Version 5 Pseudocode

Enum gameResult = lose, win, draw

Main
{
Set random number seed
define 10 structures each to hold player and dealer cards
Define structure to hold win results
Define variables to hold card value totals, cards in play, score, checks, and choices

Draw dealer cards
Increment number of cards in play
Check for/replace repeat cards
If an ace is drawn, make its value 11 if it doesn't bust the dealer

Draw player cards
Increment number of cards in play
Check for/replace repeat cards

Output dealer's first card and player's two cards
If the player draws a(n) ace(s) gets their choice for the aces value(only allows 1 or 11)
Gets the player's choice to hit or stay with input validation

If stay is chosen,
    Calculates player card total
    Reveals dealer's second card
    If dealer's card total is <= 16,
        draws cards, iterates cards in play, and checks
        for/replaces repeats until a total of >16 is achieved
        If dealer draws ace, sets it to 11 if it doesn't bust the dealer
    Calculates dealer card total

If hit is chosen, loops until stay is chosen
    Calculates player card total
    Draw card and iterate cards in play
    Checks for/replaces repeat cards

    Display card drawn
    If ace is drawn, get choice for ace value(only allows 1 or 11)
    Add card value to player card total

    If player card total >= 21, set choice to stay
    If player card total < 21, get choice to hit or stay(with input validation)

    Reveals dealer's second card
    If dealer's card total is <= 16,
        draws cards, iterates cards in play, and checks
        for/replaces repeats until a total of >16 is achieved
        If dealer draws ace, sets it to 11 if it doesn't bust the dealer
    Calculates dealer card total

Find difference of dealer and player total from 21
Find and output game results based on difference

Define character arrays to hold result message and game name
If win result is win, message is winner
If win result is lose, message is loser
If win result is draw, message is draw

Write the result message to a binary file
Write the game name to another binary file
Write the first card information into another binary file

Output the game name

Free up the used memory and exit the program

}

getCard

{

Get random number between 1 and 13

Get random number between 1 and 4

Assign card name and value to first number drawn

1 = ace, with value 1

2 = two, with value 2

3 = three, with value 3

4 = four, with value 4

5 = five, with value 5

6 = six, with value 6

7 = seven, with value 7

8 = eight, with value 8

9 = nine, with value 9

10 = ten, with value 10

11 = jack, with value 10

12 = queen, with value 10

13 = king, with value 10

Assign card suit to second number drawn

1 = spades

2 = clubs

3 = diamonds

4 = hearts

}

check
{
Compare card passed in with every card played before it
If card passed has been played before, return -1
Otherwise return 1
}

aceGet
{
Prompt user to choose value 1 or 11 for ace
If neither 1 or 11 is entered, output error message and prompt another input
Return the value chosen
}

binaryFileManip
{
Define pointer to hold a char array
Create file stream for input and output to a binary file
Go to start of file
Write the passed array to the file
Go back to start of file
Read the char array from the file
Close the file

}

results
{
define structure to return
Define lose message to display

Output dealer and player totals
If both dealer and player total difference are 0, output no winner and the result is recorded as draw
If dealer and player total differences are the same, output no winner and the result is recorded as draw
If both player and dealer total difference is negative, output no winner and the result is recorded as draw
If dealer total is negative and player total isn't, output win message and the result is recorded as win
If player total is negative and the dealer total isn't, output lose message and the result is recorded as lose
If dealer's total is smaller than player's and is not negative, output lose message and the result is recorded as lose
If player's total is smaller than the dealer's and is not negative, output win message and the result is recorded as win

Return the structure holding the results
}

binaryRecord
{
Create file stream to write to a binary file
Go to the start of the file
Write the passed structure to the file
Close the file
}