2018-1 Computer Programming Assignment 1

You should do the assignment on your own. You are not allowed to share codes with others and/or copy codes from other resources. If you get caught, you will lose all points from this assignment.

Grading will be done in Linux environment using Java (OpenJDK) 8, identical to that inside the lab machines. Keep that in mind when writing code in other environments. Programs having any kinds of compile errors will receive neither compile nor test case points.

Do not change the format of input and output. You cannot get any points if you do not follow the output specification and print any sorts of different output from the one in this PDF file.

Write everything, including comments, in English. Koreans may incur compile errors, which will make you get no points.

Graders do not expect you to write the codes using terms and/or classes and/or methods that are not discussed in lectures and lab sessions before the deadline. If you use those, you could get minor grade deduction. If you need any guidance related to this, first find the announcement that could be posted in ETL. Then, unless you found what you want, ask to the TA.

Upload your work in ETL. You should submit only a single TAR or ZIP file containing those files:

[1] Problem 1: Dollar.java (D in upper case!).
[2] Problem 2: Tennis.java (T in upper case!).

Do not include any subdirectories or any other files inside your archive, so that we can see your source codes right after unzipping it. Graders will deduct some of your grade if you do not follow this: note that you are not going to get any grades regarding this issue sometime later.

Due of this assignment is 11PM on March 31st. No late submission is allowed.

If you have any questions, write an article to the Class Q&A board in ETL so that everyone can see what is going on. TAs will try to respond your questions and announce modifications of the specification promptly, if any. However, TAs will not be able to answer questions after 5PM on March 30th: it is kindly recommended to start your work as early as possible.

Problem 1

Write a Java code Dollar.java that gets a natural number as an input and displays star-like (a) full shape and (b) border in the screen. The input will be the maximum length of star, which will be given through co nsole.

Examples: (User enters the bold-faced part.)

```
[cp00@cp ~]$ javac Dollar.java
[cp00@cp ~]$ java Dollar
```

```
Type the maximum length: 1
(a)
$
$
$
(b)
$
$
$
```

```
Type the maximum length: 2
(a)
$$
$$
$$
(b)
$$
$$
$$
```

```
Type the maximum length: 3
(a)
 $
$$$
 $
$$$
 $
(b)
 $
$ $
@$@
$ $
 $
```

```
Type the maximum length: 4
(a)
 $$
$$$$
 $$
$$$$
 $$
(b)
 $$
$  $
@$$@
$  $
 $$
```

```
Type the maximum length: 6
(a)
  $$
$$$$$$
  $$
$$$$$$
  $$
(b)
  $$
$    $
 @$$@
$    $
  $$
```

```
Type the maximum length: 9
(a)
    $
  $$$$$
$$$$$$$$$
  $$$$$
$$$$$$$$$
  $$$$$
    $
(b)
    $
  $   $
$       $
  $   $
$       $
  $   $
    $
```

```
Type the maximum length: 11
(a)
      $
    $$$$$
  $$$$$$$$$
$$$$$$$$$$$
  $$$$$$$$$
$$$$$$$$$$$
  $$$$$$$$$
    $$$$$
      $
(b)
      $
    $    $
  $        $
$            $
  $        $
$            $
  $        $
    $    $
      $
```

```
Type the maximum length: 12
(a)
      $$
    $$$$$$
  $$$$$$$$$$
$$$$$$$$$$$$
  $$$$$$$$$$
    $$$$$$
  $$$$$$$$$$
$$$$$$$$$$$$
  $$$$$$$$$$
    $$$$$$
      $$
(b)
      $$
    $      $
  $          $
$              $
  $          $
   @$      $@
  $          $
$              $
  $          $
    $      $
      $$
```

```
[cp00@cp ~]$
```

The content continues on the next page.

@ You should output in exactly the same format as in the examples, including the number of blanks. No points will be given if the output is different from what is written in the specification.
@ These are some of the relationships between the length and the number of dollar signs (`$`) (and at sig ns (`@`), for (b)) printed per line. Try to find the relation between the input and the number of dollar signs pr inted: we will not answer any questions asking for the clarifications regarding the rules.

- 1: 1-1-1.
- 2: 2-2-2.
- 3: 1-3-@1@-3-1.
- 4: 2-4-@2@-4-2.
- 5: 1-5-@1@-5-1.
- 6: 2-6-@2@-6-2.
- 7: 1-5-7-5-7-5-1.
- 8: 2-6-8-6-8-6-2.
- 11: 1-5-9-11-9-11-9-5-1.
- 12: 2-6-10-12-10-@6@-10-12-10-6-2.
- 16: 2-6-10-14-16-14-10-14-16-14-10-6-2.
- 28: 2-6-10-14-18-22-26-28-26-22-18-@14@-18-22-26-28-26-22-18-14-10-6-2.
- 35: 1-5-9-13-17-21-25-29-33-35-33-29-25-21-25-29-33-35-33-29-25-21-17-13-9-5-1.
- 41: 1-5-9-13-17-21-25-29-33-37-41-37-33-29-25-21-25-29-33-37-41-37-33-29-25-21-17-13-9-5-1.

@ You may write additional methods, but everything needs to be inside a class `Dollar`. In addition, ma ke sure you put everything into Dollar.java.
@ Graders will not enter wrong inputs.

Problem 2

Write a Java code Tennis.java that implements tennis scoring system (https://en.wikipedia.org/wiki/Tennis_scoring_system). In this task, you need to consider two major events in Tennis: (a) Australian Open (https://en.wikipedia.org/wiki/Australian_Open) and (b) US Open (https://en.wikipedia.org/wiki/US_Open_(tennis)).

