```
Script started on Mon 05 Nov 2012 11:46:39 AM CST
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at markov.h
#ifndef MARKOV_H_INC
#define MARKOV_H_INC

#include<iostream>
#include<fstream>
#include<vector>
#include<string>
#include<cstdlib>


using namespace std;

// holds one "event" string and vector of subsequent event pairs, with parralel
// vector of tally and probability
class sequence
{
    string event;
    vector<string> second_events;
    vector<string> third_events;
    vector<short> tallies;
    vector<double> probs;

    void update_prob_list(const vector<short> & tals, vector<double> & prbs);

public:

    sequence(void) : event(string()), second_events(vector<string>()),
                                      third_events(vector<string>()),
                                      tallies(vector<short>()),
                                      probs(vector<double>()) { }

    sequence(const sequence & l) : event(l.event),
                                   second_events(l.second_events),
                                   third_events(l.third_events),
                                   tallies(l.tallies), probs(l.probs) { }

    sequence(const vector<string> & trio, short tal) : event(trio[0]),
                                      second_events(vector<string>(1, trio[1])),
                                      third_events(vector<string>(1, trio[2])),
                                      tallies(vector<short>(1, tal)),
                                      probs(vector<double>(1,0.0)) { }

    bool set_event(string s);
    string get_event(void)
    {
        return event;
    }
    void add_pair(string f, string s, short tal=1);
    bool is_empty(void)
    {
        return event.empty();
    }

    void choose(string & f, string & s);


    void print_to_file(ofstream & out);

};

// holds a vector of sequences with some utility functions - a markov chain!
class mkvchain
{
    vector<sequence> chain;

public:

    mkvchain(void) : chain(vector<sequence>()) { }
    mkvchain(mkvchain & c) : chain(vector<sequence>())
    {
        vector<sequence>::size_type i;
        for (i=0; i< c.chain.size(); i++)
        {
            chain.push_back(c.chain[i]);
        }
    }



    void add_pair(vector<string> trio, short tal=1);

    void get_next_pair(const string & f, string & s, string & t);

    void read_event_file(ifstream & input);

    bool load_chain(ifstream & input);

    void save_chain(ofstream & out);

    sequence get_sequence(vector<sequence>::size_type i)
    {
        return i >= chain.size() ? sequence() : chain[i];
    }
};

void add_sequence(vector<sequence> & seqvec, vector<string> trio);

short strtoshort(string & str);

short char_to_ones(const char & c);

inline long rand_range(long min, long max)
{
       return rand()% (max-min+1)  + min;
}

inline double rand_01(void)
{
       return rand_range(0L, RAND_MAX-1L)/(RAND_MAX-1.0);
}


#endif
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
at markov.cpp
#include "markov.h"
#include<string>
#include<iostream>
#include<fstream>
#include<vector>

using namespace std;


bool sequence::set_event(string s)
{
    bool okay = false;

    if (event.empty())
    {
        event = s;
        okay = true;
    }

    return okay;
```

```
}


/*returns randomly selected "next pair" of strings, weighted by probability of
occurence*/

void sequence::choose(string & f, string & s)
{
    vector<string>::size_type i=0;
    double r = rand_01();
    double choice = probs[i];

    while (r>choice && i<probs.size()-1)
    {
        i++;
        choice += probs[i];
    }

    f = second_events[i];
    s = third_events[i];

    return;

}

// adds "next pair" of strings and updates the tally & probability

void sequence::add_pair(string f, string s, short tal)
{
    vector<string>::size_type i=0;

    //search for second pair
    while (i < second_events.size() &&
          (second_events[i] != f || third_events[i] != s))
    {
        i++;
    }

    // if pair is found increment the tally
    if (i != second_events.size())
    {
        tallies[i] += tal;
    }
    else // or add the pair to list of potential second events and start tally
    {
        second_events.push_back(f);
        third_events.push_back(s);
        tallies.push_back(tal);
        probs.push_back(0.0);
    }

    // re-calculate the probabilites
    update_prob_list(tallies, probs);

    return;
}


// calculates probabilities based on tally of each "next pair"

void sequence::update_prob_list(const vector<short> & tals,
                                      vector<double> & prbs)
{
    short total=0;
    vector<short>::size_type t;

    for (t=0; t < tals.size(); t++)
    {
