ECE175--Spring 2016 (Crazy Eight)
Grading Rubric for Final Project

As stated in the final project handout, linked list must be used for this project. If your program uses array instead, 25 points will be deducted

	Criterion	Maximum Points	Exemplary (100%)	Proficient (75%)	Marginal (15%)	Unacceptable (0%)
	Requirements					
1	Create Deck, Deck shuffling	5	Deck is created and Deck is shuffled after **every round (every new game)**. The shuffling function produces a new sequence of cards per round and every time the program is restarted	Deck is shuffled but card sequences are not sufficiently unique	Deck is not shuffled after every round or with program restart	Deck is not shuffled
2	Card Dealing	4	Cards are dealt in the proper order. First, one card to the user, followed by one card to the computer, one card to the user, etc.	The right number of cards is dealt, but not in a proper order	Cards are not dealt correctly	Card dealing is not implemented
3	Top card	2	The top card can be correctly picked from the deck			Top card is not implemented correctly
4	**Player's Turn**					
	Let a user pick a card	5	The program can prevent a user to enter that card number that is not available (i.e.continue asking if a user enter any number < 1 or, any number > 9 for the 8-card hand)	75% working (missing checking number < 1 or missing number > number of cards on the hand at that time)	the statements for checking this part are not correct but there was an attempt to do this.	No implementation for this part
	Card Matching	10	The program can 1) (7 pts) find card number on the player's hand that can be played with Top card and 2) (3 pts) keep asking if a user does not enter card that can be played	75% working (the program does not find **all the cards on the hand that can be played with the top card**)	There was an attempt in the program but it does not work	No implementation for this part
	How to deal with 8	4	If a user picks 8, the program asks a user which suit to be played next and continue the game with that suit			No implementation
	Top card updated	2	The top card is updated with the one that the user just played			No implementation
5	**Computer's Turn**					
	Card Matching	8	The program can find 1) (5 pts) card number on the computer's hand that can be played with Top card and 2) (3 pts) generate a random number that can pick card(s) that can be played with Top card	75% working (the program does not find all the cards on the hand that can be played with the top card OR the program pick the first valid card on the computer hand to play all the time	there was an attempt but not working	No implementation
	To deal with 8	2	If the card is 8, the computer can randomly generate the suit to play next			No implementation
	Top card updated	2	The top card is updated with the one that the user just played			No implementation
6	**Game Turn**					
	Player <=> Computer	2	The game starts with player and then computer. The order of play is working			The order of play is not working (i.e. the program stops after the player plays the first time)

	Remove card from the hands and the deck	8	The card on the computer and player's hand can be removed correctly in each turn. The cards from the deck can be removed correctly if the hand needs to take card from the stockpile	(2 pts) Only The card on the computer and player's hand can be removed correctly in each turn. OR (2 pts) only The cards from the deck can be removed correctly if the hand needs to take card from the stockpile; **(-4 pts) if free ( ) is not used in the program**	there was an attempt but not working	No implementaion
	Criterion	**Maximum Points**	**Exemplary (100%)**	**Proficient (75%)**	**Marginal (15%)**	**Unacceptable (0%)**
7	**Game end**					
	Winner annouce	3	At the end of each game, the winner is announced;	The game ends but no winner is announced	there was an attempt but not working	No implementaion
	Case of 0 card in the stockpile	5	there is code segment dealing with the case when there is no card left in the stockpile => the hand with minimum points (add up all the face(1 - 13) values of cards left on the hand) wins.	there is code segment dealing with the case BUT wrong idea on how to choose a winner	there was an attempt but not working	No implementaion
8	Game Interface	8	(4 pts) The interface is intuitive. The user is able to play the game with minimal intructions. Transitions from computer's turn and player's turn (vice versa) are clear. (4 pts) The user can start a new game (or q to quit).	The interface is intuitive for the most part. Some difficulty in understanding the game navigation. (4 pts) The user can start a new game (or q to quit).	The interface is counter-intuitive. Navigation options are not clearly stated. Interface limitations prevent proper game functionality. The user canNOT start a new game (or q to quit).	The interface is very basic and does not allow transitions between turn. No attempt to let a user start a new game.
	Program Design					
9	Code modularity	5	The code is logically divided to several functions that implement **important** functionality	The code is modular but further simplification could have been attempted	The code only has a few functions (ones from the Zybook) - there is no atempt in creating their own user-defined function	Code is not modular (all statements are written in main)
10	Code documentation	5	The code is properly documented. The input/output and goal of every function is adequately described. Comments are provided for various parts of the code	The code is partially documented	The code is scarcely documented	No documentation is provided
11	Compilation	10	Code succesfully compiles without errors or warnings. The code does not hang while in execution	The code succesfully compiles, but some conditions may make it hang (-5 pts)		Code does not compile
	Q&A during final project demo					
12	Questions asked during demo/check-off time	10	Both team members can take turn answering questions asked by TA/ULA correctly.	1-2 questions cannot be answered by the team	Both team members cannot give correct answers for majority of questions asked OR ***Only one team member*** can answer the questions correctly	No show
	Total	**100**				
	Extra Credit		**Exemplary (100%)**	**Acceptable (50%)**		

13	Game Statistics	4	Text file with user name, win and loss is created. A new user can be added and the file is correctly updated.	Text file with user name, win and loss is created but 1) a new user cannot be added or 2) an existing user cannot be updated (one of the two). The file is correctly updated for the case that is working		
14	Graphics	2	See Bonus points section of the final project handout for details			
15	Lazy shuffling	2	See Bonus points section of the final project handout for details			
16	Using several functions (at least CreateDeck, removecard) that shows understanding of how to use pointer to pointer	4	See Bonus points section of the final project handout for details			
	Total	**12**				