_limits<streamsize>::max(), '\n');
                    switch (tolower(str_input[0])) {
                        case '1': {
                            cout << "\nNew name: ";
                            getline(cin, str_input);
                            rolodex_entries[hold_pos].set_name(str_input.c_str());
                            break;
                        } case '2': {
                            cout << "\nNew address: ";
                            getline(cin, str_input);
                            rolodex_entries[hold_pos].set_address(str_input.c_str());
                            break;
                        } case '3': {
                            cout << "\nNew phone number: ";
                            getline(cin, str_input);
                            rolodex_entries[hold_pos].set_phone_number(str_input.c_str());
                            break;
                        } case '4': {
                            cout << "\nNew email: ";
                            getline(cin, str_input);
                            rolodex_entries[hold_pos].set_email(str_input.c_str());
                            break;
                        } case '5': {
                            cout << "\nNew zipcode: ";
                            getline(cin, str_input);
                            rolodex_entries[hold_pos].set_zip(str_input.c_str());
                            break;
                        }
                    }
                }
            }
        }
    }
}

```

```

        rolodex_entries[hoi
        break;
    } case '4': {
        cout << "\nNew ema:
        getline(cin, str_i
        rolodex_entries[hoi
        break;
    } case '5': {
        cout << "\nNew zip
        getline(cin, str_i
        rolodex_entries[hoi
        break;
    } case '6': case 'e': case
        exit_edit = true;
        break;
    } default: {
        cout << "\nInvalid
        break;
    }
}

}

}

cout << endl;
break;
} case '3': { //delete entries

```

```

        cout << "\nType 0 to cancel.";
        cout << "\nWhich entry would you like to delete: ";
        cin >> int_input;
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        if (int_input > 0 || int_input <= rolodex_entries.size()) {
            rolodex_entries[int_input - 1].output();
            cout << "\nIs this the rolodex you would like to de
            getline(cin, str_input);
            if (tolower(str_input[0]) == 'y') {
                for (unsigned int i = int_input - 1; i < rolodex_entries.size(); i++)
                    rolodex_entries[i] = rolodex_entries[i + 1];
                rolodex_entries.pop_back();
            }
        }
        cout << endl;
        break;
    } case '4': { //find entries
        bool exit_find = false;
        while (!exit_find) {
            disp_findMenu();
            cout << "\n\tChoice: ";
            cin >> str_input;
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            switch (tolower(str_input[0])) {
                case '1': { //name search
                    cout << "\nEnter name to search for: ";

```

```

        getline(cin, str_input);

        for (unsigned int i = 0; i < rolodex_entries.size(); i++)
            if (find(rolodex_entries[i].get_name(), str_input))
                printf("\nRolodex %i has the name\n", i);

    }

    cout << endl;

    break;

} case '2': { //address search

    cout << "\nEnter address to search for: ";

    getline(cin, str_input);

    for (unsigned int i = 0; i < rolodex_entries.size(); i++)
        if (find(rolodex_entries[i].get_address(), str_input))
            printf("\nRolodex %i has the address\n", i);

    }

    cout << endl;

    break;

} case '3': { //phone number search

    cout << "\nEnter phone number to search for: ";

    getline(cin, str_input);

    for (unsigned int i = 0; i < rolodex_entries.size(); i++)
        if (find(rolodex_entries[i].get_phone_number(), str_input))
            printf("\nRolodex %i has the phone number\n", i);

    }

    cout << endl;

```

```

        break;

    } case '4': { //email search

        cout << "\nEnter email to search for: ";

        getline(cin, str_input);

        for (unsigned int i = 0; i < rolodex_entries.size(); i++)
            if (find(rolodex_entries[i].get_email(), str_input))
                printf("\nRolodex %i has the email\n", i);

        }

        cout << endl;

        break;

    } case '5': { //zip search

        cout << "\nEnter zip to search for: ";

        getline(cin, str_input);

        for (unsigned int i = 0; i < rolodex_entries.size(); i++)
            if (find(rolodex_entries[i].get_zip(), str_input))
                printf("\nRolodex %i has the zip\n", i);

        }

        cout << endl;

        break;

    } case '6': case 'e': case 'q': { //quit

        exit_find = true;

        break;