- Choose the type of match and gender. For instance, (US Open/Female), (Australian/Male), etc. .
- Start a game and proceed by entering inputs. After each input, print the current status.

Example: (User enters the bold-faced part.)

// Do not print the comments (/* */) below: these are intended for hints!

/ * Although the following case seems quite long, TAs expect you to follow each line so that the program may calculate the correct score for every possible situation. Note that there are many possible cases. */

```
[cp00@cp ~]$ javac Tennis.java
[cp00@cp ~]$ java Tennis
Type the match (A: Australian Open/U: US Open): A
Type the gender (F: Female/M: Male): M
Australian Open/Male chosen.
Current: 0-0
Type the winner (L: Left/R: Right): L
Current: 0-0(15-0)
Type the winner (L: Left/R: Right): L
Current: 0-0(30-0)
Type the winner (L: Left/R: Right): R
Current: 0-0(30-15)
Type the winner (L: Left/R: Right): L
Current: 0-0(40-15)
Type the winner (L: Left/R: Right): L
Current: 1-0
Type the winner (L: Left/R: Right): L
Current: 1-0(15-0)
Type the winner (L: Left/R: Right): R
Current: 1-0(15-15)
Type the winner (L: Left/R: Right): L
Current: 1-0(30-15)
Type the winner (L: Left/R: Right): R
Current: 1-0(30-30)
Type the winner (L: Left/R: Right): L
Current: 1-0(40-30)
Type the winner (L: Left/R: Right): R
Current: 1-0(40-40) /* Deuce. */
Type the winner (L: Left/R: Right): R
Current: 1-0(40-40A)
Type the winner (L: Left/R: Right): L
Current: 1-0(40-40)
Type the winner (L: Left/R: Right): R
Current: 1-0(40-40A)
Type the winner (L: Left/R: Right): R
Current: 1-1
/* Omitted. */
Current: 5-5(40-15) /* Play until 7-5 when this set becomes 5-5. */
```

```
Type the winner (L: Left/R: Right): L
Current: 6-5
Type the winner (L: Left/R: Right): L
Current: 6-5(15-0)
Type the winner (L: Left/R: Right): L
Current: 6-5(30-0)
Type the winner (L: Left/R: Right): R
Current: 6-5(30-15)
Type the winner (L: Left/R: Right): L
Current: 6-5(40-15)
Type the winner (L: Left/R: Right): L
Current: 7-5
Type the winner (L: Left/R: Right): L
Current: 7-5 0-0(15-0)
Type the winner (L: Left/R: Right): L
Current: 7-5 0-0(30-0)
Type the winner (L: Left/R: Right): R
Current: 7-5 0-0(30-15)
Type the winner (L: Left/R: Right): L
Current: 7-5 0-0(40-15)
Type the winner (L: Left/R: Right): L
Current: 7-5 1-0
/* Omitted. */
Current: 7-5 4-6
Type the winner (L: Left/R: Right): L
Current: 7-5 4-6 0-0(15-0)
Type the winner (L: Left/R: Right): R
Current: 7-5 4-6 0-0(15-15)
Type the winner (L: Left/R: Right): R
Current: 7-5 4-6 0-0(15-30)
Type the winner (L: Left/R: Right): L
Current: 7-5 4-6 0-0(30-30)
Type the winner (L: Left/R: Right): R
Current: 7-5 4-6 0-0(30-40)
Type the winner (L: Left/R: Right): L
Current: 7-5 4-6 0-0(40-40)
Type the winner (L: Left/R: Right): L
Current: 7-5 4-6 0-0(40A-40)
Type the winner (L: Left/R: Right): L
Current: 7-5 4-6 1-0
/* Omitted. */
Current: 7-5 4-6 5-5(30-40)
Type the winner (L: Left/R: Right): R
Current: 7-5 4-6 5-6
Type the winner (L: Left/R: Right): L
Current: 7-5 4-6 5-6(15-0)
Type the winner (L: Left/R: Right): L
Current: 7-5 4-6 5-6(30-0)
Type the winner (L: Left/R: Right): R
Current: 7-5 4-6 5-6(30-15)
Type the winner (L: Left/R: Right): L
Current: 7-5 4-6 5-6(40-15)
Type the winner (L: Left/R: Right): L
```

```
Current: 7-5 4-6 6-6 /* Tie Break at 6. */
Type the winner (L: Left/R: Right): L
Current: 7-5 4-6 6-6(1-0)
Type the winner (L: Left/R: Right): L
Current: 7-5 4-6 6-6(2-0)
Type the winner (L: Left/R: Right): R
Current: 7-5 4-6 6-6(2-1)
Type the winner (L: Left/R: Right): R
Current: 7-5 4-6 6-6(2-2)
/* Omitted. */
Current: 7-5 4-6 6-6(3-5)
Type the winner (L: Left/R: Right): R
Current: 7-5 4-6 6-6(3-6)
Type the winner (L: Left/R: Right): R
Current: 7-5 4-6 6-7(3-7)
Type the winner (L: Left/R: Right): L
Current: 7-5 4-6 6-7(3-7) 0-0(15-0)
/* Omitted. */
Current: 7-5 4-6 6-7(3-7) 2-5(15-40)
Type the winner (L: Left/R: Right): R
Current: 7-5 4-6 6-7(3-7) 2-6
Game finished! /* Australian Open Male. */
[cp00@cp ~]$
```

@ You do not need to take care about doubles (복식).
@ You do not need to consider withdrawals (기권), order of service (서브 순서) and changing courts (코트 체인지).
@ You should output in exactly the same format as in the example. No points will be given if the output is different from what is written in the specification.
@ You may write additional methods, but everything needs to be inside a class `Tennis`. In addition, make sure you put everything into Tennis.java.
@ Graders will not enter wrong inputs.