// Brianna Soriano

// Olena Bilinska

// Kenry Yu

//Demo 5:10

#include <iostream>

using namespace std;

int Greg\_to\_Jul(int, int, int);

void Jul\_to\_Greg(int, int &, int &, int &);

int main() {

int m, d;

//lop for numbers asked for

for (int i = 1000; i <= 1500; i++) {

// Febuary month and leap day

m = 2, d = 29;

//passing Gregorian into Julian then back to Jul\_to\_Greg

//checking to see if its still Febuary after adding 1 day or if its March

Jul\_to\_Greg(Greg\_to\_Jul(i, m, d), i, m, d);

if (m == 2)

cout << i <<" is a leap year.\n";

}

return 0;

}

//gregorian function

int Greg\_to\_Jul(int year, int month, int day) {

int i, j, k;

i = year;

j = month;

k = day;

int val;

//formula computes gregorian dates

val = k - 32075 + 1461 \* (i + 4800 + (j - 14) / 12) / 4 +

367 \* (j - 2 - (j - 14) / 12 \* 12) / 12 -

3 \* ((i + 4900 + (j - 14) / 12) / 100) / 4;

return val;

}

void Jul\_to\_Greg(int JD, int &year, int &month, int &day) {

int i, j, k, jd;

i = year;

j = month;

k = day;

int L = JD + 68569;

int N = 4 \* L / 146097;

L = L - (146097 \* N + 3) / 4;

i = 4000 \* (L + 1) / 1461001;

L = L - 1461 \* i / 4 + 31;

j = 80 \* L / 2447;

k = L - 2447 \* j / 80;

L = j / 11;

j = j + 2 - 12 \* L;

i = 100 \* (N - 49) + i + L;

year = i;

month = j;

day = k;

}

Graphical user interface

Description automatically generated with medium confidence