/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Programming excercise 73007 \*

\* Maxwell Stephens \*

\* 12:30 TTh \*

\* 73007, 3/23/17 \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

This program calculates the day of the week of particular date

#include <iostream>

#include <fstream>

using namespace std;

//function identifiers

const int JANUARY = 1;

const int FEBRUARY = 2;

const int MARCH = 3;

const int APRIL = 4;

const int MAY = 5;

const int JUNE = 6;

const int JULY = 7;

const int AUGUST = 8;

const int SEPTEMBER = 9;

const int OCTOBER = 10;

const int NOVEMBER = 11;

const int DECEMBER = 12;

// Returns true if the given year is a leap year

bool isLeapYear(int year);

// Returns a value computed from the century of the year

int getCenturyValue(int year);

// Returns a value computed based on the years since the beginning of the century.

int getYearValue(int year);

// Returns a value for month

int getMonthValue(int month, int year);

int main() {

int month = 0, day = 0, year = 0, dayOfWeek = 0;

cout << "Please enter a day: \n";

cin >> day;

cout << "Please enter a year: \n";

cin >> year;

cout << "Please enter a month: \n";

cin >> month;

cout << endl;

// Compute the day of the week

dayOfWeek = day + getMonthValue(month, year) + getYearValue(year) + getCenturyValue(year);

dayOfWeek = dayOfWeek % 7;

cout << "\nThe date " << month << "/" << day << "/" << year

<< " is a ";

if (dayOfWeek == 0)

{

cout << "Sunday.";

}

else if (dayOfWeek == 1)

{

cout << "Monday.";

}

else if (dayOfWeek == 2)

{

cout << "Tuesday.";

}

else if (dayOfWeek == 3)

{

cout << "Wednesday.";

}

else if (dayOfWeek == 4)

{

cout << "Thursday.";

}

else if (dayOfWeek == 5)

{

cout << "Friday.";

}

else if (dayOfWeek == 6)

{

cout << "Saturday.";

}

cout << endl;

system("pause");

return 0;

}

// Returns true if the given year is a leap year

bool isLeapYear(int year)

{

return (((year % 400) == 0) || (((year % 4) == 0)

&& ((year % 100) != 0)));

}

// Returns a value computed from the century of the year

int getCenturyValue(int year)

{

int century;

int remainder;

century = year / 100;

remainder = (century % 4);

return ((3 - remainder) \* 2);

}

// Returns a value computed based on the years since

// the beginning of the century.

int getYearValue(int year)

{

int sinceCentury;

sinceCentury = year % 100;

return (sinceCentury + (sinceCentury / 4));

}

// Returns a value (from a table) for the given month

int getMonthValue(int month, int year)

{

int result = 0;

if (month == JANUARY)

{

if (isLeapYear(year))

{

result = 6;

}

else

{

result = 0;

}

}

if (month == FEBRUARY)

{

if (isLeapYear(year))

{

result = 2;

}

else

{

result = 3;

}

}

else if (month == MARCH)

{

result = 3;

}

else if (month == APRIL)

{

result = 6;

}

else if (month == MAY)

{

result = 1;

}

else if (month == JUNE)

{

result = 4;

}

else if (month == JULY)

{

result = 6;

}

else if (month == AUGUST)

{

result = 2;

}

else if (month == SEPTEMBER)

{

result = 5;

}

else if (month == OCTOBER)

{

result = 0;

}

else if (month == NOVEMBER)

{

result = 3;

}

else if (month == DECEMBER)

{

result = 5;

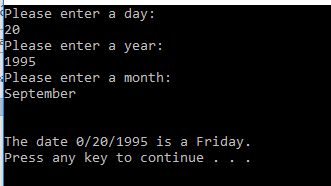
}

return result;

}

/\*

SAMPLE OUTPUT:



Self-Evaluation:

4: Works perfectly, code properly documented

I believe I earned 4 points.

\*/