Program 2

# Problem Statement

The given task was to create a multiprocessing application that studies various strategies for the card game twenty-one. The program was to be written C/C++ and have the manager process be the dealer and the worker processes be the players. At least one thousand trials are to be run and report the number of times the dealer won and the number of times the players won. The code is meant to be tested in the previously created Ubuntu LTS environment.

# Approach

To create the program C++ was used. Classes were created to represent cards, a deck, and players. The use of a pipe and forking was used for multiprocessing. Then to implement the multiprocessing different functions were created to simulate the dealer and players. Two separate functions were created to simulate two different strategies for each player. Finally, the program was run in the Ubuntu environment to ensure everything worked properly.

# Solution

The task was to implement twenty-one and have two players with different strategies and keep track of wins.

Fork was used to create child and the parent process. Then a pipe was used to send data between the child and parent processes. Each child processes calculated a player hand, and the parent calculated the dealer’s hand and checked to see which hands won. An array was created to store the hand values and passed through the pipe to get the data to the parent.

A vector was used to hold the dynamic hands of the dealer and players. An array of object cards was created for the deck. A for loop was used to keep track of the number of trials. Standard twenty-one rules were implemented such as hitting, busting, standing, and an ace being worth one or eleven. Finally, integer variables were used to keep track of wins.

# Strategies

After examining the rules and researching blackjack and twenty-one strategies I decided on two.

The first strategy is a very conservative and passive strategy. In short, if the player’s hand value is over is 11, we stand and if it is below 11 it draws another card. This to avoid busting as there are a lot of cards with a value of 10. This simple strategy just mainly relies on having a good start or the dealer busting.

The second strategy I created by simplifying strategies I have seen online. The strategy just follows the rule that if the player’s hand is less than 16 to draw another card. This strategy tries to get a high enough number and minimizing the chance of the player busting.

# Problems

There were no instructions on how to handle a tie occurring. As a result, ties were not tracked or reported.



Figure 1 Win Tracking

The instructions did not specify if the win tracking is desired against each player or total wins. In the implemented program each hand of the dealer plays against both strategies of the player. Then as shown in **Figure 1** I tracked how each hand performed against the two player hands.