Mid-final project (***Go Fish***) Assignment

Due Date: Wednesday November 15, 2017 by 1 PM, via D2L

Administrative details:

- Since the project can be completed in a team of two students (maximum) or by yourself. **This mid-final project can be completed in a team.**
  Sign up for your team (group of 2 or by yourself (work alone)) using the sign-up link below by 5 pm on Thursday November 9, 2017. After this date, you cannot change your team.
  https://docs.google.com/spreadsheets/d/1qphgzm5HRqTMWhrJFZTUzBC7uw2j_LqF4bkhP2An88A/edit?usp=sharing

- Submit your code (mid_final.c) on D2L Dropbox by the due date and time above.

- **This mid-project assignment worth 10 out of 100 points** in your final project score. Late submission:
  - by 1 pm on Thursday November 16, 2017 (full score is 7 out of 10)
  - by 1 pm on Friday November 17, 2017 (full score is 4 out of 10)
  - 0 point for this part after 1 pm Friday Nov 17, 2017.

- Your mid-project code will be graded by ULA (no in-person grading for this part).

Note: The "complete" final project handout is posted in the other pdf file on D2L

Write a C program to

1) Create a full deck of 52 cards that are in order. In other words, for each of the four suits, the cards should be in order from Ace (1) through King (13).
   each card is consisted of
   a suit [clubs (♣),spades (♠),hearts (♥), or diamonds (♦))
   a face (1(Ace) – 10, Jacks (J), Queens (Q), and Kings (K))

   To simulate the deck of 52 cards, and each of the hands, **your team MUST use a dynamic list of cards** with the following strict type:

```
typedef struct card_s {
        char suit;
        int face;
        struct card_s *list;
} card;
```

   Print the deck that you create.

2) Shuffle the deck (52 cards). Note: Your code must have a ***ShuffleCard( ) function*** and should work with *any size deck of cards*, i.e. shuffling 53 cards, 20 cards or 5 cards.

   One algorithm that you may use to shuffle the deck.

(a) For each card in the deck, get a random number in the range of 0 to 51 to be used as the index of the element to swap that card with, i.e.

if deck[0] holds the Jack of clubs (J♣) and the random number generated was 24, and deck[24] holds the 9 of diamonds(9♦), then after the first swap, deck[0] would hold the 9 of diamonds (9♦), and deck[24] would hold the Jack of clubs (J ♣). You would then proceed to deck[1], find a random index of a card to swap with, and swap those cards, etc.

(b) Repeat step a) at least 1000 times.

Print the deck after the shuffle is done.

3) Deal the cards (after shuffling the deck) by giving one card to the user/player, followed by one card to the computer/dealer, followed by one card to the user/player, etc. until **each player get 7 cards.** There are only 2 players (computer/dealer and one user/player).

The player's hand is represented as a dynamic list of cards. The list is populated with the cards drawn by the player.

The dealer's hand is represented as a dynamic list of cards. The list is populated with the cards drawn by the dealer.

Note: the card(s) added to each of the player/dealer's hand (drawn from the deck) must be added to that player/dealer's linked list correctly and MUST be removed from the deck.

Print both computer's hand and user's hand
Print the deck after the hands are created

Sample code execution 1: *NO shuffling* the cards for this one

```
Create 52-card deck
A♠      A♦      A♥      A♣      2♠      2♦      2♥      2♣
3♠      3♦      3♥      3♣      4♠      4♦      4♥      4♣
5♠      5♦      5♥      5♣      6♠      6♦      6♥      6♣
7♠      7♦      7♥      7♣      8♠      8♦      8♥      8♣
9♠      9♦      9♥      9♣      10♠     10♦     10♥     10♣
J♠      J♦      J♥      J♣      Q♠      Q♦      Q♥      Q♣
K♠      K♦      K♥      K♣

Shuffle card
A♠      A♦      A♥      A♣      2♠      2♦      2♥      2♣
3♠      3♦      3♥      3♣      4♠      4♦      4♥      4♣
5♠      5♦      5♥      5♣      6♠      6♦      6♥      6♣
7♠      7♦      7♥      7♣      8♠      8♦      8♥      8♣
9♠      9♦      9♥      9♣      10♠     10♦     10♥     10♣
J♠      J♦      J♥      J♣      Q♠      Q♦      Q♥      Q♣
K♠      K♦      K♥      K♣

Player's hand has 7 cards
A♠      A♥      2♠      2♥      3♠      3♥      4♠

Computer's hand has 7 cards
A♦      A♣      2♦      2♣      3♦      3♣      4♦

Deck after cards are distributed.
4♥      4♣      5♠      5♦      5♥      5♣      6♠      6♦
6♥      6♣      7♠      7♦      7♥      7♣      8♠      8♦
8♥      8♣      9♠      9♦      9♥      9♣      10♠     10♦
10♥     10♣     J♠      J♦      J♥      J♣      Q♠      Q♦
Q♥      Q♣      K♠      K♦      K♥      K♣
```

Note: you can see that the cards are distributed to each hand in an alternate fashion and starts with a player/user hand.

- 1st card from the deck, A♠, is assigned to the player's hand.
- 2nd card from the deck, A♦, is assigned to
The computer's hand.
- 3rd card from the deck, A♥, is assigned to the player's hand.
- etc.

Sample code execution 2: *shuffling* the cards for this one

```
Create 52-card deck
A♠ A♦ A♥ A♣ 2♠ 2♦ 2♥ 2♣
3♠ 3♦ 3♥ 3♣ 4♠ 4♦ 4♥ 4♣
5♠ 5♦ 5♥ 5♣ 6♠ 6♦ 6♥ 6♣
7♠ 7♦ 7♥ 7♣ 8♠ 8♦ 8♥ 8♣
9♠ 9♦ 9♥ 9♣ 10♠ 10♦ 10♥ 10♣
J♠ J♦ J♥ J♣ Q♠ Q♦ Q♥ Q♣
K♠ K♦ K♥ K♣

Shuffle card
5♠ A♥ 4♦ J♦ 9♠ 4♥ Q♥ 7♣
9♥ 3♠ Q♣ 2♣ 6♠ 3♥ 8♠ Q♠
9♦ K♦ 10♥ Q♦ 6♥ 8♥ A♦ 6♣
K♥ 7♦ 5♦ J♣ J♠ 8♦ 5♥ 9♣
8♣ K♠ A♣ 10♣ 5♣ 10♠ 4♣ J♥
10♦ 2♥ 7♠ 4♠ 3♦ 2♦ A♠ 6♦
3♣ 7♥ 2♠ K♣

Player's hand has 7 cards
5♠ 4♦ 9♠ Q♥ 9♥ Q♣ 6♠

Computer's hand has 7 cards
A♥ J♦ 4♥ 7♣ 3♠ 2♣ 3♥

Deck after cards are distributed.
8♠ Q♠ 9♦ K♦ 10♥ Q♦ 6♥ 8♥
A♦ 6♣ K♥ 7♦ 5♦ J♣ J♠ 8♦
5♥ 9♣ 8♣ K♠ A♣ 10♣ 5♣ 10♠
4♣ J♥ 10♦ 2♥ 7♠ 4♠ 3♦ 2♦
A♠ 6♦ 3♣ 7♥ 2♠ K♣
```

Your program will NOT get exactly these shuffle cards since rand() is used!

Note: you can see that the shuffled cards are distributed to each hand in an alternate fashion and starts with a player/user hand.

- 1st card from the shuffled one, 5♠, is assigned to the player's hand
- 2nd card from the shuffled one, A♥, is assigned to the computer's hand
- 3rd card from the shuffled one, 4♦, is assigned to the player's hand