

homework 1 Time Conversion

question description:

2022/9/19 10:00 to UTC to BDS time then to GPS time.

found a interesting that can solve this: [leap second](#)

some definitions

- UT, universal time, time scales based on rotation of earth
- UT0, solar time from astronomical observations
- UT1, UT0 corrected for polar motion based on measurements
- LT, local time, 24 time zone according to longitude
- ET, ephemeris time, time associated with any ephemeris
- TAI, international atomic time, uniform and accurate time
- UTC, universal time coordinated; UT is not precise, while TAI has a problem that earth loses 1 day every 8000 years;
will be adjusted on the last second of Jun 30th or Dec 31st every year;
- GPST, GPS time, TAI
- BDS, BeiDou time, TAI

time conversion

There is a transform chart according to the definitions and some formulas.

$$TAI = \begin{cases} GPST + 19.0s \\ UTC + \text{leap second} \\ BDT + 33.0s \\ TT - 32.184s \end{cases}$$

to be advised, leap second is 37.0s currently;

it's easy to get the answer with input 10:00 2022-9-19, but it's a little complicated to realize a random time conversion.

we need:

- a fully time add and subtraction function, params including year, month, day, hour, min, sec; and the data type needs to be considered;
- GPST, BDT and TT is easy to complete. However, leap second is different by years and months. So we need a leap time judge function;

So we got 2 key points;

parameters

if I give a string like '2022091910000000' to program, then splitting it, converting string to int and double ;

reachable, but complex;

use year, month, day, hour, min, and sec instead;

headers

```
//
// Created by hazyparker on 22-9-19.
//

#ifndef HOMEWORK1_TIMECONVERSION_H
#define HOMEWORK1_TIMECONVERSION_H

// this is a header for class TimeConversion
// to compose time converting from local time to different time

// include libs
#include <vector>
#include <iostream>
#include <tuple>

using namespace std;

typedef vector< vector<int> > Vec2int;
typedef tuple<int, int, int, int, int, double> Time;

class TimeConversion {
private:
    // define time parameters
    Time local_time;
    Time gps_time;
    Time bds_time;
    Time utc_time;
    Time ia_time;

    // define leap second table, table from
    // https://en.wikipedia.org/wiki/Leap\_second
    Vec2int leap_sec_vec;

public:
    /**
     * construction function
     * @param mYear
     * @param mMonth
     * @param mDay
     * @param mHour
     * @param mMin
     * @param mSec
     */
    TimeConversion(int mYear, int mMonth, int mDay,
                  int mHour, int mMin, double mSec);

    /**
     * compose add and subtraction operation to known time spot
     * @param time_origin; time spot for input
     * @param delta; bias of few seconds
     * @param sign; true means positive add, false means negative subtraction
     * @return the result time, in the data type of tuple(type defined before)
     */
}
```

```

static Time TimeAddSubtraction(Time &time_origin, double sec, bool sign);

/**
 * judge if this year is a leap year
 * @param year
 * @return true or false, is or not
 */
static bool isLeapYear(int year);

/**
 * local time to UTC time
 */
void LT2UTC();

/**
 * local time to GPS time
 */
void LT2GPS();

/**
 * local time to BDS time
 */
void LT2BDS();

static void Show(Time &time);

/**
 * show all time
 */
void ShowAllTime();
};

#endif //HOMEWORK1_TIMECONVERSION_H

```

results

for 2022-9-19-10:00:00

```

$/home/hazyparker/project/SJTU-MAI-
SpaceEngineering/1_Courses/NavigationPrinciple/homework1/cmake-build-
debug/homework1
local time, 2022- 9-19, 10:0:0
  IA time, 2022- 9-19,  9:59:27.816
  GPS time, 2022- 9-19,  9:59:8.816
  BDS time, 2022- 9-19,  9:58:54.816
  UTC time, 2022- 9-19,  9:58:50.816

```

for 2000-2-29-9:59:45

```
$/home/hazyparker/project/SJTU-MAI-  
SpaceEngineering/1_Courses/NavigationPrinciple/homework1/cmake-build-  
debug/homework1
```

```
local time, 2000- 2-29, 9:59:45  
  IA time, 2000- 2-29, 9:59:12.816  
  GPS time, 2000- 2-29, 9:58:53.816  
  BDS time, 2000- 2-29, 9:58:39.816  
  UTC time, 2000- 2-29, 9:58:40.816
```