Comp120 Assignment #3
Simplified Black Jack

Objective

You are to design a program to play a simplified version of Black Jack. The outline of the algorithm and data structures you are supposed to use are given below. Your task is to design the **complete** algorithm and to **implement** it following the top-down approach.

Simplified Computer Black Jack

The game is played with one full deck of 52 cards. The values of the cards are as follows:
Ace is 1 point,‘2’ is 2 points, ..., ‘9’ is 9 points and Ten, Jack, Queen and King are 10 points.

The game is for two players. A player’s turn consists of drawing cards from the deck and adding up their values. The aim is to get to 21 points or close to 21 points. If the drawn cards add up to more than 21 points, the sum is reset to 0 and the player can’t draw any more cards.

Players start the game with a pot of 10 (this number should be easy to change in your program) cherries each. A round of the game consists of one turn of each player. In each round each player bets one cherry. The player, whose score is closer to 21, gets the other person’s cherry.
One of the players is simulated by the computer and the other one is a real person, i.e. you are supposed to design a game in which a person can play against the computer. The user goes first. S/he draws cards from the deck and adds their values. S/he can decide to stop drawing cards at any point, but can not draw any more cards if the accumulated sum is bigger than 21. The computer draws cards until the accumulated sum becomes bigger than 15.

The user can decide to quit the program after any number of rounds, but the game should stop if one of the players captured all the other person’s cherries.

Data Structure

The deck of cards can be represented as an array of integers. Assume that the cards are sorted in the following way: spades from the ace to the king, hearts from the ace to the king, diamonds from the ace to the king and clubs spades from the ace to the king. We can assign numbers 0-12 to sorted spades, 13-25 to sorted hearts, 26-38 to sorted diamonds and 39-51 to sorted clubs.

To recover a card from the integer that it is associated with it, we can divide the integer by 13 to get the suit of the card and look at the integer modulo 13 to get the rank of the card.

In order to shuffle the deck, start with the sorted deck as described above and perform the following: Start with the last card and generate a random number between 0 and 51. Swap the last card with the card that is in the position determined by the random number. Generate a random number between 0 and 50 and swap the card which is in that position and the card that is in the 50th position. Generate a random number between 0 and 49 and swap the appropriate cards. Do these until you reach the 1st card.

Outline of the Algorithm

Your program is to behave in the following manner

- display a greeting and the rules of the game
- repeat until the user decides to stop or either of the cherry pots is empty
    - clear the screen
    - display the current contents of the pots
    - play the user's turn and display the drawn cards in a nice way – use S,H,D,C, to represent the suit and A, 2, 3, ..., 9, T, J, Q, K to represent the rank – on the left part of the screen
    - keep displaying the current score as the user is drawing cards from the deck
    - play computer's turn and display the drawn cards in a nice way on the right part of the screen
    - if at any moment all the cards from the deck have been used, reshuffle the deck
    - display the winner of the round
    - ask the user if s/he wants to continue
- display the winner and a goodbye message

A snapshot of the program after a round is finished may look like below:

```
Human: 11(cherries)  Computer: 9(cherries)

9D                        4H
QS                        AH
                          7S
                          AS

Score: 19                 Score: 17

The human won this round!
Shall we continue?
```

Evaluation

Marks will be given according to the following:

- Design - code is clear, efficient, logical, and correct; functions are small and single purpose; function parameters are used instead of global variables.
- Documentation - program is headed by an informative comment; all variables are commented; all complex sections of code are commented; all functions are commented.
- Execution –program works as specified above; program validates all user input; output of program follows above specifications; output is neatly formatted; program is bug-free.