

Switches, Strings and Unicode

Purpose:

Practise manipulating `Strings`, `switches`, `%`, and logic.

Due Dates:

- Complete scripts and testing for
 - Chinese numbers: task 1are due Friday 26, October 2012 *at the beginning of lecture*.
 - Complete scripts and testing for
 - Chinese numbers: task 2, and
 - Chinese numbers: task 3are due Friday 2, November 2012 *at the beginning of lecture*.
-

Translating into Chinese

The goal of this section is produce a program that is capable of printing out the word for a number in Chinese. The underlying Computer Science goal is to learn how to use `switch`-statements, `Strings`, to understand the role of Unicode, and to practise using methods.

The Chinese¹ characters are shown in Table 1 on page 4. It is better to think of these as words than as numerals. For instance, in English the word is “thirty-two” although in numerals it is “32”. In Chinese, the construction of number words is much simpler than in English or French, and the word for “thirty-two” is more like “three-ten-two” and is written (looking at Table 1 on page 4) as “三十二”. Similarly “fourteen” is “十四” (and not “一十四” or “一四”); “two thousand one hundred thirty-one” is “二千百三十一”. It turns out that in order to prevent mistakes and fraud, Chinese accountants use more complicated characters. In case you are curious, these are also shown in Table 1. You don’t need to use the accounting characters for this lab assignment.

Be sure that you understand these rules. What is one hundred twenty-one?²

After the thousands, the rules get a bit complicated because Chinese and Japanese group by powers of 10^4 (万, 亿, and 兆), unlike American English, which groups by powers of 10^3 (thousand, million, billion, trillion, ...) and British English, which uses both 10^3 and 10^6 (thousand, million, milliard, billion, thousand billion, ?, ...).

¹These characters are used in Chinese and Japanese and Korean. These languages also use the western numerals “0” through “9” depending on context.

²Answer: hundred two-ten one or 百二十一. If you don’t understand when to put in ones, ask!

Chinese numbers: task 1

Sometimes JAVA just isn't as modern as you would like it to be. In particular `System.out` is an old part of JAVA that doesn't handle Linux terminals as well as it ought to. In order to get the Chinese characters to print in a terminal window, you need to do some fancy stuff. Here is a really short "main" that shows how you can get Chinese output to work

```

1  import java.io.OutputStreamWriter ;
2
3  public static void main(String [] args)
4      throws java.io.UnsupportedEncodingException
5      {
6      PrintWriter out = new PrintWriter(
7          new java.io.OutputStreamWriter(System.out,"utf8")) ;
8          // from here on you can use out in place of System.out
9          out.println("I should see a Chinese 6: \u516D") ;
10         out.println("\u60A8\u597D, \u4E16\u754C\u3002") ; /* Chinese Hello World */
11         System.exit(0) ;
12     }

```

Here are some notes on the code:

1. You need to use the `OutputStreamWriter` class, so you need to use this `import` statement.
4. Unfortunately, what we are going to try might not work on all machines. The JAVA compiler insists that we add line 4 to acknowledge this. You are going to need to add "`throws java.io.UnsupportedEncodingException`" after any method that contains as discussed in class. "`new java.io.OutputStreamWriter...`"
8. Once you have declared `out` you can use it as a replacement for `System.out`. However, if you use `out.print` just before `Scanner` input, you first need to add a `.flush()` operation, like this:

```

    out.print("Answer my question! ") ; out.flush() ;
    double x = scanner.nextDouble() ;

```

9–10. These lines should print

```

I should see a Chinese 6: 六
您好,世界。

```

⇒ Write a small class to test this code. Be sure that you have it working before going on. Script and hand in your test code on Friday 26, October 2012.

Chinese numbers: task 2

Write a method that looks like

```
public static String chineseCharacter(int n)
    { // to be written ...
    }
```

using `switch`-statements that returns a `String` with one Chinese character for each of the numbers between 1 and 10^8 in Table 1 on the following page. Use the characters in the second column (the hex values are in the third column). If the input number is not in the table, the function should return the empty string `""`.

A typical part of the switch statement might look like:

```
case 6 : answer = "\u516D"; break ;
```

Once you believe that you have a method written correctly, combine it with the `PrintWriter` from `main` from task one as follows. Write a program that asks the user for a number, and prints out the corresponding Chinese character.

Here is sample input and output.

```
Number to print the character of? 100
The standard character for    100 is 百
```

If there is no single character for this number, print out a message saying so.

```
Number to print the character of? 34
There is no Chinese character for 34
```

- ⇒ When you have a working program, create a script file showing your code, your compilation, and your test runs. This and the following task are for hand-in on Friday 2, November 2012.

Chinese numbers: task 3

Here you really need to put on your thinking cap. Write *another* method

```
public static String chineseNumberWord(int n)
    { // to be written ...
    }
```

that *uses* `chineseCharacter` to build more complicated numbers. Figure 1 on the next page illustrates the idea. *Beware that it only works correctly for numbers between 20 and 99!* It pays to plan out how you intend to tackle this problem. Talk to your Laboratory Instructor or the course instructor if you would like feedback on how you intend to tackle the problem before you begin coding.

For almost full marks, it suffices to deal with numbers between 1 and 9,999.

- ⇒ When you have a working program, create a script file showing your code, your compilation, and your test runs. This and the preceding task are for hand-in on Friday 2, November 2012.

```

public static String chineseWordForNumber(int n)
{
    // Warning: this code is buggy!
    int tens = n / 10 ;
    int ones = n % 10 ;
    String answer =
        chineseCharacter(tens) +
        chineseCharacter(10)  +
        chineseCharacter(ones)
    ;
    return answer ;
}

```

Figure 1: Partial program for converting integers to Chinese.

number	C	hex	A	hex
0			零	C1E3
1	一	4E00	壹	58F9
2	二	4E8C	贰	8D30
3	三	4E09	叁	53C1
4	四	56DB	肆	8086
5	五	4E94	伍	4F0D
6	六	516D	陆	9646
7	七	4E03	柒	67D2
8	八	516B	捌	634C
9	九	4E5D	玖	7396
10	十	5341	拾	62FE
100	百	767E	百	767E
1000	千	5343	千	5343
10000	万	4E07	万	4E07
10 ⁸	亿	4EBF	亿	4EBF
10 ¹²	兆	5146	兆	5146

C Ordinary Chinese word for this number

A Accounting Chinese word for this number

Table 1: Chinese numerals in Unicode