        total += tals[t]; // sum all the tallies
    }

    for (t=0; t < prbs.size(); t++)
    {
        if (total != 0) // take the probability for each event
        {
            prbs[t] = static_cast<double>(tals[t])/static_cast<double>(total);
        }
    }

    return;
}

// prints table of events and tallys to a file, to be loaded or updated later
void sequence::print_to_file(ofstream & out)
{
    vector<string>::size_type i;

    out << "\nEVENT: " << event;

    for (i=0; i < second_events.size(); i++)
    {
         out << "\nSECOND: " << second_events[i]
              << "\nTHIRD: " << third_events[i]
              << "\nTALLY: " << tallies[i] << '\n';
    }

    return;
}

/* either adds a pair to an existing sequence, or creates a new sequence if it
hasn't happened yet*/

void mkvchain::add_pair(vector<string> trio, short tal)
{
    short i=0;
    while (i < chain.size() &&
               chain[i].get_event() != trio[0] &&
              !chain[i].is_empty())
    {
        i++;
    }
    if (i != chain.size())
    {
        chain[i].add_pair(trio[1], trio[2], tal);
    }
    else
    {
        chain.push_back(sequence());
        if (chain[i].set_event(trio[0]))
        {
            chain[i].add_pair(trio[1], trio[2], tal);
        }
    }
    return;
}


// prints table of events and probabilities of entire chain to file
void mkvchain::save_chain(ofstream & out)
{
    vector<sequence>::size_type i=0;
    while (i<chain.size() && !chain[i].is_empty())
    {
        chain[i].print_to_file(out);
        i++;
    }
}
```

```
// reads in raw data from file line by line and creates a chain object.

void mkvchain::read_event_file(ifstream & input)
{
    string event;
    vector<string> seq;
    vector<string> subseq;
    vector<string>::size_type i;
    while(!input.eof())
    {
        getline(input, event);
        seq.push_back(event);
    }
    for(i=0; i+2<seq.size(); i++)
    {
        subseq.push_back(seq[i]);
        subseq.push_back(seq[i+1]);
        subseq.push_back(seq[i+2]);
        this->add_pair(subseq);
        subseq.clear();
    }
    return;
}

// reads table of events and tallies from file and stores them in memory
bool mkvchain::load_chain(ifstream & input)
{

    string line, label, value;
    string::iterator vit;
    string::size_type pos;
    vector<string> three,seq;
    vector<short> tals;
    vector<string>::size_type vi, t;
    bool event = false;
    long filepos;
    short i, counter=0;
    bool endblock = false;


    while (!input.eof() && !endblock)
    {

        filepos = input.tellg();
        getline(input, line);

        pos = line.find(':');
        label = string(line, 0, pos);
        value = string(line, pos+1, line.length()-1);
        vit = value.begin();

        while(isspace(*vit))
        {
            value.erase(vit);
            vit = value.begin();
        }

        if (label.find("EVENT") != string::npos)
        {
            if (!event)
            {
                three.push_back(value);
                event = true;
            }
            else
            {
                endblock = true;
                input.seekg(filepos);
            }
        }

        else if (label.find("SECOND") != string::npos)
        {

            three.push_back(value);
        }
        else if (label.find("THIRD") != string::npos)
        {

            three.push_back(value);
        }
        else if (label.find("TALLY") != string::npos)
        {
            tals.push_back(strtoshort(value));
        }
    }

    for (vi=0, t=0; vi+2 < three.size(); vi+=2, t++)
    {
        seq.push_back(three[0]);
        seq.push_back(three[vi+1]);
        seq.push_back(three[vi+2]);
        this->add_pair(seq, tals[t]);
        seq.clear();
    }

    return endblock;
}


// finds event of string f, and generates the next events
void mkvchain::get_next_pair(const string & f, string & s, string & t)
{
    short i=0;
    while (i < chain.size()-1 &&
               chain[i].get_event() != f &&
              !chain[i].is_empty())
    {
        i++;
    }
    if (i != chain.size())
    {
        chain[i].choose(s, t);
    }
    return;
}

// non member version to add a sequence to a chain
void add_sequence(vector<sequence> & seqvec, vector<string> trio)
{
    short i=0;
    while (i < seqvec.size() &&
           seqvec[i].get_event() != trio[0] &&
           !seqvec[i].is_empty())
    {
        i++;
    }


    if (!seqvec[i].is_empty())
    {
        seqvec[i].add_pair(trio[1], trio[2]);
    }
    else
    {
        seqvec.push_back(sequence());
        if (seqvec[i].set_event(trio[0]))
```

```
        {
            seqvec[i].add_pair(trio[1], trio[2]);
        }
    }
    return;
}


short strtoshort(string & str)
{
