Lab 01: Programming Practice

In this lab, you will write a small game in C, and a simple algorithm in x86 (32- or 64-bit) assembly. You may complete the lab using any C compiler and x86 assembler of your choice. The program must run on one of the major x86-based operating systems (Linux, macOS, Windows, 64- or 32-bit). If you are especially daring, you may complete the lab on an arm64-based system, e.g., "Apple Silicon," instead, but please understand, the course will be conducted on x86-based, e.g., "Intel," systems. The professor is less familiar with arm64, and this option will not be offered for all labs. If this is your intent and earnest desire, the please coordinate with the professor earlier rather than later.

Purpose

To refresh your skills in C and assembly programming. This will also help you become familiar with a development environment suitable for the remaining labs in the course.

Task

Your task is to write a simplified 5-card stud poker game in C. There will be one human player, playing against a (rather naïve) computer player. Rounds will continue until someone wins, or the human quits.

A template has been provided for you. It implements everything except for shuffling and scoring. For reference, the rules of this simplified version follow:

Each player starts with 100 chips. Each round, both players pay 2 chips, called the "ante" to the pot. A standard poker deck of 52 cards is shuffled, and 5 cards, called a "hand," is dealt to each player. Betting is simplified from the standard poker rules. The first better is chosen randomly, but it alternates each round after that. The betting player either plays the hand, by betting 3 additional chips, or folds the hand. The calling player either calls, matching the 3 chips, or folds. There is no staying, raising, re-raising, etc. If either player folds, the other player wins the round, but the hands are not revealed. If the second player calls, both hands are revealed, and the winner is chosen by standard poker hand scoring. The winning player of the round takes the pot, and the next round begins. A player wins the game when the opponent cannot afford the ante.

All of this is already implemented by the template except the `shuffle` and `score` functions. Your task is to implement those functions. The shuffling function must be implemented in a separate assembly file for the platform of your choice. A stub for this file is provided for you for a variety of platforms. The scoring function must be implemented in a separate C file.

Tips and Caveats

- Consider a separate function to check for each kind of hand (pair, two pair, three of a kind, strait, etc.)
- I will test your shuffling algorithm for uniformity. "Looking random" is not sufficient.

Grading

There is no report for this lab. Please submit source code and compiled binaries. This lab is due midnight after 2 weeks.

Rubric

points	Accomplishment
3	C/Assembly program compiles for an approved platform
3	5-card hands are dealt randomly each round
1	Provided template code and rules have not been modified
3	Hand scoring is properly implemented

Appendix: Poker Hands

A standard poker deck consists of 52 cards formed from each combination of 4 suits and 13 ranks. From lowest to highest, the ranks are 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack (J), Queen (Q), King (K), and Ace (A). If it is advantageous, a player

may choose to play an Ace "low." That is, it may be ranked as if a 1. This is only useful to form a straight: 5, 4, 3, 2, A[1]. For this straight, the highest card is the 5, since the Ace is low. The suits, from lowest to highest[2], are Clubs ♣, Diamonds ♢, Hearts ♡, and Spades ♠. No card is duplicated, and a standard deck does not contain any Jokers, i.e., wild cards. The following is a table of categories of poker hands, from highest to lowest, including an example for each. The order in which the cards are dealt is irrelevant. The player may arrange the cards to form the highest hand possible. It is useful to arrange the cards in the order their ranks are compared. The cards participating in the hand are arranged first, in highest-to-lowest order by rank, then suit. The remaining cards, often called "kickers," if any, are arranged after, again in highest-to-lowest order by rank, then suit. Each example in the table is ordered this way:

Category	**Description**	**Example Hand**
Royal Flush	The highest-rank Straight Flush	A♠ K♠ Q♠ J♠ 10♠
Straight Flush	A Straight and a Flush	9♡ 8♡ 7♡ 6♡ 5♡
Four of a Kind	4 cards of equal rank	5♠ 5♡ 5♣ 5♢ J♣
Full House	Three of a Kind and a distinct Pair	6♠ 6♣ 6♢ Q♡ Q♣
Flush	5 cards of equal suit	J♣ 9♣ 8♣ 4♣ 3♣
Straight	5 cards of consecutive rank	J♡ 10♣ 9♣ 8♠ 7♡
Three of a Kind	3 cards of equal rank	K♠ K♡ K♢ 8♢ 5♣
Two Pair	A Pair and another distinct Pair	9♡ 9♢ 4♡ 4♣ J♠
Pair	2 cards of equal rank	5♡ 5♣ K♢ Q♣ 3♠
High Card	None of the categories above	K♠ J♠ 7♢ 6♢ 3♣

The winner is the player with the highest-scoring hand. Hands are compared first by category, then by rank, then by suit, per the house rule. Here are a few sample rounds to illustrate some basic and corner cases:

Human's Hand	**AI's Hand**	**Winner**	**Explanation**
A♠ K♠ Q♠ J♠ 10♠	8♠ 8♡ 8♣ 4♡ 4♢	Human	The Human has a Straight Flush (Royal), which is a higher category than the AI's Full House.
5♡ 5♣ 5♢ K♠ K♡	8♠ 8♡ 8♣ 4♡ 4♢	AI	Both players have a Full House, so the ranks of the Threes of a Kind are compared. The AI's Three 8s rank higher than the Human's Three 5s. Note that the ranks of the Pairs do not matter in this case, since two players cannot both have Three of a Kind in the same rank. There are only 4 of each rank in a deck.
7♠ 7♢ 6♠ 6♡ K♣	7♡ 7♣ 6♣ 6♢ 9♠	Human	Both players have Two Pair, so the ranks of the higher Pairs are compared, then the ranks of the lower Pairs. The Human's Pair of 7s ties the AI's Pair of 7s, so we move on to the lower Pairs. The Human's Pair of 6s also ties the AI's Pair of 6s, so we move the on to the remaining cards. The Human's King ranks higher than the AI's 9.
7♠ 7♢ 6♠ 6♡ K♣	7♡ 7♣ 6♣ 6♢ K♠	Human	Like the Two-Pair example above, both players have equally-ranked pairs, so we compare the ranks of the remaining card. In this case, the Human's King also ties the AI's King. In the standard rules, this would be considered a tie, and the pot would be split. However, we use a house rule to break the tie, by comparing the suits. Each player's 7s are the highest cards participating in the hand. Even though the Kings are higher, they are not paired in either hand. The Human's best 7 is in ♠s, which is better than the AI's best 7, which is in ♡s.

[1]The Ace can be either high or low, not both. The hand 4, 3, 2, A, K is *not* a straight.
[2]Scoring by suit is a "house rule" not part of the official rules.

This should illustrate all the relevant cases. If you have questions, you may consult me or online resources. Beware that the online resources can be overwhelming if you're not already familiar with poker. There are many variants of poker, sometimes with different deck compositions, and a variety of betting rules. You may also find yourself swamped in strategy guides when you really only need the rules.