Daily Planner Project, CS 135, Fall 2024

Tong Yi

In this project, we read a daily planner in **csv** (comma separated values) format, list all the tasks by its priority, and tally the number of hours for each priority category.

Warning:

1. This is copyrighted materials; you are not allowed to upload to the Internet.
2. Our project is more complicate than similar projects in the Internet and uses a different approach.
   (a) Ask help only from teaching staff of this course.
   (b) Use solutions from ChatGPT or online tutoring websites like, but not limited to, chegg.com violates academic integrity and is not allowed.

1 Time in 24-hour Notation and 12-hour Notation

A time of day is written in the 24-hour notation in the form hh:mm (for example 01:23), where hh (00 to 23) is the number of full hours that have passed since midnight, mm (00 to 59) is the number of full minutes that have passed since the last full hour. For more details, see wikipedia link.

The following figure is taken from https://www.britannica.com/topic/24-hour-clock.

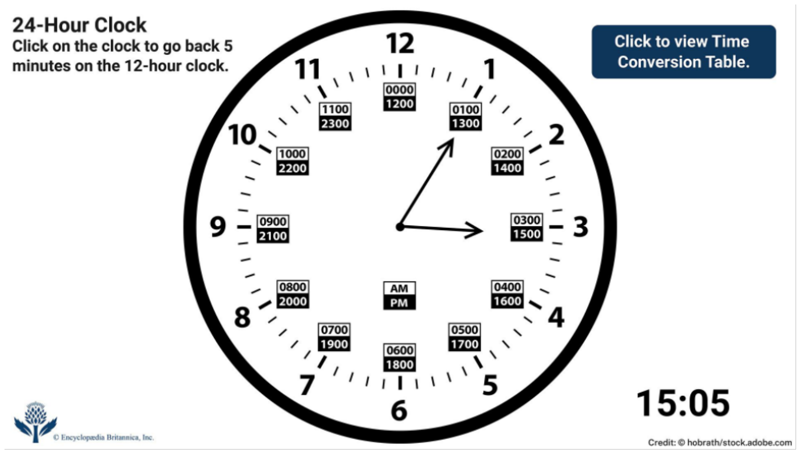
24-Hour Clock
Click on the clock to go back 5 minutes on the 12-hour clock.
Click to view Time Conversion Table.
12 0000 1200
1 0100 1300
2 0200 1400
3 0300 1500
4 0400 1600
5 0500 1700
6 0600 1800
7 0700 1900
8 0800 2000
9 0900 2100
10 1000 2200
11 1100 2300
AM PM
15:05
© Encyclopædia Britannica, Inc.
Credit: © hobrath/stock.adobe.com

Figure 1: convertion between 24-hour notation and 12-hour notation

2 Task A: Convert Time from 24-hour Notation to 12-hour Notation

Relation between 24-hour notation and 12-hour notation is shown in the following table. Let mm be minutes.

24-hour notation	**12-hour notation**
00:mm	12:mm AM
01:mm	1:mm AM
02:mm	2:mm AM
03:mm	3:mm AM
04:mm	4:mm AM
05:mm	5:mm AM
06:mm	6:mm AM
07:mm	7:mm AM
08:mm	8:mm AM
09:mm	9:mm AM
10:mm	10:mm AM
11:mm	11:mm AM
12:mm	12:mm PM
13:mm	1:mm PM
14:mm	2:mm PM
15:mm	3:mm PM
16:mm	4:mm PM
17:mm	5:mm PM
18:mm	6:mm PM
19:mm	7:mm PM
20:mm	8:mm PM
21:mm	9:mm PM
22:mm	10:mm PM
23:mm	11:mm PM

From the above table, we notice the processing of 24-hour format can be categorized by four categories, depending on the value of hour. Each category is represented by different color.

In Task A, input time in 24-hour notation and print out the corresponding 12-hour notation. There are several steps.

