Homework #1
RELEASE DATE: 09/21/2015
DUE DATE: 10/12/2015, 12:00

As directed below, you need to upload your submission files to the github repository under the exact guidelines of the TAs.

Any form of cheating, lying or plagiarism will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts. Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.

Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.

Only English is allowed for writing any part of your homework. We do not accept any other languages.

1 Description

In this homework, you have to design a variant of the old-maid card game, as shown in `https://en.wikipedia.org/wiki/Old_maid_(card_game)`. In the game, we will take 52 regular cards + 2 joker cards (one black and one red), played by four players. The players are of IDs 0, 1, 2, 3. The text format for outputting each of the 54 cards are

```
[suit of the card][number of the card]
```

where

- Suits for regular (non-joker) cards: Clubs (`C`), Diamonds (`D`), Hearts (`H`), Spades(`S`)
- "Suits" for joker cards: Red (`R`), Black (`B`)
- Rank: `A`, `K`, `Q`, `J`, `10`, `9`, `8`, ..., `2`; a special number `0` denotes jokers

For instance, Hearts Ace: `HA`, Clubs 10: `C10`, Spades 3: `S3`, Red Joker: `R0`. Then, a hand of cards should be sorted in ascending order and formatted as

```
[1st card] [2nd card] ... [last card]
```

The primary key of sorting is the rank with `0` < `2` < `3` < ... < `K` < `A`, and the secondary key is the suit, with `C` < `D` < `H` < `S` for non-jokers and `R` < `B` for jokers. For instance,

```
R0 B0 C7 D7 HJ SQ DK DA SA
```

The steps of the game are as follows. Please note that **whenever outputting multiple cards, always sort them before outputting** as directed above.

(1) In the beginning, deal 54 cards randomly to four players, with players 0 and 1 having 14 cards, and the other two players having 13 cards. Output the following five lines on screen:

```
Deal cards
Player0: [dealt cards of player 0]
Player1: [dealt cards of player 1]
Player2: [dealt cards of player 2]
Player3: [dealt cards of player 3]
```

(2) Each player should then identify the pairs on their hand, and drop all the pairs. Each pair consists of two regular (non-joker) cards that come with the same rank. If there are three cards with the same rank, please keep the card with the highest suit and drop the other two. For instance, {`H5`, `C5`, `S5`} means dropping `C5` and `H5` while keeping `S5`. Output the following five lines on screen:

```
Drop cards
Player0: [remaining cards of player 0]
Player1: [remaining cards of player 1]
Player2: [remaining cards of player 2]
Player3: [remaining cards of player 3]
```

(3) Next, the game starts. Output the following line on screen:

```
Game start
```

(4) In round i of the game, where $i = 0, 1, \ldots$, player (i%4) draws one random card from player (($i+1$)%4). After drawing, output the following line on screen:

```
Player[IDTo] draws a card from Player[IDFrom] [suit of the card][rank of the card]
```

(5) After the drawing, if player (i%4) finds a pair in her/his hand, drop the pair. Then, *regardless of whether a pair is dropped*, output the following two lines:

```
Player[IDTo]: [remaining cards of player IDTo]
Player[IDFrom]: [remaining cards of player IDFrom]
```

(6) Keep going on each round of the game until someone has no card in her/his hand. Then, output the following line:

```
Player[ID] wins
```

If two players win at the same time, print out both players' winning message by ascending ID (i.e. [ID1] < [ID2]).

```
Player[ID1] and Player[ID2] win
```

