Challenge: Fix the Code

Here is a challenge to fix the code using assert checks.

WE'LL COVER THE FOLLOWING

Problem statement

The program given in this lesson includes a number of `assert` checks.
Compile and run the program to discover the bugs revealed by the `assert` checks.
The program takes a start time and a duration from the user and calculates the end time by adding the duration to the start time:

```
10 hours and 8 minutes after 06:09 is 16:17.
```

Run the program and enter 06:09 as the start time and 1:2 as the duration. Observe that the program terminates normally; you may notice a problem with the output. Although the time that has been entered is 06:09, the output contains 09:06.

Run the program again and this time enter 06:09 and 15:2. Observe that the program is terminated by an `AssertError`. Go to the line of the program that is indicated in the assert message and see which one of the `assert` checks have failed. It may take a while to discover the cause of this particular failure.

Input

```
06:09 15:2
```

```
06:09 15:2
```

Output

Once the code is fixed, the program should produce the following output:

```
15 hours and 2 minutes after 06:09 is 21:11.
```

> **Note:** Enter the input in the stdin block below the code before running the program. Input format example: 06:09 1:2

Challenge

This problem is designed for you to practice, so try to solve it on your own first. If you get stuck, you can always refer to the explanation and solution provided in the next lesson. Good luck!

```
import std.stdio;
import std.string;
import std.exception;

/* Reads the time as hour and minute after printing a
 * message. */
void readTime(string message,
              out int hour,
              out int minute) {
    write(message, "? (HH:MM) ");

    readf(" %s:%s", &hour, &minute);

    enforce((hour >= 0) && (hour <= 23) &&
            (minute >= 0) && (minute <= 59),
            "Invalid time!");
}

/* Returns the time in string format. */
string timeToString(int hour, int minute) {
    assert((hour >= 0) && (hour <= 23));
    assert((minute >= 0) && (minute <= 59));

    return format("%02s:%02s", hour, minute);
}

/* Adds duration to start time and returns the result as the
 * third pair of parameters. */
void addDuration(int startHour, int startMinute,
                 int durationHour, int durationMinute,
                 out int resultHour, out int resultMinute) {
    resultHour = startHour + durationHour;
    resultMinute = startMinute + durationMinute;
    if (resultMinute > 59) {
        ++resultHour;
    }
}

void main() {
    int startHour;
    int startMinute;
    readTime("Start time", startMinute, startHour);

    int durationHour;
    int durationMinute;
    readTime("Duration", durationHour, durationMinute);

    int endHour;
    int endMinute;
    addDuration(startHour, startMinute,
                durationHour, durationMinute,
                endHour, endMinute);

    writefln("%s hours and %s minutes after %s is %s.",
             durationHour, durationMinute,
             timeToString(startHour, startMinute),
             timeToString(endHour, endMinute));
}
```

Program to calculate end time

In the next lesson, you will find a solution to this coding challenge.