#include <Arduino.h>

#include <math.h>

struct Location {

const char\* name;

double latitude;

double longitude;

};

Location locations[] = {

{ "Chicago", 41.8781, -87.6298 },

{ "Mumbai", 19.0760, 72.8777 },

{ "New York City", 40.7128, -74.0060 },

{ "Los Angeles", 34.0522, -118.2437 },

{ "Montreal", 45.5017, -73.5673 },

{ "Seoul", 37.5665, 126.9780 },

{ "Denver", 39.7392, -104.9903 }

};

void setup() {

Serial.begin(9600);

while (!Serial) {

; // Wait for serial port to connect. Needed for native USB port only

}

}

void loop() {

Serial.println("Enter your location:");

while (Serial.available() == 0) {

// Wait for user input

}

String userLocation = Serial.readStringUntil('\n');

Serial.println("You entered: " + userLocation);

int inputMonth = 0;

bool locationFound = false;

for (int i = 0; i < sizeof(locations) / sizeof(locations[0]); ++i) {

if (userLocation.equalsIgnoreCase(locations[i].name)) {

locationFound = true;

break;

}

}

if (!locationFound) {

Serial.println("Location not found.");

return;

}

while (true) {

Serial.println("Enter a month (digits, 1-Jan, 2-Feb, ...): ");

while (Serial.available() == 0) {

// Wait for user input

}

String monthInput = Serial.readStringUntil('\n');

if (isdigit(monthInput.charAt(0))) {

inputMonth = monthInput.toInt();

if (inputMonth >= 1 && inputMonth <= 12) {

Serial.println("You entered month: " + String(inputMonth));

break;

} else {

Serial.println("Invalid month format. Please enter digits between 1 and 12.");

}

} else {

// Attempt to parse month as a string abbreviation

monthInput.toLowerCase();

String months[] = { "jan", "feb", "mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec" };

for (int i = 0; i < 12; ++i) {

if (monthInput.equals(months[i])) {

inputMonth = i + 1;

Serial.println("You entered month: " + monthInput);

break;

}

}

if (inputMonth >= 1 && inputMonth <= 12) {

break;

} else {

Serial.println("Invalid month format. Please enter digits between 1 and 12 or use 3-letter abbreviations (e.g., 1 or Jan).");

}

}

}

Serial.println("Enter a specific time (24-hour clock, e.g., 13:30): ");

while (Serial.available() == 0) {

// Wait for user input

}

String timeString = Serial.readStringUntil('\n');

if (!isValidTime(timeString)) {

Serial.println("Invalid time format. Please enter time in 24-hour format (e.g., 13:30).");

return;

}

Serial.println("You entered time: " + timeString);

Serial.println("Enter a year: ");

while (Serial.available() == 0) {

// Wait for user input

}

int year = Serial.parseInt();

Serial.println("You entered year: " + String(year));

int day;

while (true) {

Serial.println("Enter a day: ");

while (Serial.available() == 0) {

// Wait for user input

}

day = Serial.parseInt();

if (isValidDate(day, inputMonth, year)) {

Serial.println("You entered day: " + String(day));

break;

} else {

Serial.println("Invalid day format. Please enter a valid day for the selected month and year.");

}

}

// Calculate solar declination

double declination = 23.45 \* sin((360.0 / 365.0) \* (day-81));

// Calculate hour angle (15 degrees per hour)

double hourAngle = 15.0 \* atof(timeString.c\_str()) - 180;

// Convert latitude and longitude to radians

double latitudeRad = radians(locations[inputMonth - 1].latitude);

double longitudeRad = radians(locations[inputMonth - 1].longitude);

// Calculate solar altitude

double sinAltitude = sin(radians(declination)) \* sin(latitudeRad) + cos(radians(declination)) \* cos(latitudeRad) \* cos(radians(hourAngle));

double altitude = asin(sinAltitude) \* 180.0 / PI;

// Calculate solar azimuth

double cosAzimuth = (sin(radians(declination)) \* sin(latitudeRad) - sin(radians(altitude)) \* sin(latitudeRad)) / (cos(radians(declination)) \* cos(radians(altitude)));

double azimuth = acos(cosAzimuth) \* 180.0 / PI;

if (hourAngle < 0) {

azimuth = 360.0 - azimuth;

}

// Output results

Serial.print("Solar Altitude: ");

Serial.print(altitude);

Serial.println(" degrees");

Serial.print("Solar Azimuth: ");

Serial.print(azimuth);

Serial.println(" degrees");

}

bool isValidTime(String timeString) {

if (timeString.length() != 5 || timeString.charAt(2) != ':') {

return false;

}

int hour = timeString.substring(0, 2).toInt();

int minute = timeString.substring(3).toInt();

return hour >= 0 && hour < 24 && minute >= 0 && minute < 60;

}

bool isValidDate(int day, int month, int year) {

if (month < 1 || month > 12 || day < 1 || day > 31 || year < 1900 || year > 2100) {

return false;

}

if (month == 2) {

if (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0)) {

return day <= 29;

} else {

return day <= 28;

}

} else if (month == 4 || month == 6 || month == 9 || month == 11) {

return day <= 30;

} else {

return true;

}

}