    } default: {

        cout << "\nInvalid Choice." << endl;

        break;

```

```

        }
    }

}

cout << "\nReturning to Main Menu." << endl;
break;
} case '5': {
    for (unsigned int i = 0; i < rolodex_entries.size(); i++) {
        printf("\n\tEntry %i: ", i + 1);
        rolodex_entries[i].output();
    }
    break;
} case '6': {
    //sorting rolodex vector goes here
    rolodex swap;
    disp_sortMenu();
    cout << "\n\tChoice: ";
    cin >> str_input;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    switch (tolower(str_input[0])) {
        case '1': { //name sort
            for (unsigned int i = 0; i < rolodex_entries.size(); i++) {
                int l = i - 1;
                swap = rolodex_entries[i];
                while(l-1 >= 0 && strcmp(str_input, rolodex_entries[l].name) < 0)
                    rolodex_entries[l + 1] = rolodex_entries[l];
                rolodex_entries[l] = swap;
                l--;
            }
        }
    }
}

```

```

        rolodex_entries[l + 1] = swap;
    }
    cout << "\n Sorted by name." << endl;
    break;
} case '2': { //address sort
    for (unsigned int i = 0; i < rolodex_entries.size(); i++) {
        int l = i - 1;
        swap = rolodex_entries[i];
        while (l - 1 >= 0 && strcmp(str_input, rolodex_entries[l].address) < 0)
            rolodex_entries[l + 1] = rolodex_entries[l];
        rolodex_entries[l] = swap;
        l--;
    }
    rolodex_entries[l + 1] = swap;
}
    cout << "\n Sorted by address." << endl;
    break;
} case '3': { //phone sort
    for (unsigned int i = 0; i < rolodex_entries.size(); i++) {
        int l = i - 1;
        swap = rolodex_entries[i];
        while (l - 1 >= 0 && strcmp(str_input, rolodex_entries[l].phone) < 0)
            rolodex_entries[l + 1] = rolodex_entries[l];
        rolodex_entries[l] = swap;
        l--;
    }
    rolodex_entries[l + 1] = swap;
}
    cout << "\n Sorted by phone number." << endl;
}

```

```

        break;
    } case '4': { //email sort
        for (unsigned int i = 0; i < rolodex_entries.size(); i++)
        {
            int l = i - 1;
            swap = rolodex_entries[i];
            while (l - 1 >= 0 && strComp_notCase(swap, rolodex_entries[l]) < 0)
            {
                rolodex_entries[l + 1] = rolodex_entries[l];
                l--;
            }
            rolodex_entries[l + 1] = swap;
        }
        cout << "\n Sorted by email." << endl;
        break;
    } case '5': { //zip sort
        for (unsigned int i = 0; i < rolodex_entries.size(); i++)
        {
            int l = i - 1;
            swap = rolodex_entries[i];
            while (l - 1 >= 0 && strComp_notCase(swap, rolodex_entries[l]) < 0)
            {
                rolodex_entries[l + 1] = rolodex_entries[l];
                l--;
            }
            rolodex_entries[l + 1] = swap;
        }
        cout << "\n Sorted by zip code." << endl;
        break;
    } default: {
        cout << "\nInvalid Choice." << endl;
    }
}

```

```

        break;
    }
}
break;
} case '7': case 'e': case 'q': {
    exit_main = true;
    break;
} case 'd': {
    debug();
    break;
} default: {
    cout << "\nInvalid Choice." << endl;
    break;
}

}

cout << "\nExiting program." << endl; //program end statement
return 0;
}

void disp_mainMenu() {
    cout << "\n\t\tMain Menu" << endl;
    cout << "\n\t1) Add entry";
    cout << "\n\t2) Edit entry";
    cout << "\n\t3) Delete entry";
    cout << "\n\t4) Find entry";
}

```

```

        cout << "\n\t5) Print all entries";
        cout << "\n\t6) Sort entries";
        cout << "\n\t7) Quit" << endl;
        return;
    }

void disp_editMenu() {
    cout << "\n\t\tEdit rolodex" << endl;
    cout << "\n\t1) Edit Name";
    cout << "\n\t2) Edit Address";
    cout << "\n\t3) Edit Phone number";
    cout << "\n\t4) Edit Email address";
    cout << "\n\t5) Edit Zip code";
    cout << "\n\t6) Quit" << endl;
    return;
}

void disp_findMenu() {
    cout << "\n\t\tFind rolodex" << endl;
    cout << "\n\t1) find by Name";
    cout << "\n\t2) find by Address";
    cout << "\n\t3) find by Phone number";
    cout << "\n\t4) find by Email address";
    cout << "\n\t5) find by Zip code";
    cout << "\n\t6) Return to Main Menu" << endl;
    return;
}

```

