#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct person {

string ssn, dob, zip, title, fname, lname, street, town, state;

person(string s, string d, string z, string ti, string f, string l, string str, string to, string st) {

ssn = s;

dob = d;

zip = z;

title = ti;

fname = f;

lname = l;

street = str;

town = to;

state = st;

}

};

struct List {

protected:

struct Link {

person\* data;

Link\* next;

Link(person\* d, Link\* n = NULL) {

data = d;

next = n;

}

~Link() {

delete data;

next = NULL;

}

};

Link\* first;

Link\* last;

int length;

public:

List() {

first = NULL;

last = NULL;

length = 0;

}

~List() {

Link\* prev = NULL;

while (first != NULL) {

prev = first;

first = prev->next;

delete prev;

}

first = NULL;

last = NULL;

length = 0;

}

void add\_to\_front(person\*);

void add\_to\_end(person\*);

void read\_file(string);

void printall();

List\* user\_query();

};

void List::add\_to\_front(person\* x) {

first = new Link(x, first);

if (last==NULL) last = first;

length++;

}

void List::add\_to\_end(person\* x) {

Link\* n = new Link(x, NULL);

if (last != NULL) last->next = n;

else first = n;

last = n;

length++;

}

void List::read\_file(string file) {

ifstream f(file.c\_str());

if (f.fail()) {

cout << "Error opening file\n";

return;

}

int counter = 0;

for (int i=0; i<5000; i++) {

string ssn, dob, zip, title, fname, lname, street, town, state;

f >> ssn >> dob >> title >> fname >> lname >> street >> town >> state >> zip;

person\* p = new person(ssn, dob, zip, title, fname, lname, street, town, state);

add\_to\_end(p);

counter++;

if (f.fail()) break;

}

f.close();

}

void List::printall() {

if (first==NULL) cout << "Empty list\n";

else {

Link\* temp = first;

while (temp!=NULL) {

cout << temp->data->ssn << " " << temp->data->dob << " " << temp->data->title << " " << temp->data->fname << " " << temp->data->lname << " " << temp->data->street << " " << temp->data->town << " " << temp->data->state << " " << temp->data->zip << "\n";

temp = temp->next;

}

}

}

bool equal(person\* a, string b, string attribute) {

if (attribute=="ssn") {

if (a->ssn==b) return true; }

else if (attribute=="dob") {

if (a->dob==b) return true; }

else if (attribute=="title") {

if (a->title==b) return true; }

else if (attribute=="fname") {

if (a->fname==b) return true; }

else if (attribute=="lname") {

if (a->lname==b) return true; }

else if (attribute=="street") {

if (a->street==b) return true; }

else if (attribute=="town") {

if (a->town==b) return true; }

else if (attribute=="state") {

if (a->state==b) return true; }

else if (attribute=="zip") {

if (a->zip==b) return true; }

return false;

}

bool begin(person\* a, string b, string attribute) {

int length = b.length();

if (attribute=="ssn") {

for (int i=0; i<length; i++)

if (a->ssn[i] != b[i]) return false;

return true;

}

else if (attribute=="dob") {

for (int i=0; i<length; i++)

if (a->dob[i] != b[i]) return false;

return true;

}

else if (attribute=="title") {

for (int i=0; i<length; i++)

if (a->title[i] != b[i]) return false;

return true;

}

else if (attribute=="fname") {

for (int i=0; i<length; i++)

if (a->fname[i] != b[i]) return false;

return true;

}

else if (attribute=="lname") {

for (int i=0; i<length; i++)

if (a->lname[i] != b[i]) return false;

return true;

}

else if (attribute=="street") {

for (int i=0; i<length; i++)

if (a->street[i] != b[i]) return false;

return true;

}

else if (attribute=="town") {

for (int i=0; i<length; i++)

if (a->town[i] != b[i]) return false;

return true;

}

else if (attribute=="state") {

for (int i=0; i<length; i++)

if (a->state[i] != b[i]) return false;

return true;

}

else if (attribute=="zip") {

for (int i=0; i<length; i++)

if (a->zip[i] != b[i]) return false;

return true;

}

return false;

}

List\* List::user\_query() {

if (first==NULL) {

cout << "Empty list\n";

return NULL;

}

List\* results = new List;

results->read\_file("/home/www/class/een218/ass7f132.txt");

cout << "Input field, 'equal' or 'begins', and a value\n";

while (true) {

string field, comparison, value, attribute, filename;

cin >> field;

if (field=="print" || field=="Print") {

results->printall();

}

else if (field=="clear" || field=="Clear") {

delete results;

