

# Challenge: Fix the Code

Here is a challenge to fix another bug in the code given in the previous lesson

## WE'LL COVER THE FOLLOWING ^

- Problem statement
- Input
- Output
- Challenge

## Problem statement #

In the code given below, enter 06:09 and 20:0 and observe that the same `assert` check still fails. Your task is to fix the code so that the `assert` check does not fail.

## Input #

```
06:09 20:0
```

## Output #

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

```
20 hours and 0 minutes after 06:09 is 02:09.
```

## 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", startHour, startMinute);

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

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

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



Program to calculate the end time

---

In the next lesson, you will find solution a to the challenge given above.