



# Time Zones

Tony Wong

2016-01-22

---

## Task Description

- Given 10 different time zones.
- Convert a time from one to another
- 04:08 PST into 20:08 HKT



# Subtasks

	Points	Constraints
1	15	Time zone $A$ is <code>PST</code> . Time zone $B$ is not <code>IST</code> , <code>NPT</code> nor <code>ACDT</code> .
2	13	Time zone $A$ is <code>PST</code> .
3	26	Neither time zones is <code>IST</code> , <code>NPT</code> nor <code>ACDT</code> .
4	22	Neither time zones is <code>NPT</code> .
5	24	No additional constraints

# Statistics

- 100: 41
  - 76: 3
  - 41: 7
  - 54: 3
  - 28: 6
  - 15: 6
  - 0: 9
  - No Attempt: 8
- First solve by  
Lam Kin Long @ 0:19:28

# Common Mistakes

- Those who fail to handle borrowing (退位) for **Hour** will fail subtasks 3, 4, 5
- Those who fail to handle borrowing for **Minutes** will fail subtasks 4, 5

	Points	Constraints
1	15	Time zone $A$ is <code>PST</code> . Time zone $B$ is not <code>IST</code> , <code>NPT</code> nor <code>ACDT</code> .
2	13	Time zone $A$ is <code>PST</code> .
3	26	Neither time zones is <code>IST</code> , <code>NPT</code> nor <code>ACDT</code> .
4	22	Neither time zones is <code>NPT</code> .
5	24	No additional constraints

# Incorrect borrowing

- Check *Hour* before *Minutes*
- Result:
  - 23:67 becomes 24:07

```
if (h < 0) {  
    h += 24;  
}  
if (h >= 24) {  
    h -= 24;  
}  
if (m < 0) {  
    m += 60;  
    h--;  
}  
if (m >= 60) {  
    m -= 60;  
    h++;  
}
```

## Implementation

- Write a function to return number of **minutes** relative to PDT
- Subtract Time Zone A
- Add Time Zone B

```
4 int diff(char* s) {
5     if (s[0] == 'P') return 0;
6     if (s[0] == 'E') return 60 * 3;
7     if (s[0] == 'G') return 60 * 8;
8     if (s[0] == 'I') return 60 * 13 + 30;
9     if (s[0] == 'N') return 60 * 13 + 45;
10    if (s[0] == 'H') return 60 * 16;
11    if (s[0] == 'J') return 60 * 17;
12    if (s[0] == 'A') return 60 * 18 + 30;
13    if (s[2] == 'T') return 60;
14    return 60 * 11;
15 }
```

No need to handle  
"the day before"

```
16 int main() {
17     int h, m;
18     scanf("%d %d %s %s", &h, &m, s, t);
19     int x = h * 60 + m + 60 * 24;
20     x -= diff(s);
21     x += diff(t);
22     printf("%02d %02d\n", x / 60 % 24, x % 60);
23     return 0;
24 }
```

# More Statistics

- Pascal: 19
  - Min: 32 lines    Median: 66 lines    Max: 151 lines
- C/C++: 22
  - Min: 29 lines    Median: 61 lines    Max: 123 lines
  - Shortest code by Lai Wing Yin