List\* results = new List;

results->read\_file("/home/www/class/een218/ass7f132.txt");

cout << "Search cleared.\n";

}

else if (field=="exit" || field=="Exit") {

return this;

}

else if (field=="save" || field=="Save") {

cin >> filename;

ofstream fo(filename.c\_str());

if (fo.fail()) {

cout << "Error creating file\n";

exit(1);

}

Link\* cur = results->first;

if (cur==NULL) break;

while (cur != NULL) {

fo << cur->data->ssn << " " << cur->data->dob << " " << cur->data->title << " " << cur->data->fname << " " << cur->data->lname << " " << cur->data->street << " " << cur->data->town << " " << cur->data->state << " " << cur->data->zip << "\n";

cur = cur->next;

}

fo.close();

cout << "File saved.\n";

}

else cin >> comparison >> value;

if (field=="dob" || field=="birthday" || field=="Dob" || field=="DOB") attribute = "dob";

else if (field=="ssn" || field=="SSN" || field=="Ssn") attribute = "ssn";

else if (field=="title" || field=="Title" || field=="prefix" || field=="Prefix") attribute = "title";

else if (field=="fname" || field=="Fname" || field=="firstname" || field=="First" || field=="first") attribute = "fname";

else if (field=="lname" || field=="Lastname" || field=="lastname" || field=="Last" || field=="last") attribute = "lname";

else if (field=="street" || field=="Street" || field=="Address" || field=="address") attribute = "street";

else if (field=="town" || field=="Town" || field=="city" || field=="City") attribute = "town";

else if (field=="state" || field=="State") attribute = "state";

else if (field=="zip" || field=="Zip" || field=="zipcode" || field=="Zipcode") attribute = "zip";

Link\* prev = NULL;

Link\* curr = results->first;

if (comparison=="equal" || comparison=="Equal" || comparison=="equals" || comparison=="Equals") {

if (curr==NULL) break;

while (curr != NULL) {

if (equal(curr->data,value,attribute)==false) {

results->length = results->length-1;

if (prev==NULL) {

results->first = curr->next;

Link\* temp = curr;

curr = curr->next;

delete temp;

}

else if (curr->next==NULL) {

results->last = prev;

prev->next = NULL;

delete curr;

curr = NULL;

}

else {

prev->next = curr->next;

delete curr;

curr = prev->next;

}

}

else {

prev = curr;

curr = curr->next;

}

}

cout << "Your search returned " << results->length << " results.\n";

}

else if (comparison=="begin" || comparison=="Begin" || comparison=="begins" || comparison=="Begins") {

if (curr==NULL) break;

while (curr != NULL) {

if (begin(curr->data,value,attribute)==false) {

results->length = results->length-1;

if (prev==NULL) {

results->first = curr->next;

Link\* temp = curr;

curr = curr->next;

delete temp;

}

else if (curr->next==NULL) {

results->last = prev;

prev->next = NULL;

delete curr;

curr = NULL;

}

else {

prev->next = curr->next;

delete curr;

curr = prev->next;

}

}

else {

prev = curr;

curr = curr->next;

}

}

cout << "Your search returned " << results->length << " results.\n";

}

}

}

void main() {

List\* a = new List;

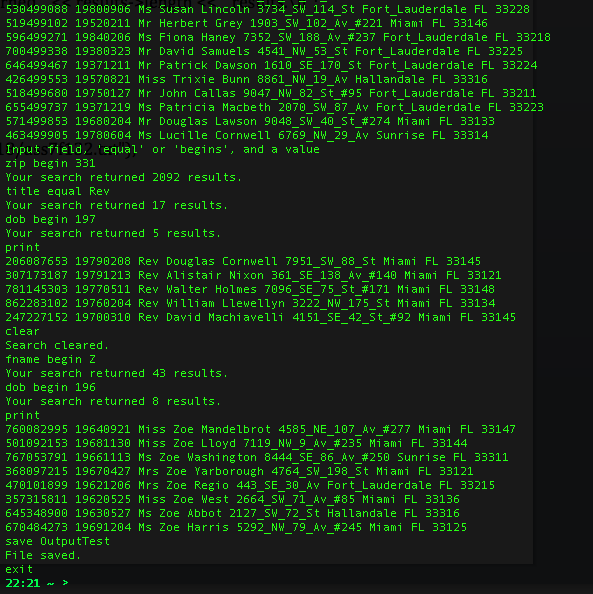
a->read\_file("/home/www/class/een218/ass7f132.txt");

a->printall();

a->user\_query();

}

**Output:**



**Output File**

