# **Requirements**

## **Introduction :-**

* A program that estimates the chances of each hand winning in poker in a situation described by an input file.
* A computer poker player is a computer program designed to play the game of [poker](https://en.wikipedia.org/wiki/Poker" \o "Poker) against human opponents or other computer opponents. It is commonly referred to as pokerbot or just simply [bot](https://en.wikipedia.org/wiki/Computer_game_bot" \o "Computer game bot).

## **Research :-**

**Getting familiar with game first to design program:**

Poker is any of a number of [card games](https://en.wikipedia.org/wiki/Card_game" \o "Card game) in which [players](https://en.wikipedia.org/wiki/Card_player" \o "Card player) [wager](https://en.wikipedia.org/wiki/Betting_(poker)" \o "Betting (poker)) over which [hand](https://en.wikipedia.org/wiki/Poker_hand" \o "Poker hand) is best according to that specific game's rules in ways similar [to these rankings](https://en.wikipedia.org/wiki/List_of_poker_hands" \o "List of poker hands). Often using a [standard deck](https://en.wikipedia.org/wiki/Standard_52-card_deck" \o "Standard 52-card deck), poker games vary in deck configuration, the number of cards in play, the number [dealt](https://en.wikipedia.org/wiki/Poker_dealer" \o "Poker dealer) face up or face down, and the number [shared by all players](https://en.wikipedia.org/wiki/Community_card_poker" \o "Community card poker), but all have rules that involve one or more rounds of betting. In most modern poker games, the first round of betting begins with one or more of the players making some form of a forced bet (the [blind](https://en.wikipedia.org/wiki/Blind_(poker)" \o "Blind (poker)) or ante). In standard poker, each player bets according to the rank they believe their hand is worth as compared to the other players. The action then proceeds clockwise as each player in turn must either match (or "call") the maximum previous bet, or [fold](https://en.wikipedia.org/wiki/Betting_(poker)" \l "Fold" \o "Betting (poker)), losing the amount bet so far and all further involvement in the hand. A player who matches a bet may also "raise" (increase) the bet. The betting round ends when all players have either called the last bet or folded. If all but one player folds on any round, the remaining player collects the pot without being required to reveal their hand. If more than one player remains in contention after the final betting round, a [showdown](https://en.wikipedia.org/wiki/Showdown_(poker)" \o "Showdown (poker)) takes place where the hands are revealed, and the player with the winning hand takes the pot.

**Gameplay:**

### In casual play, the right to [deal](https://en.wikipedia.org/wiki/Poker_dealer" \o "Poker dealer) a hand typically rotates among the players and is marked by a token called a [dealer button](https://en.wikipedia.org/wiki/Button_(poker)" \o "Button (poker)) (or buck). In a [casino](https://en.wikipedia.org/wiki/Casino" \o "Casino), a house dealer handles the [cards](https://en.wikipedia.org/wiki/Playing_card" \o "Playing card) for each hand, but the button (typically a white plastic disk) is rotated clockwise among the players to indicate a nominal dealer to determine the order of betting. The cards are dealt clockwise around the [poker table](https://en.wikipedia.org/wiki/Poker_table" \o "Poker table), one at a time.

### One or more players are usually required to make [forced bets](https://en.wikipedia.org/wiki/Betting_(poker)" \l "Forced_bets" \o "Betting (poker)), usually either an ante or a blind bet (sometimes both). The dealer [shuffles](https://en.wikipedia.org/wiki/Shuffle" \o "Shuffle) the cards, the player on the chair to his or her right [cuts](https://en.wikipedia.org/wiki/Cut_(playing_cards)" \o "Cut (playing cards)), and the dealer deals the appropriate number of cards to the players one at a time, beginning with the player to his or her left. Cards may be dealt either face-up or face-down, depending on the variant of poker being played. After the initial deal, the first of what may be several [betting](https://en.wikipedia.org/wiki/Betting_(poker)" \o "Betting (poker)) rounds begins. Between rounds, the players' hands develop in some way, often by being dealt additional cards or replacing cards previously dealt. At the end of each round, all bets are gathered into the central pot.