(7) After some one wins, output the following line:

```
Basic game over
```

2 Sample Output

```
Deal cards
Player0: C2 H4 S4 C5 D5 H5 S5 D6 H6 S6 D8 H8 S9 DJ
Player1: B0 S2 C3 D3 S3 D7 H9 D10 S10 SJ CQ SQ CA HA
Player2: H3 C4 C6 H7 S7 C8 S8 C9 D9 CJ SK DA SA
Player3: R0 D2 H2 D4 C7 C10 H10 HJ DQ HQ CK DK HK
Drop cards
Player0: C2 S6 S9 DJ
Player1: B0 S2 S3 D7 H9 SJ
Player2: H3 C4 C6 CJ SK
Player3: R0 D4 C7 HJ HK
Game start
Player0 draws a card from Player1 B0
Player0: B0 C2 S6 S9 DJ
Player1: S2 S3 D7 H9 SJ
Player1 draws a card from Player2 H3
Player1: S2 D7 H9 SJ
Player2: C4 C6 CJ SK
Player2 draws a card from Player3 D4
Player2: C6 CJ SK
Player3: R0 C7 HJ HK
```

```
Player3 draws a card from Player0 S6
Player3: R0 S6 C7 HJ HK
Player0: B0 C2 S9 DJ
Player0 draws a card from Player1 H9
Player0: B0 C2 DJ
Player1: S2 D7 SJ
Player1 draws a card from Player2 C6
Player1: S2 C6 D7 SJ
Player2: CJ SK
Player2 draws a card from Player3 HK
Player2: CJ
Player3: R0 S6 C7 HJ
Player3 draws a card from Player0 C2
Player3: R0 C2 S6 C7 HJ
Player0: B0 DJ
Player0 draws a card from Player1 D7
Player0: B0 D7 DJ
Player1: S2 C6 SJ
Player1 draws a card from Player2 CJ
Player1: S2 C6
Player2:
Player2 wins
Basic game over
```

3 Writing Java Programs

Assuming that you know C programming in some sense, you should only need to know the following to finish this homework.

- Entry point of your program should be `main` inside the `PlayGame` class, as taught in the 0921 class
- outputting: use `System.out.print` and `System.out.println`. For instance, the following prints a line of `Basic game over`.

  ```
  System.out.println("Basic game over");
  ```

- All the flow control terms (`if`, `for`, `while`, etc.) are almost the same as C. All the variable declaration and use, at least in the scope of local variables and function parameter passing, are almost the same as C.
- You can put a function *inside your class* with `static` before the function header to make it callable from `main`. For instance,

  ```
  public class PlayGame{
    static int lalala(){

    }

    public static void main(String[] argv){
      int lululu = lalala();
    }
  }
  ```

- If you must use an array, here is the step of initialization.

  ```
  int[] arr = new int[1126];
  ```

We intend to leave all the other steps for finishing your homework as your *trial-and-error* burden. Please feel free to discuss with the TAs for any questions. This homework is *not* intended to be a "good-use-of-Java" homework. So any noodle-oriented-programming skills can be used.

4 Submission File

Please submit your code with github as directed in the homework submission guide. You need to have at least the following files.

- `PlayGame.java`
- any other java code that you write
- `Makefile`: the TAs will type `make` to compile compile your source files from the command line

The TAs will use the command `java PlayGame` to grade your HW1. So please do make sure that you have a valid `main` in `public class PlayGame` of `PlayGame.java`.

5 Bonus

The TAs will consider giving a 20 point bonus to implementations that can continue playing the game to let the rest of the players keep playing the game. That is, each remaining player now needs to draw one card from the *next non-winning player* in the cycle of players. For instance, if player 2 wins, then player 1 should draw card from player 3. The "Bonus Game" ends when only one player has cards.

To get the bonus, please output one line of

```
Continue
```

from the basic game above, and then keep each round of the game going as usual. In the end of the bonus game, output one line of

```
Bonus game over
```

The sample output of the bonus game is as follows.

```
[continue from the sample output above until Basic game over]
Continue
Player3 draws a card from Player0 D7
Player3: R0 C2 S6 HJ
Player0: B0 DJ
Player0 draws a card from Player1 S2
Player0: B0 S2 DJ
Player1: C6
Player1 draws a card from Player3 S6
Player1:
Player3: R0 C2 HJ
Player1 wins
Player3 draws a card from Player0 S2
Player3: R0 HJ
Player0: B0 DJ
Player0 draws a card from Player3 R0
Player0: R0 B0 DJ
Player3: HJ
Player3 draws a card from Player0 DJ
Player3:
Player0: R0 B0
Player3 wins
Bonus game over
```