```

void disp_sortMenu() {
    cout << "\n\t\tSort by" << endl;
    cout << "\n\t1) Name";
    cout << "\n\t2) Address";
    cout << "\n\t3) Phone number";
    cout << "\n\t4) Email address";
    cout << "\n\t5) Zip code";
    return;
}

//class debugging
void debug(void) {
    rolodex test;
    vector<rolodex> vec_test(3);

    test.set_name("Jimmy");
    test.set_address("123 main street");
    test.set_phone_number("1-234-555-5555");
    test.set_email("four");
    test.set_zip("55555");
    test.output();

    vec_test[0].set_name("zero");
    vec_test[0].set_name("one");
    vec_test[0].set_name("two");
}

```

```

        return;
    }

int randomizer(unsigned int min, unsigned int max) {
    srand(static_cast<unsigned>(time(nullptr)));
    return rand() % (max - min + 1) + min;
}

a_vitale7@ares:~$ cat rolodex.cpp
#include "rolodex.h"

#include <iostream>
#include <cstring>

using namespace std;

rolodex::rolodex(void) {
    name[0] = '\0';
    address[0] = '\0';
    phone_number[0] = '\0';
    email[0] = '\0';
    zip[0] = '\0';
    return;
}

rolodex::rolodex(const char new_name[], const char new_address[], const char new_ph
    set_name(new_name);
    set_address(new_address);
    set_phone_number(new_phone_number);
    set_email(new_email);

```

```

        set_zip(new_zip);
        return;
    }

/*
rolodex::rolodex(const rolodex& i) {
    strcpy(name, i.name);
    strcpy(address, i.address);
    strcpy(phone_number, i.phone_number);
    strcpy(email, i.email);
    strcpy(zip, i.zip);
    return;
}

*/

void rolodex::input(void) {
    cout << "\nName: ";
    cin >> name;
    cout << "\nAddress: ";
    cin >> address;
    cout << "\nPhone number: ";
    cin >> phone_number;
    cout << "\nEmail: ";
    cin >> email;
    cout << "\nZipcode: ";
    cin >> zip;
    cout << endl;
}

```



```

        return;
    }

void rolodex::output(void) const {
    printf("\nName: %s", name);
    printf("\nAddress: %s", address);
    printf("\nPhone number: %s", phone_number);
    printf("\nEmail: %s", email);
    printf("\nZip: %s", zip);
    cout << endl;
    return;
}

/*
void rolodex::get_name(char Name[], const unsigned long length) const {
    strncpy(Name, name, MAX_NAME);
    //strncpy_s(Name, length, name, MAX_NAME);
    return;
}

void rolodex::get_address(char Address[], const unsigned long length) const {
    strncpy(Address, address, MAX_ADDRESS);
    //strncpy_s(Address, length, address, MAX_ADDRESS);
    return;
}

void rolodex::get_phone_number(char Phone_number[], const unsigned long length) coi

```

```

    strncpy(Phone_number, phone_number, MAX_PHONE);
    //strncpy_s(Phone_number, length, phone_number, MAX_PHONE);
    return;
}

void rolodex::get_email(char Email[], const unsigned long length) const {
    strncpy(Email, email, MAX_EMAIL);
    //strncpy_s(Email, length, email, MAX_EMAIL);
    return;
}

void rolodex::get_zip(char Zip[], const unsigned long length) const {
    strncpy(Zip, zip, MAX_ZIP);
    //strncpy_s(Zip, length, zip, MAX_ZIP);
    return;
}

*/

void rolodex::set_name(const char new_name[]) {
    strncpy(name, new_name, MAX_NAME);
    //strncpy_s(name, new_name, MAX_NAME);
    return;
}

void rolodex::set_address(const char new_address[]) {
    strncpy(address, new_address, MAX_ADDRESS);
    //strncpy_s(address, new_address, MAX_ADDRESS);
}

```

