

Write a Java program for a simple version of *Go Fish* game.

<https://cardgames.io/gofish/>

The game is played with 24-card deck (4 times numbered from 1 to 6).

The game is played by 2 players (human vs computer).

Objective

The objective of the game is to collect as many "books" as possible. A *book* is four cards of the same rank. The player with the most books at the end of the game wins.

Dealing and setup

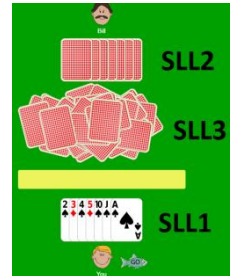
First, cards are dealt to the players, each player gets 7 cards. Once the dealing is done, the rest of the deck is put in a random pile on the table.

There are three single-linked lists (SLL):

1- SLL1 to hold the cards of the first player (human)

2- SLL2 to hold the cards of the second player (computer)

3- SLL3 to hold the cards on the table



Asking and fishing

You opened the game. You asks the computer for a particular rank. For example, you might ask if it has any sixes. You may only ask for ranks that you already have at least one card of. For instance, if you don't have any sixes yourself you can't ask for them.

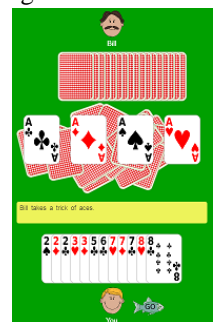
- If the computer has any sixes, then it must give them to you, and you get another turn and can ask again.

- If the computer doesn't have any sixes then it will tell you to "Go Fish" which means that you will draw one card from the pile on the table. If you get a six, then you get a book.

When the computer plays, it randomly selects and asks a rank that it already has at least one card of. The rest of the game is the same.

Scoring

If you have 4 of the same rank then you show the cards to the other player, and then place the four cards in a pile next to you. This is called a *book*. The player with the most books at the end of the game wins.



End of the game

If one of the players (human or computer) finishes all the cards in his hand, the game is over.

The winner should be displayed, if exists. The game may be ended without any winner (in the case of tie).

The program must display all steps until the game is over.

In your program, you can use [single-linked list](#) as you want, but you must use only [single-linked list](#), don't use other data structures such as array, stack and queue.

The user only must enter numbers highlighted in **yellow**.

Don't use array.

Don't use stack or queue.

Don't use linked list class embedded in Java. Write your own single-linked list class.

Sample output:

Turn: 1

You: 1 2 2 2 5 5 6 book: 0
Computer: 1 2 3 4 5 6 6 book: 0

Table

1 1 6 3 5 4 3 4 4 3

You ask: 2

2 2 2 2

Turn: 2

You: 1 5 5 6 book: 1
Computer: 1 3 4 5 6 6 book: 0

Table

1 1 6 3 5 4 3 4 4 3

You ask: 1

Turn: 3

You: 1 1 5 5 6 book: 1
Computer: 3 4 5 6 6 book: 0

Table

1 1 6 3 5 4 3 4 4 3

You ask: 5

Turn: 4

You: 1 1 5 5 5 6 book: 1

Table

1 1 6 3 5 4 3 4 4 3

Computer: 3 4 6 6 book: 0

You ask: 6

Turn: 5

You: 1 1 5 5 5 6 6 6 book: 1
Computer: 3 4 book: 0

Table

1 1 6 3 5 4 3 4 4 3

You ask: 1

Computer says "Go Fish"

Turn: 6

You: 1 1 1 5 5 5 6 6 6 book: 1
Computer: 3 4 book: 0

Table

1 6 3 5 4 3 4 4 3

Computer asks: 3

You say "Go Fish"

Turn: 7

You: 1 1 1 5 5 5 6 6 6 book: 1
Computer: 1 3 4 book: 0

Table

6 3 5 4 3 4 4 3

You ask: 1

1 1 1 1

Turn: 8

You: 5 5 5 6 6 6 book: 2
Computer: 3 4 book: 0

Table

6 3 5 4 3 4 4 3

You ask: 5

Computer says "Go Fish"

Turn: 9

You: 5 5 5 6 6 6 6 book: 2
Computer: 3 4 book: 0

Table

3 5 4 3 4 4 3

6 6 6 6

Computer asks: 3

You say "Go Fish"

Turn: 10

You: 5 5 5 book: 3
Computer: 3 3 4 book: 0

Table

5 4 3 4 4 3

You ask: 5

Computer says "Go Fish"

Turn: 11

You: 5 5 5 5 book: 3
Computer: 3 3 4 book: 0

Table

4 3 4 4 3

5 5 5 5

Game is over.

You win the game !!