1. Name your source code `convert_24_to_12.cpp`
2. Input time in 24-hour notation to a string variable.
3. Extract values of hour and minute from input.
   (a) Use `find` method of `string` class to find out the index of colon (:), a character separating hour and minute.
   (b) Use `substr` method of `string` class to get the substring representing hour and minute.
   (c) Use `stoi` function to convert a string to the corresponding integers. For example, `stoi("12")` returns 12.
4. If hour is not in [0, 23] (that is, hour is smaller than 0 or hour is larger than 23) or minute is not in [0, 59], then print "invalid input" and return -1. The word "invalid" cannot be missed.
5. Depending on the values of hour, print out the corresponding 12-hour notation.
6. Here are some sample input/output.
   (a) When hour or minute is not valid.

```
Enter time (hh:mm) in 24-hour notation (for example, 12:56): 5:61
invalid format
```

(b) When input format is 00:mm. The highlights are input and key part of output.

```
Enter time (hh:mm) in 24-hour notation (for example, 12:56): 00:21
24-hour notation 00:21 in 12-hour notation is 12:21 AM
```

(c) When input hour is in [1, 11]. The highlights are input and key part of output.

```
Enter time (hh:mm) in 24-hour notation (for example, 12:56): 11:05
24-hour notation 11:05 in 12-hour notation is 11:05 AM
```

(d) When input format is 12:mm. The highlights are input and key part of output.

```
Enter time (hh:mm) in 24-hour notation (for example, 12:56): 12:05
24-hour notation 12:05 in 12-hour notation is 12:05 PM
```

(e) When input hour is in [13, 23]. The highlights are input and key part of output.

```
Enter time (hh:mm) in 24-hour notation (for example, 12:56): 17:51
24-hour notation 17:51 in 12-hour notation is 5:51 PM
```

Here are references for `https://cplusplus.com/reference/string/string/find/` and `https://cplusplus.com/reference/string/string/substr/` methods of `string` class.

2.1 example of using find and substr methods of string class

Input a name in the format of LastName,FirstName. Extract first name and last name.

```cpp
#include <iostream>
#include <string>
using namespace std;

//purpose: enter a full name, extract last and first name.
//Sample input/output:
//Enter your full name (LastName,FirstName): Yi,Tong
//Your last name is Yi.
//Your first name is Tong.
int main() {
    cout << "Enter your full name (LastName,FirstName): ";
    string fullName;
    cin >> fullName;

    //find out the index of character ',' in fullName and save it in commaIndex.
    int commaIndex = fullName.find(',');

    //(1) The index of the first character of a string is 0.
    //(2) By Statement,
    //        int commaIndex = fullName.find(',');
    //    commaIndex saves the index of character ',' in string fullName.
    //(3) fullName.substr(0, commaIndex)
    //    extracts a substring from string fullName,
    //    starting from the first character -- whose index is 0 --
    //    all the way to the character just BEFORE the character at index commaIndex,
    //    which is character ','.
    //    Between index 0 and index commaIndex -1,
    //    there are a total of commaIndex -1 - 0 + 1 = commaIndex characters,
    //    that is why the value for the second
   parameter of substr method is commaIndex, as shown in
    //    fullName.substr(0, commaIndex).
    //
    //    So fullName.substr(0, commaIndex) returns the substring
    //    from the first letter of fullName all the way to character just BEFORE ','
    string lastName = fullName.substr(0, commaIndex);
    cout << "Your last name is " << lastName << "." << endl;

    //Note that the following version of substr has only one parameter.
    //fullName.substr(commaIndex+1) extracts a substring from fullName,
    //starting from the character at commaIndex+1,
    //which is the character right AFTER ',' character,
    //all the way to the last character in fullName.
    string firstName = fullName.substr(commaIndex+1);
    cout << "Your first name is " << firstName << "." << endl;
    return 0;
}
```

3 Task B: convert 12-hour time notation to 24-hour time notation

Input time in 12-hour notation and print out the corresponding 24-hour notation. There are several steps.

1. Name your source code `convert_12_to_24.cpp`
2. Input time in 12-hour notation to a string variable.
   (a) The format of 12-hour notation is `hh:mm AM` or `hh:mm PM`, where hh is in [1, 12] and mm is in [0, 59], the input may contain space.
   We cannot use `cin >> variableName;` to read the input, since `>>` stops at the first space or return key, whichever is encounted first.
   To read input with spaces, use `getline(cin, variableName);` read a line, and put the input line to `veriableName`.
   Operator `>>` read an int, a double number, a string, $\cdots$, to the corresponding variable.
   Function `getline` read a line and save that line to a string variable.
3. Extract the last two characters from input, it should be either AM or PM.
4. Extract values of hour and minute from input.
   (a) Use `find` method of `string` class to find out the index of colon (:), a character separating hour and minute.