    short answer=0, t;
    string::iterator it = str.begin();

    if (*it == '-' || *it == '+')
    {
        str.erase(it);
    }


    for (it = str.end()-1, t=1; it >= str.begin(); it--,t*=10)
    {
        answer += (char_to_ones(*it) * t);
    }


    return answer;
}

short char_to_ones(const char & c)
{
    short answer;
    switch (c)
    {
    case '0':
        answer = 0;
        break;
    case '1':
        answer = 1;
        break;
    case '2':
        answer = 2;
        break;
    case '3':
        answer = 3;
        break;
    case '4':
        answer = 4;
        break;
    case '5':
        answer = 5;
        break;
    case '6':
        answer = 6;
        break;
    case '7':
        answer = 7;
        break;
    case '8':
        answer = 8;
        break;
    case '9':
        answer = 9;
        break;
    default:
        answer = 0;
    }
    return answer;
}
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ c
```

```
at markov&033[Kdrive.cpp
#include<iostream>
#include<string>
#include<fstream>
#include<vector>
#include<climits>
#include<ctime>
#include "markov.h"

using namespace std;

int main(void)
{
    srand(time(NULL));
    char choice;
    ifstream infile;
    ifstream eventfile;
    ofstream outfile;
    string filename;
    short i, j;
    string data, second, third;
    vector<string> events;
    vector<string> subtrio;
    mkvchain mchain;

    cout << "\n\n\tWelcome to the Markov chain generator!\n";
    do {
        cout << "Choose from the menu:"
             << "\n1. Load Chain From File"
             << "\n2. Read Event File into chain"
             << "\n3. Save Chain To File"
             << "\n4. Generate Some Output!"
             << "\n5. Quit\n";
        cin >> choice;
        cin.clear();
        cin.ignore(INT_MAX, '\n');
        if (choice == '1')
        {
            cout << "\nChain file to load: ";
            getline(cin, filename);
            infile.open(filename.c_str());
            while (!infile)
            {
                infile.close();
                infile.clear();

                cout << "File Not Found!" << endl;

                cout << "Input file name:  ";
                getline(cin, filename);

                infile.open(filename.c_str());
            }
            infile.peek();
            while(!infile.eof())
            {
                mchain.load_chain(infile);
                infile.peek();
            }
            infile.close();
            infile.clear();

            cout << "\nChain loaded successfully!!\n";
        }
        else if (choice == '2')
        {
            cout << "\nEvent file to read: ";
            getline(cin, filename);
            eventfile.open(filename.c_str());
```

```
            while (!eventfile)
            {
                eventfile.close();
                eventfile.clear();

                cout << "File Not Found!" << endl;

                cout << "Input file name:  ";
                getline(cin, filename);

                eventfile.open(filename.c_str());
            }
            mchain.read_event_file(eventfile);

            eventfile.close();
            eventfile.clear();
            cout << "\nEvent file read successfully!\n";
        }
        else if (choice == '3')
        {
            cout << "\nOutput file name:  ";

            getline(cin, filename);
            outfile.open(filename.c_str());
            while (!outfile)
            {
                outfile.close();
                outfile.clear();
                cout << "Invalid output file name!" << endl;

                cout << "Enter the output file name:  ";
                getline(cin, filename);

                outfile.open(filename.c_str());
            }
            mchain.save_chain(outfile);

            outfile.close();
            outfile.clear();

            cout << "\nChain saved to file successfully!!\n";
        }
        else if (choice == '4')
        {
            cout << "\nEnter a string and see what happens! ";


            getline(cin,data);

            for (i=0; i<40; i++)
            {
                mchain.get_next_pair(data, second, third);
                cout << ' ' << second << ' ' << third;
                data = third;
            }

            cin.ignore(INT_MAX, '\n');
        }
        else
        {
            cout << "\nThanks for the memories!!!\n\n";
        }

        } while (choice != '5' && toupper(choice) != 'Q');

    return 0;
}
\033]0;georgia@georgia-MT6017: ~/cplusplus\007georgia@georgia-MT6017:~/cplusplus$ .
/markovdrive.out
       Welcome to the Markov chain generator!
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
2

Event file to read: dickens.in

Event file read successfully!
