Assignment 1: Stems and Branches

Due: 20:00, Thu 20 Sep 2018 Full marks: 100

Introduction

This assignment allows you to become familiar with Visual Studio Community 2017. (Details on using Visual Studio will be covered in Tutorial 1.) You will write a simple program on the topic of *Stems-and-Branches* (干支; Cantonese romanization gon1-ji1).

Stems-and-Branches, a.k.a. *sexagenary cycle*, is a cycle of sixty terms used for indicating dates, years, etc. in ancient China. Each *term* in the cycle consists of two Chinese characters: the first is called a *Heavenly Stem* (天干; Cantonese romanization tin1-gon1) and the second is called an *Earthly Branch* (地支; Cantonese romanization dei6-ji1). Heavenly Stem can have 10 possibilities, while Earthly Branch can have 12 possibilities. Tables 1 and 2 show the characters for the 10 stems and 12 branches respectively.

Table 1: The Ten Heavenly Stems

Stem Number	1	2	3	4	5	6	7	8	9	10
Chinese Character	甲	乙	丙	丁	戊	己	庚	辛	壬	癸
Cantonese Romanization	gaap3	yut3	bing2	ding1	mou6	gei2	gang1	san1	yam4	gwai3

Table 2: The Twelve Earthly Branches

Branch Number	1	2	3	4	5	6	7	8	9	10	11	12
Chinese Character	子	丑	寅	卯	辰	巳	午	未	申	酉	戌	亥
Cantonese Romanization	ji2	chau 2	yan 4	maau 5	san 4	ji6	ng 5	mei 6	san 1	yau 5	seut 1	hoi 6

The first term in the sexagenary cycle is called 甲子 which combines the first stem and the first branch. The second term in the cycle is called 乙丑 which combines the second stem and the second branch. This pattern continues as 甲子, 乙丑, 丙寅, 丁卯, 戊辰, 己巳, 庚午, 辛未, 壬申, 癸酉, 甲戌, 乙亥, 丙子, 丁丑, …, until it concludes at the 60th term 癸亥. After that, the cycle begins again at 甲子. In this assignment, for the convenience of those unfamiliar with Chinese characters, we use the notation "Sp-Bq" to denote a term in the sexagenary cycle, where p and q are the stem number and branch number respectively. For example, S8-B12 means 辛亥.

The sexagenary cycle can be used for indicating years. For example, year 2018 is called a 戊戌 year (S5-B11). The next year 2019 is 己亥 (S6-B12), and so on. Similarly, the cycle can indicate dates. For example, 31/8/2018 is called a 乙未 day (S2-B8). The next day 1/9/2018 is called a 丙申 day (S3-B9), and so on. (Obviously, using this method of numbering years and dates is not unique, because the cycle contains 60 terms only. But this method plays an important role in Chinese fortune telling.) In

this assignment, you will write a program to convert a Western date into sexagenary dates. The conversion method is stated below.

Converting from Western Years to Cyclic Years

Given a Western year Y, its stem number p_y and branch number q_y can be computed as follows:

$p_y = (Y - 3) \bmod 10$ (However, if $p_y = 0$, then set $p_y = 10$ instead.)
$q_y = (Y - 3) \bmod 12$ (However, if $q_y = 0$, then set $q_y = 12$ instead.)

Note that mod is the *modulo* operation. For example, $7 \bmod 3 = 1$.

Example: year 2013
$p_y = (2013 - 3) \bmod 10 = 2010 \bmod 10 = 0$. As $p_y = 0$, we set $p_y = 10$ instead.
$q_y = (2013 - 3) \bmod 12 = 2010 \bmod 12 = 6$.
Thus, year 2013 is S10-B6 (癸巳).

Converting from Western Dates to Cyclic Dates

Given a Western date $D/M/Y$, its stem number p_d and branch number q_d can be computed as follows:

$$t = \begin{cases} Y - 1, & M \leq 2 \\ Y, & M > 2 \end{cases}$$
$$r = \begin{cases} M + 12, & M \leq 2 \\ M, & M > 2 \end{cases}$$
$$C = \left\lfloor \frac{t}{100} \right\rfloor$$
$$a = t \bmod 100$$
$$g = 4C + \left\lfloor \frac{C}{4} \right\rfloor + 5a + \left\lfloor \frac{a}{4} \right\rfloor + \left\lfloor \frac{3(r+1)}{5} \right\rfloor + D - 3$$
$$i = \begin{cases} 6, & r \text{ is odd} \\ 0, & r \text{ is even} \end{cases}$$
$$z = 8C + \left\lfloor \frac{C}{4} \right\rfloor + 5a + \left\lfloor \frac{a}{4} \right\rfloor + \left\lfloor \frac{3(r+1)}{5} \right\rfloor + D + 1 + i$$
$p_d = g \bmod 10$ (However, if $p_d = 0$, then set $p_d = 10$ instead.)
$q_d = z \bmod 12$ (However, if $q_d = 0$, then set $q_d = 12$ instead.)

Note that $\lfloor x \rfloor$ means the *floor* of x, that is, the largest integer not greater than x. For example, $\lfloor 3.2 \rfloor = \lfloor 3.98 \rfloor = 3$.

Example: date 4/9/2018
$t = 2018$
$r = 9$
$C = \left\lfloor \frac{2018}{100} \right\rfloor = 20$
$a = 2018 \bmod 100 = 18$
$g = 4 \times 20 + \left\lfloor \frac{20}{4} \right\rfloor + 5 \times 18 + \left\lfloor \frac{18}{4} \right\rfloor + \left\lfloor \frac{3 \times (9+1)}{5} \right\rfloor + 4 - 3 = 186$
$i = 6$
$z = 8 \times 20 + \left\lfloor \frac{20}{4} \right\rfloor + 5 \times 18 + \left\lfloor \frac{18}{4} \right\rfloor + \left\lfloor \frac{3 \times (9+1)}{5} \right\rfloor + 4 + 1 + 6 = 276$

$p_d = 186 \bmod 10 = 6$
$q_d = 276 \bmod 12 = 0$ As $q_d = 0$, we set $q_d = 12$ instead.
Thus, 4/9/2018 is a S6-B12 day (己亥).

Program Specification

The program should obtain three integers as user input, which represents a date. You do *not* have to validate the inputs. (That is, we assume that *all inputs are always valid dates*.) Then you apply the above methods to compute the cyclic year and cyclic dates of the input, and print out the result.

Program Output

The following shows some sample output of the program. The blue text is user input and the other text is the program output. You can try the provided sample program for other input. *Your program output should be exactly the same as the sample program* (i.e., same text, same symbols, same letter case, same number of spaces, etc.). Otherwise, it will be considered as *wrong*, even if you have computed the correct result.

```
Enter a date (D M Y): 4 9 2018↵
Year:  S5-B11
Month: 9
Day:   S6-B12
```

```
Enter a date (D M Y): 14 2 2013↵
Year:  S10-B6
Month: 2
Day:   S8-B12
```

```
Enter a date (D M Y): 25 12 2046↵
Year:  S3-B3
Month: 12
Day:   S5-B7
```

Submission and Marking

- Your program file name should be `stembranch.cpp`. Submit the file in Blackboard (https://blackboard.cuhk.edu.hk/).
- Insert your name, student ID, and e-mail as comments at the beginning of your source file.
- You can submit your assignment multiple times. Only the latest submission counts.
- Your program should be *free of compilation errors and warnings*.
- Your program should *include suitable comments as documentation*.
- *Plagiarism* is strictly monitored and *heavily punished* if proven. Lending your work to others is subjected to the same penalty as the copier.