```

        return;
    }

void rolodex::set_phone_number(const char new_phone_number[]) {
    strncpy(phone_number, new_phone_number, MAX_PHONE);
    //strncpy_s(phone_number, new_phone_number, MAX_PHONE);
    return;
}

void rolodex::set_email(const char new_email[]) {
    strncpy(email, new_email, MAX_EMAIL);
    //strncpy_s(email, new_email, MAX_EMAIL);
    return;
}

void rolodex::set_zip(const char new_zip[]) {
    strncpy(zip, new_zip, MAX_ZIP);
    //strncpy_s(zip, new_zip, MAX_ZIP);
    return;
}

a_vitale7@ares:~$ cat rolodex.h
#ifndef ROLODEX_CLASS_HEADER

#define ROLODEX_CLASS_HEADER

const short MAX_NAME = 63, MAX_ADDRESS = 63, MAX_PHONE = 15, MAX_EMAIL = 63, MAX_ZIP = 15;

class rolodex
{
    char name[MAX_NAME] = {'\0'};

```

```

    char address[MAX_ADDRESS] = {'\0'};
    char phone_number[MAX_PHONE] = {'\0'};
    char email[MAX_EMAIL] = {'\0'};
    char zip[MAX_ZIP] = {'\0'};

public:
    rolodex(void); //default constructor
    rolodex(const char new_name[], const char new_address[], const char new_phone_number[], const char new_email[], const char new_zip[]) {
        //rolodex(const rolodex& i); //constructor
        ~rolodex() {}; //destructor?

    void input(void); //input
    void output(void) const; //output

    const char* get_name(void) const { return name; }; //accessor
    const char* get_address(void) const { return address; }; //accessor
    const char* get_phone_number(void) const { return phone_number; }; //accessor
    const char* get_email(void) const { return email; }; //accessor
    const char* get_zip(void) const { return zip; }; //accessor

    /*
void get_name(char Name[], const unsigned long length = 0) const; //copy access
void get_address(char Address[], unsigned long length = 0) const; //copy access
void get_phone_number(char Phone_number[], const unsigned long length = 0) const; //copy access
void get_email(char Email[], const unsigned long length = 0) const; //copy access
void get_zip(char Zip[], const unsigned long length = 0) const; //copy access
*/

```

```

void set_name(const char new_name[]); //mutator
void set_address(const char new_address[]); //mutator
void set_phone_number(const char new_phone_number[]); //mutator
void set_email(const char new_email[]); //mutator
void set_zip(const char new_zip[]); //mutator
};

#endif

a_vitale7@ares:~$ cat strextra.cpp
#include "strextra.h"

#include <iostream>

using namespace std;

char peek_forward() {
    while (cin.peek() == static_cast<int>(' ')) {
        cin.ignore(1);
    }
    return static_cast<char>(cin.peek());
}

char peek_forward(char ch) {
    while (cin.peek() != static_cast<int>(ch)) {
        cin.ignore(1);
    }
    return static_cast<char>(cin.peek());
}

long find(const char* string, const char find) {

```

```

    long i = 0;
    while (string[i] != find && string[i] != null) {
        i++;
    }
    return (string[i] == find) ? (i) : (-1);
}

long find(const char* string, const char* find) {
    long i = 0, y = 0;
    while (string[i] != find[y] && string[i] != null) {
        i++;
    }
    while (string[i + y] != null && find[y] != null && string[i + y] == find[y]) {
        y++;
    }
    return (string[i] == find[0] && find[y] == null) ? (i) : (-1);
}

long find(const char* string, unsigned long search_from, const char find) {
    long i = search_from;
    while (string[i] != find && string[i] != null) {
        i++;
    }
    return (string[i] == find) ? (i) : (-1);
}

long find(const char* string, unsigned long search_from, const char* find) {

```

```

    long i = search_from, y = 0;
    while (string[i] != find[y] && string[i] != null) {
        i++;
    }
    while (string[i + y] != null && find[y] != null && string[i + y] == find[y])
        y++;
    }
    return (string[i] == find[0] && find[y] == null) ? (i) : (-1);
}

short chComp_notCase(const char char_one, const char char_two) {
    return (tolower(char_one) == tolower(char_two)) ? 0 : (tolower(char_one) >
}

short chComp_case(const char char_one, const char char_two) {
    return (char_one == char_two) ? 0 : (char_one > char_two) ? 1 : -1;
}

short strComp_notCase(const char* string_one, const char* string_two, bool skip_punctu
    unsigned long i_1 = 0, i_2 = 0;
    bool equal = false;
    while (string_one[i_1] != null && string_two[i_2] != null) {
        while ((skip_punctuation && (string_one[i_1] == '.' || string_one[i_1]
            i_1++;
        }
    }
    while ((skip_punctuation && (string_two[i_2] == '.' || string_two[i_2]
        i_2++;

```