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! the
 season of foolishness it was the age of Darkness it was the epoch of times it was
the winter of Light it was the season of wisdom it was the spring of hope it was th
e season of times it was the epoch of belief it was the winter of comparison only o
nly  only  only  only  only  only  only  only  only  only  only  only  only  only
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! it
 was the winter of comparison only only  only  only  only  only  only  only  only
only  only  only  only  only  only  only  only  only  only  only  only  only  only
 only  only  only  only  only  only  only  only  only  only  only  only  only  only
  only
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! it
 was the age of belief it was the age of times it was the worst of belief it was th
e period was so far like the other way in short the period that some of its being r
eceived for good or for evil in the superlative degree of foolishness it was the wi
nter of despair we had nothing before us we were all going direct to heaven we were
 all going direct the other way in the superlative degree of
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
2

Event file to read: conrad.in

Event file read successfully!
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
```

```
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! it
 was the superlative degree of comparison only  background  background  background
 background  background  background  background  background  background  background
  background  background  background  background  background  background  backgroun
d  background  background  background  background  background  background  backgrou
nd  background  background  background  background  background  background  backgro
und  background  background  background  background  background
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4t

Now enter some strings and see what happens! it
 is before us we were all going direct to heaven we were all going direct to be alm
ost black fringed with white surf ran straight like the present period that some of
 times it was the age of comparison only background  background  background  backgr
ound  background  background  background  background  background  background  backg
round  background  background  background  background  background  background  back
ground  background  background
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! the
 making with an air of whispering come and she called in every blamed port they hav
e out there for as to be almost black fringed with steam here and there grayish-whi
tish specks showed up clustered inside the white surf ran straight like the present
 period that some of wisdom it was the worst of a colossal jungle so far like the p
resent period that some of Light it was the epoch of times it was the spring of whi
spering come
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
2

Event file to read: stein.in

Event file read successfully!
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! it
 is before us we had nothing before us we were all going direct the son was very di
sturbed then and always mute with an air of whispering come and find out This one w
as the winter of despair we had nothing before us we had nothing before us we were
 all going direct the father said to the season of monotonous grimness the edge of a
 French steamer and she called in him that he does something that he does
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! the
 son was the season of Darkness it was a cruel thing that he does it very often tha
t you are certain this one the winter of comparison only when it was the period was
 fierce the boy was all exciting to him and it slips by the ship is like thinking a
bout it together the two of them and more and more and more and more they talked ab
out it was the son you are wanting to be almost black
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! a
 young man One of despair we were all going direct to heaven we were all going dire
ct the little boy was convinced it was almost featureless as if still in short the
epoch of monotonous grimness the edge of a man has it in a French steamer and the s
on you are wanting to him and still no bigger than pinheads on the untouched expans
e of Darkness it was a man does it very disturbed then and then and then
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! the
 son was blurred by a creeping mist the superlative degree of comparison only  when
 it in him that a cruel one was almost featureless as it slips by the untouched exp
anse of their background  when it was the son was the epoch of belief it was so far
 like the edge of times it was the son you are certain this is a young man when he
is not a blue sea whose glitter was blurred by a
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! man
 does something that he would not a cruel one  when it was the boy was the epoch of
 belief it was the age of foolishness it was the epoch of belief it was a cruel thi
ng that some of their background when it was the epoch of butterflies and it was th
e worst of times it was blurred by the ship is like the present period that a man h
as it together the coast watching a coast watching a
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! a
 cruel one was almost featureless as if still in the superlative degree of comparis
on only  when it was the boy was the best of times it was the epoch of belief it wa
s the epoch of incredulity it was convinced it was all going direct the other way i
n a French steamer and then the two of them and the son was very often that you are
 wanting to glisten and still no bigger than pinheads on the untouched
Choose from the menu:
1. Load Chain From File
```

```
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
4

Now enter some strings and see what happens! a
 flag flying above them and the boy was the age of foolishness it was all arranged
then and she called in every blamed port they have out there grayish-whitish specks
 showed up clustered inside the white surf ran straight like a ruled line far like
the age of monotonous grimness the edge of a cruel one the little son wanted to mak
e a collection of butterflies and beetles and it then and always mute with an aspec
t of Darkness it
Choose from the menu:
1. Load Chain From File
2. Read Event File into chain
3. Save Chain To File
4. Generate Some Output!
5. Quit
q

Thanks for the memories!!!

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

Script done on Mon 05 Nov 2012 11:49:49 AM CST
```