

CS2315 Computer Programming

Assignment One (2022-23 Sem. A)

Deadline: **8-Oct-2022 23:59** (Week 6 Sat) No Late submission will be accepted

Problem description

For convenience, dates are usually expressed in the form of six digits. For instance, **31-OCT-1988** could be expressed as **311088**. Among the six digits, the first two (**31**) represent the day, the next two (**10**) represent the month and the last two (**88**) represent the year (**1988**). Therefore this format is being referred to as the **DDMMYY** format.

However, besides the **DDMMYY** format, there exists another format – **YYMMDD** which is also very common. In **YYMMDD** format, **31-OCT-1988** is expressed as **881031**. Of course, for **881031**, there is no ambiguity as **88** can only be the year field (*day cannot be greater than 31*). However, for dates like **111022**, the meaning is ambiguous – it is unknown whether the date is 11-OCT-2022 or 22-OCT-2011.

In this assignment, you will write a program to figure out the original date from the six-digit code (*which could be **DDMMYY** or **YYMMDD***). If the code could be interpreted into two different dates, the program will also output the length of the period between the two dates. The assignment requires the use of **cout**, **iomanip**, **variables** and **if**, it is okay not to use functions or loops. In this assignment you may use *only* the functions from **<iostream>** and **<iomanip>** whenever appropriate (*i.e. no other libraries like **<cmath>**, **<cstring>** or **<cstdlib>***).

The detailed requirements are as follows:

- The input is guaranteed to be a six-digit integer. No special checking is needed.
(*i.e. it will not be shorter/longer or contains non-digit like alphabets*)
- The code could only be **DDMMYY** or **YYMMDD**. (*i.e. no **MMYYDD***)
- The program should output in **exactly the same format** as the sample output. For instance, valid dates should be printed in non-ambiguous form, containing:
 - Two-digit **day** field
 - Three-alphabet **month** field in capital, and
 - Four-digit **year** field
- You may assume that the year range is **1950** to **2049** inclusive.
- The program should also consider **leap year**[#]
- If the six-digit code could represent only one valid date, print the six-digit code and the date as shown in sample output.

- If the six-digit code could represent two different valid dates:
 - Print the two dates as sample output with the **earlier date printed first**.
 - The program should also print the length of the time difference in days.
 - The sentence for time difference should be inside a **box of # characters with a length which is just enough**.
- If the code could not be interpreted as a valid date, report that the date is not valid as in the sample output.

#In a leap year, there are 29 days in February. A year is a leap year if the four-digit year number is divisible by 4 and not divisible by 100. (Multiples of 400 are also leap years, despite that it's divisible by 100).

Sample Input / Output:

Example 1: (User Input is underlined)

```
Input the six-digit date: 311088
311088 should be 31-OCT-1988
```

Example 2: (User Input is underlined)

```
Input the six-digit date: 881031
881031 should be 31-OCT-1988
```

Example 3: (User Input is underlined)

```
Input the six-digit date: 121314
121314 is not a valid date.
```

Example 4: (User Input is underlined)

```
Input the six-digit date: 111213
111213 could be 13-DEC-2011 or 11-DEC-2013
#####
# The ambiguity causes a time difference of 729 days! #
#####
```

Example 5: (User Input is underlined)

```
Input the six-digit date: 131211
131211 could be 13-DEC-2011 or 11-DEC-2013
#####
# The ambiguity causes a time difference of 729 days! #
#####
```

Example 6: (User Input is underlined)

```
Input the six-digit date: 010130
010130 could be 30-JAN-2001 or 01-JAN-2030
#####
# The ambiguity causes a time difference of 10563 days! #
#####
```

Marking criteria

Submitted program will be tested repeatedly with PASS. Marks will be graded objectively based on the number of correct outputs reported by PASS.

- If the program is not compilable, zero mark will be given.
- Make sure that the output format (spelling, punctuations, spacing...etc) follows *exactly* the sample output above otherwise PASS will consider your answer incorrect. Note that the marker will make NO manual attempt to check or re-mark program output.

Unlike tutorial exercises, for assignment, PASS will NOT make ALL test cases visible online. (*i.e. there are hidden test cases*). Queries regarding hidden test case(s) will not be entertained. It is the responsibility of a programmer to design own test cases in order to verify the correctness of the written program.

Note that the marking in PASS is automatic, therefore the following situations may lead to extremely low or zero marks:


- Input/output does not match the requirements as defined.
(*e.g. incorrect spacing, incorrect spelling, letter cases or punctuations*)
- Submission of non **.cpp** files (*e.g. .exe or .zip files*)
- Submission with the “**test**” function rather than the “**submit**” function
(*Students may test or submit many times, only the last “submit” will be marked*)

Please also note that the PASS plagiarism check will also be turned on. **The same disciplinary action will be applied without distinguishing which one is the source / copier.** Please safeguard your files if you work on your assignment using public computers (e.g. CityU Lab)

Testing and Submission

Students should submit cpp file to PASS before the deadline. **No other submission method (e.g. hardcopy, email...etc) is accepted.** Students may use whatever IDE/compiler for development, but programming using non-standard, platform-specific features (which is not compilable in PASS/MS Visual Studio) will lead to zero mark. The marker will make no attempt to fix the syntax error and make the code compilable.

If you observe that the answer in PASS is different from the one you get in your PC, **most likely it is caused by programming bugs like uninitialized variables**. The result by PASS will be used for marking without consideration of what you observe in your local PC.

It is advised that students test the programs with PASS (*using the  Test function*) before final submission (*with the “submit” button*). Be warned that the system could be extremely busy and may become sluggish in responding near deadline. Only **.cpp** files are accepted. Do not submit **.obj** , **.exe** or any other files.

To protect yourself, it is advised that you write down your particulars (full name, student number, eid...) in the beginning of your source code *as comment*.