```

    }
    if (tolower(string_one[i_1]) == tolower(string_two[i_2])) {
        equal = true;
    }
    else {
        return (tolower(string_one[i_1]) > tolower(string_two[i_2]) ? 1 : -1;
    }
    i_1++;
    i_2++;
}
return (equal) ? 0 : -2;
}

short strComp_case(const char* string_one, const char* string_two, bool skip_punctu
    unsigned long i_1 = 0, i_2 = 0;
    bool equal = false;
    while (string_one[i_1] != null && string_two[i_2] != null) {
        while ((skip_punctuation && (string_one[i_1] == '.' || string_one[i_1]
            i_1++;
        }
    }
    while ((skip_punctuation && (string_two[i_2] == '.' || string_two[i_2]
        i_2++;
    }
    if (string_one[i_1] == string_two[i_2]) {
        equal = true;
    }
    else {

```



```
const char null = '\0';

char peek_forward(void);

char peek_forward(char ch);

long find(const char* string, const char find);

long find(const char* string, const char* find);

long find(const char* string, unsigned long search_from, const char find);

long find(const char* string, unsigned long search_from, const char* find);

short chComp_notCase(const char char_one, const char char_two);

short chComp_case(const char char_one, const char char_two);

short strComp_notCase(const char* string_one, const char* string_two, const bool sl

short strComp_case(const char* string_one, const char* string_two, const bool skip_

short strComp_notCase(const char* string_one, const unsigned long maxLength_one, c

short strComp_case(const char* string_one, const unsigned long maxLength_one, consi

#endif

a_vitale7@ares:~$ CPP rolodex_entry.cpp rolodex.cpp rolodex.h strextra.cpp strex
tra.h
rolodex.cpp...
rolodex_entry.cpp***
strextra.cpp...