   (b) Use `substr` method of `string` class to get the substring representing hour and minute.
   (c) Use `stoi` function to convert a string to the corresponding integers. For example, `stoi("12")` returns 12.

5. If hour is not in [1, 12] (that is, hour is smaller than 0 or hour is larger than 12) or minute is not in [0, 59], then print "invalid input" and return -1. The word "invalid" cannot be missed.

   (a) Optional (will not test in gradescope script): test whether input is ended with either "AM" or "PM".

6. Depending on the values of suffix "AM" or "PM" and hour, print out the corresponding 24-hour notation.

7. Here are some sample input/output.

   (a) When hour or minute is not valid.

   ```
   Enter time (hh:mm) in 12-hour notation (for example, 12:56 AM): 5:61 AM
   invalid format
   ```

   (b) When input format is 12:mm AM. The highlights are input and key part of output.

   ```
   Enter time (hh:mm) in 12-hour notation (for example, 12:56): 12:21 AM
   12-hour notation 12:21 AM in 24-hour notation is 00:21.
   ```

   (c) When input format is hh:mm AM, where hh is in [1, 11] and mm is in [0, 59]. The highlights are input and key part of output.

   ```
   Enter time (hh:mm) in 12-hour notation (for example, 12:56): 01:36 AM
   12-hour notation 01:36 AM in 24-hour notation is 01:36.
   ```

   (d) When input format is 12:mm PM, where mm is in [0, 59]. The highlights are input and key part of output.

   ```
   Enter time (hh:mm) in 12-hour notation (for example, 12:56): 12:00 PM
   12-hour notation 12:00 PM in 24-hour notation is 12:00.
   ```

   (e) When input hour is in hh:mm PM, where hh is in [1, 11] and mm is in [0, 59]. The highlights are input and key part of output.

   ```
   Enter time (hh:mm) in 12-hour notation (for example, 12:56): 01:26 PM
   12-hour notation 1:26 PM in 24-hour notation is 13:26.
   ```

4 Task C: calculate duration of two time 24-hour time notations

Enter start- and end- time in 24-hour notation. If start time is later than end time, print "invalid input" and return -1. Otherwise, calculate its duration in whole-number hours and minutes, where minutes is between 0 and 59.

For example, if start time is 12:45 and end time is 11:05, print "invalid input". If start time is 12:45 and end time is 15:30, the duration is 2 hours and 45 minutes.

1. Name the file `get_duration.cpp`.

2. Input start- and end time in 24-hour notation.

3. Extract hours and minutes from start- and end- time.
4. There are two ways to calculate duration between start- and end- time and display the result in whole-number hours and minutes, where minutes is an integer between 0 and 59.
    (a) Approach 1:
        i. Calculate the number of minutes elapsed from midnight 00:00 to end time.
        ii. Calculate the number of minutes elapsed from midnight 00:00 to start time.
        iii. Calculate duration between start- and end- time, in minutes.
        iv. Calculate whole hours and remainder minutes from the previous result using modular operator % and integer division /.
    (b) Approach 2:
        i. If minutes in end time is smaller than minutes in start time, subtract a hour from current hours of end time and add ?? (you find the correct value) minutes to the current minutes in end time.
        ii. Calculate the number of hours and number of minutes for the duration.
        iii. Similar example: to calculate 82 - 17, since ones digit (the rightmost digit) of the minuend (left operand), which is 2, is smaller than the ones digit of the subtrahend (right operand), which is 7, borrow 1 from tens digit from the minuend 82, then the tens digit is changed from 8 to 7.
            A. Perform subtraction for ones digit. In this example, 12 - 7 and get 5.
            B. Perform subtraction for tens digit. In this example, 7 - 1 and get 6.
            C. So the result of 82 - 17 is 65.