### At any time during a betting round, if one player bets, no opponents choose to call (match) the bet, and all opponents instead fold, the hand ends immediately, the bettor is awarded the pot, no cards are required to be shown, and the next hand begins. This is what makes [bluffing](https://en.wikipedia.org/wiki/Bluff_(poker)" \o "Bluff (poker)) possible. Bluffing is a primary feature of poker, one that distinguishes it from other vying games and from other games that make use of [poker hand rankings](https://en.wikipedia.org/wiki/Hand_rankings" \o "Hand rankings).

### At the end of the last betting round, if more than one player remains, there is a [showdown](https://en.wikipedia.org/wiki/Showdown_(poker)" \o "Showdown (poker)), in which the players reveal their previously hidden cards and evaluate their hands. The player with the best hand according to the poker variant being played wins the pot. A poker hand comprises five cards; in variants where a player has more than five cards available to them, only the best five-card combination counts. There are 10 different kinds of poker hands such as straight flush, four of a kind etc.

**Future Scope:**

The subject of online cpu player and computer assistance, while playing online poker, is very controversial. Player opinion is quite varied when it comes to deciding which types of computer software fall into the category unfair advantage. One of the primary factors in defining a cpu bot is whether or not the computer program can interface with the poker client (in other words, play by itself) without the help of its human operator. Computer programs with this ability are said to have or be an autoplayer and are universally defined to be in the category of bots regardless of how well they play poker.

The issue of unfair advantage has much to do with what types of information and artificial intelligence are available to the [computer program](https://en.wikipedia.org/wiki/Computer_program" \o "Computer program). In addition, cpu bots can play for many hours at a time without human weaknesses such as [fatigue](https://en.wikipedia.org/wiki/Fatigue_(medical)" \o "Fatigue (medical)) and can endure the natural variances of the game without being influenced by human [emotion](https://en.wikipedia.org/wiki/Emotion" \o "Emotion) (or "[tilt](https://en.wikipedia.org/wiki/Tilt_(poker)" \o "Tilt (poker))"). On the other hand, bots have some significant disadvantages - for example, it is very difficult for a bot to accurately read a bluff or adjust to the strategy of opponents the way humans can.

## **Cost and Features**

Feature | Cost | Time Create Patient Record |

## **4. Defining Our System :-**

## To get started, take a look at cards.h. You will see that it starts by definingan enum suits (SPADES, HEARTS, DIAMONDS, and CLUBS). This enum also has NUM\_SUITS, which will have a numeric value of 4 (indicating how many suits there are), and can also be used to indicate an invalid suit.

## Next, you will see a struct for a card. This struct has two parts, a value (2,3,4,5,6,7,8,9,10,J,Q,K,A) and a suit (s,h,d,c). Following the struct declaration, there are some #defines for constants for the values of Ace, King, Queen, and Jack. Accordingly, a card's value should be between 2 and 14 (inclusive). There is also an enum for the hand ranking (what kind of poker hand you get). We won't be doing anything with these at this point, except for writing a function to convert from the enumerated values to a string.

## Last are some function prototypes. You will write each of these in cards.c. Now go into cards.c, and write each of these functions. Here are the specifics:

## - void assert\_card\_valid(card\_t c);

This function should use assert() to check that the card passed in has valid values. In particular, its value should be between 2 and VALUE\_ACE (inclusive of both), and its suit should be between SPADES and CLUBS (inclusive of both).

**- const char \* ranking\_to\_string(hand\_ranking\_t r);**

This function should convert the hand\_ranking\_t enumerated value passed in to a string that describes it. Remember that Drew showed you a nice way to do this with emacs keyboard macros!