a_vitale7@ares:~$ ./rolodex_entry.out

Rolodex class program

Main Menu

1) Add entry
2) Edit entry
3) Delete entry
4) Find entry
5) Print all entries
6) Sort entries
7) Quit
```

Choice: 5

Main Menu

1) Add entry
2) Edit entry
3) Delete entry
4) Find entry
5) Print all entries
6) Sort entries
7) Quit

Choice: 1

Name: test one
Address: 111 street
Phone number: 1-222-333-4444
Email: email@email.mail
Zipcode: 55555

New Entry Added!

Main Menu

1) Add entry
2) Edit entry
3) Delete entry
4) Find entry
5) Print all entries
6) Sort entries
7) Quit

Choice: 5

Entry 1:
Name: test one
Address: 111 street
Phone number: 1-222-333-4444
Email: email@email.mail
Zip: 55555

Main Menu

1) Add entry
2) Edit entry
3) Delete entry
4) Find entry
5) Print all entries
6) Sort entries
7) Quit

Choice: 2

Type 0 to cancel.
Which entry would you like to edit: 1

Name: test one
Address: 111 street
Phone number: 1-222-333-4444
Email: email@email.mail
Zip: 55555

type exit to cancel.
Is this the rolodex you would like to edit: yes

- Edit rolodex
- 1) Edit Name
 - 2) Edit Address
 - 3) Edit Phone number
 - 4) Edit Email address
 - 5) Edit Zip code
 - 6) Quit

Choice: 1

New name: renamed one

- Edit rolodex
- 1) Edit Name
 - 2) Edit Address
 - 3) Edit Phone number
 - 4) Edit Email address
 - 5) Edit Zip code
 - 6) Quit

Choice: 2

New address: 112 renamed street

- Edit rolodex
- 1) Edit Name
 - 2) Edit Address
 - 3) Edit Phone number
 - 4) Edit Email address
 - 5) Edit Zip code
 - 6) Quit

Choice: 3

New phone number: 9-888-777-6666

- Edit rolodex
- 1) Edit Name

- 2) Edit Address
- 3) Edit Phone number
- 4) Edit Email address
- 5) Edit Zip code
- 6) Quit

Choice: 4

New email: notemail@not.mail

- Edit rolodex
- 1) Edit Name
 - 2) Edit Address
 - 3) Edit Phone number
 - 4) Edit Email address
 - 5) Edit Zip code
 - 6) Quit

Choice: 5

New zip code: 54321

- Edit rolodex
- 1) Edit Name
 - 2) Edit Address
 - 3) Edit Phone number
 - 4) Edit Email address
 - 5) Edit Zip code
 - 6) Quit

Choice: q

- Main Menu
- 1) Add entry
 - 2) Edit entry
 - 3) Delete entry
 - 4) Find entry
 - 5) Print all entries
 - 6) Sort entries
 - 7) Quit

Choice: 5

Entry 1:
Name: renamed one
Address: 112 renamed street
Phone number: 9-888-777-6666
Email: notemail@not.mail
Zip: 54321

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: two
Address: two
Phone number: two
Email: two
Zipcode: 2222

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:

Name: renamed one
Address: 112 renamed street
Phone number: 9-888-777-6666
Email: notemail@not.mail
Zip: 54321two

Entry 2:

Name: two
Address: two
Phone number: two
Email: two
Zip: 2222

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries

7) Quit

Choice: 3

Type 0 to cancel.
Which entry would you like to delete: 2

Name: two
Address: two
Phone number: two
Email: two
Zip: 2222

Is this the rolodex you would like to delete: yes

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:

Name: renamed one
Address: 112 renamed street
Phone number: 9-888-777-6666
Email: notemail@not.mail
Zip: 54321two

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 7

Exiting program.
a_vitale7@ares:~\$./rolodex_entry.out

Rolodex class program

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: one
Address: one
Phone number: one
Email: one
Zipcode: 11111

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: two
Address: two
Phone number: two
Email: two
Zipcode: 22222

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry

- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: four
Address: four
Phone number: four
Email: four
Zipcode: 4444

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: three
Address: three
Phone number: three
Email: three
Zipcode: 33333

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: THREE
Address: THREE
Phone number: THREE
Email: THREE
Zipcode: 33333

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:

Name: one
Address: one
Phone number: one
Email: one
Zip: 11111two

Entry 2:

Name: two
Address: two
Phone number: two
Email: two
Zip: 22222four

Entry 3:

Name: four
Address: four
Phone number: four
Email: four
Zip: 4444

Entry 4:

Name: three
Address: three
Phone number: three
Email: three
Zip: 33333THREE

Entry 5:

Name: THREE
Address: THREE
Phone number: THREE
Email: THREE
Zip: 33333

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry

- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 6

Sort by

- 1) Name
 - 2) Address
 - 3) Phone number
 - 4) Email address
 - 5) Zip code
- Choice: 1

Sorted by name.

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:

Name: one
Address: one
Phone number: one
Email: one
Zip: 11111four

Entry 2:

Name: four
Address: four
Phone number: four
Email: four
Zip: 4444

Entry 3:

Name: three
Address: three
Phone number: three
Email: three
Zip: 33333THREE

Entry 4:

Name: THREE
Address: THREE

Phone number: THREE
Email: THREE
Zip: 33333two

Entry 5:
Name: two
Address: two
Phone number: two
Email: two
Zip: 22222

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 6

Sort by

- 1) Name
 - 2) Address
 - 3) Phone number
 - 4) Email address
 - 5) Zip code
- Choice: 5

Sorted by zip code.

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:
Name: one
Address: one
Phone number: one
Email: one
Zip: 11111two

Entry 2:

Name: two
Address: two
Phone number: two
Email: two
Zip: 22222three

Entry 3:
Name: three
Address: three
Phone number: three
Email: three
Zip: 33333THREE

Entry 4:
Name: THREE
Address: THREE
Phone number: THREE
Email: THREE
Zip: 33333four

Entry 5:
Name: four
Address: four
Phone number: four
Email: four
Zip: 4444

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 6

Sort by

- 1) Name
 - 2) Address
 - 3) Phone number
 - 4) Email address
 - 5) Zip code
- Choice: 2

Sorted by address.

Main Menu

- 1) Add entry
- 2) Edit entry

- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:

Name: one
Address: one
Phone number: one
Email: one
Zip: 11111four

Entry 2:

Name: four
Address: four
Phone number: four
Email: four
Zip: 4444

Entry 3:

Name: three
Address: three
Phone number: three
Email: three
Zip: 33333THREE

Entry 4:

Name: THREE
Address: THREE
Phone number: THREE
Email: THREE
Zip: 33333two

Entry 5:

Name: two
Address: two
Phone number: two
Email: two
Zip: 22222

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 3

Type 0 to cancel.
Which entry would you like to delete: 2

Name: four
Address: four
Phone number: four
Email: four
Zip: 4444

Is this the rolodex you would like to delete: yes

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:

Name: one
Address: one
Phone number: one
Email: one
Zip: 11111three

Entry 2:

Name: three
Address: three
Phone number: three
Email: three
Zip: 33333THREE

Entry 3:

Name: THREE
Address: THREE
Phone number: THREE
Email: THREE
Zip: 33333two

Entry 4:

Name: two
Address: two
Phone number: two
Email: two
Zip: 22222two

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 6

Sort by

- 1) Name
 - 2) Address
 - 3) Phone number
 - 4) Email address
 - 5) Zip code
- Choice: 5

Sorted by zip code.

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:

Name: one
Address: one
Phone number: one
Email: one
Zip: 11111two

Entry 2:

Name: two
Address: two
Phone number: two
Email: two
Zip: 22222three

Entry 3:

Name: three
Address: three
Phone number: three
Email: three
Zip: 33333THREE

Entry 4:

Name: THREE
Address: THREE
Phone number: THREE
Email: THREE
Zip: 33333two

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 3

Type 0 to cancel.

Which entry would you like to delete: 4

Name: THREE
Address: THREE
Phone number: THREE
Email: THREE
Zip: 33333two

Is this the rolodex you would like to delete: yes

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 3

Type 0 to cancel.

Which entry would you like to delete: 2

Name: two
Address: two
Phone number: two
Email: two
Zip: 22222three

Is this the rolodex you would like to delete: yes

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:

Name: one
Address: one
Phone number: one
Email: one
Zip: 11111three

Entry 2:

Name: three
Address: three
Phone number: three
Email: three
Zip: 33333three

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 3

Type 0 to cancel.
Which entry would you like to delete: 1

Name: one
Address: one
Phone number: one
Email: one
Zip: 11111three

Is this the rolodex you would like to delete: yes

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 3

Type 0 to cancel.
Which entry would you like to delete: 1

Name: three
Address: three
Phone number: three
Email: three
Zip: 33333three

Is this the rolodex you would like to delete: yes

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 7

Exiting program.
a_vitale7@ares:~\$./rolodex_entry.out

Rolodex class program

Main Menu

- 1) Add entry

- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: one
Address: 1
Phone number: 1
Email: 1
Zipcode: 1

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: five
Address: 5
Phone number: 5
Email: 5
Zipcode: 5

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: FIVE
Address: 5
Phone number: 5
Email: 5
Zipcode: 5

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: three

Address: 3
Phone number: 3
Email: 3
Zipcode: 3

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: two two

Address: two two
Phone number: two two
Email: two two
Zipcode: 2 2

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:
Name: one
Address: 1
Phone number: 1
Email: 1
Zip: 1

Entry 2:
Name: five
Address: 5
Phone number: 5
Email: 5
Zip: 5

Entry 3:
Name: FIVE
Address: 5
Phone number: 5
Email: 5
Zip: 5

Entry 4:
Name: three
Address: 3
Phone number: 3
Email: 3
Zip: 3

Entry 5:
Name: two two
Address: two two
Phone number: two two
Email: two two
Zip: 2 2

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 6

Sort by

- 1) Name
- 2) Address
- 3) Phone number
- 4) Email address

5) Zip code
Choice: 5

Sorted by zip code.

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:
Name: one
Address: 1
Phone number: 1
Email: 1
Zip: 1

Entry 2:
Name: two two
Address: two two
Phone number: two two
Email: two two
Zip: 2 2

Entry 3:
Name: three
Address: 3
Phone number: 3
Email: 3
Zip: 3

Entry 4:
Name: five
Address: 5
Phone number: 5
Email: 5
Zip: 5

Entry 5:
Name: FIVE
Address: 5
Phone number: 5
Email: 5
Zip: 5

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 4

Find rolodex

- 1) find by Name
- 2) find by Address
- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: 1

Enter name to search for: two

Rolodex 2 has the name two.

Find rolodex

- 1) find by Name
- 2) find by Address
- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: 3

Enter phone number to search for: tow

Find rolodex

- 1) find by Name
- 2) find by Address
- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: 1

Enter name to search for: two

Rolodex 2 has the name two.

Find rolodex

- 1) find by Name
- 2) find by Address
- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: 1

Enter name to search for: five

Rolodex 4 has the name five.

Find rolodex

- 1) find by Name
- 2) find by Address
- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: 1

Enter name to search for: FIVE

Rolodex 5 has the name FIVE.

Find rolodex

- 1) find by Name
- 2) find by Address
- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: 2

Enter address to search for: 1

Rolodex 1 has the address 1.

Find rolodex

- 1) find by Name
- 2) find by Address
- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: e

Returning to Main Menu.

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: e

Exiting program.
a_vitale7@ares:~\$./rolodex_entry.out

Rolodex class program

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: Jimmy
Address: 445 west wills street
Phone number: 192-2394
Email: jimmys@jmail.net
Zipcode: 32213

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 1

Name: Will Jim

Address: 446 west wills street
Phone number: 385-2395
Email: will@jmail.net
Zipcode: 21354

New Entry Added!

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 2

Type 0 to cancel.
Which entry would you like to edit: 2

Name: Will Jim
Address: 446 west wills street
Phone number: 385-2395
Email: will@jmail.net
Zip: 21354

type exit to cancel.
Is this the rolodex you would like to edit: yes

Edit rolodex

- 1) Edit Name
- 2) Edit Address
- 3) Edit Phone number
- 4) Edit Email address
- 5) Edit Zip code
- 6) Quit

Choice: 5

New zip code: 32213

Edit rolodex

- 1) Edit Name
- 2) Edit Address
- 3) Edit Phone number
- 4) Edit Email address
- 5) Edit Zip code
- 6) Quit

Choice: e

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:

Name: Jimmy
Address: 445 west wills street
Phone number: 192-2394
Email: jimmys@jmail.net
Zip: 32213Will Jim

Entry 2:

Name: Will Jim
Address: 446 west wills street
Phone number: 385-2395
Email: will@jmail.net
Zip: 32213

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 6

Sort by

- 1) Name
 - 2) Address
 - 3) Phone number
 - 4) Email address
 - 5) Zip code
- Choice: 1

Sorted by name.

Main Menu

- 1) Add entry

- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:

Name: Jimmy
Address: 445 west wills street
Phone number: 192-2394
Email: jimmys@jmail.net
Zip: 32213Will Jim

Entry 2:

Name: Will Jim
Address: 446 west wills street
Phone number: 385-2395
Email: will@jmail.net
Zip: 32213

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 4

Find rolodex

- 1) find by Name
- 2) find by Address
- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: 2

Enter address to search for: 445 west

Rolodex 1 has the address 445 west.

Find rolodex

- 1) find by Name
- 2) find by Address

- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: 2

Enter address to search for: west

Rolodex 1 has the address west.
Rolodex 2 has the address west.

Find rolodex

- 1) find by Name
- 2) find by Address
- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: 4

Enter email to search for: @jmail.net

Rolodex 1 has the email @jmail.net.
Rolodex 2 has the email @jmail.net.

Find rolodex

- 1) find by Name
- 2) find by Address
- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: 5

Enter zip to search for: 32213

Rolodex 1 has the zip 32213.
Rolodex 2 has the zip 32213.

Find rolodex

- 1) find by Name
- 2) find by Address
- 3) find by Phone number
- 4) find by Email address
- 5) find by Zip code
- 6) Return to Main Menu

Choice: 6

Returning to Main Menu.

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 5

Entry 1:

Name: Jimmy
Address: 445 west wills street
Phone number: 192-2394
Email: jimmys@jmail.net
Zip: 32213Will Jim

Entry 2:

Name: Will Jim
Address: 446 west wills street
Phone number: 385-2395
Email: will@jmail.net
Zip: 32213

Main Menu

- 1) Add entry
- 2) Edit entry
- 3) Delete entry
- 4) Find entry
- 5) Print all entries
- 6) Sort entries
- 7) Quit

Choice: 7

Exiting program.
a_vitale7@ares:~\$ exit
exit

Script done on 2022-03-07 17:37:14-06:00 [COMMAND_EXIT_CODE="0"]