**- char value\_letter(card\_t c);**

This function should return the character that textually represents the value of the passed-in card. For values 2-9, this should be that digit. For 10, it should be '0', and for Jack, Queen, King, and Ace, it should be 'J', 'Q', 'K', and 'A' respectively. Hint: remember everything is a number.

For example, the character '0' has the decimal value 48, and the character '5' has the decimal value 53, so you could represent '5' as '0' + 5.

**- char suit\_letter(card\_t c);**

This function should return the letter that textually represents the suit of the card passed in ('s', 'h', 'd', or 'c' for SPADES, HEARTS, DIAMONDS, or CLUBS).

**- void print\_card(card\_t c);**

This function should print out the textual representation of the card (hint: use the functions you previously wrote). It should print the value first, then the suit. For example,

As (for Ace of spades)

0d (for 10 of diamonds)

Kc (for King of clubs) etc.

This function should not print any additional spaces or newlines after the card's text.

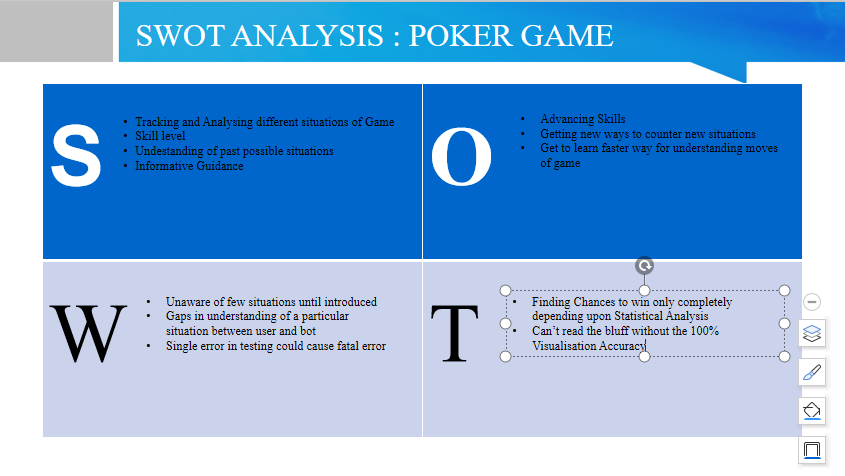
**- card\_t card\_from\_letters(char value\_let, char suit\_let);**

This function should make and return a card\_t whose value and suit correspond to the letters passed in. If the values passed in are invalid, you should use assert() or print an error message and exit(EXIuT\_FAILURE).

**- card\_t card\_from\_num(unsigned c);**

This function should take a number from 0 (inclusive) to 52 (exclusive) and map it uniquely to a card value/suit combination. Exactly how you map the numbers to values/suits is up to you, but you must guarantee that each valid value/suit combination corresponds to exactly one input value in the range [0,52). Hint: you may want to use the mod operator % to find the remainder of a number divided by 13.

## **SWOT ANALYSIS**



# **4W's and 1'H**

## **Who:**

* Players who will require to know the chances of their winning for every situation with the cards they are dealt with.

**What :**

* Many Players make the mistake of reading the cards strength they are dealt with which makes them lose higher with bad strength of cards or win lower with good strength of cards. So this Program was developed for the players to develop their skills for gameplay.

## **When:**

* The Problem started when rookies started loosing too much to the pros and still kept playing even though they understood so little of what they are playing. So the first ever computer program named Polaris was developed in 2007 to play against human professionals where program played while calculating it’s chances of winning against the professionals with the cards it was dealt with.

## **Where:**

* This program was used in most of the high level casinos in countries like US, UK, Japan, etc.

## **How:**

* There were too many players who were not able to understand the situation for a particular gameplay with the cards they were dealt with and the opponents they had to play against and the amount they should play with every hand.

# **Detail requirements**

## **High Level Requirements:**

**TBD**

## **Low level Requirements:**

**TBD**