**ICM Coding Exercise:**

OS: Cent OS

Programming language: C

1. install gcc compiler:

**sudo yum group install "Development Tools"**

1. Verifiy Version:

**gcc -v**

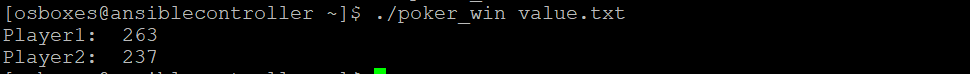
1. copy poker\_win.c in home directory:
2. compile to get binary:

**cc poker\_win.c -o poker\_win**



1. Execute binary with input file:

**./poker\_win <input\_FileName>**



-----------------

**Data Structure for each player:**

struct card

{

char suit;

int value;

};

struct player

{

int rank;

int reducedList[5];

struct card gameCard[5];

};

struct player gamePlayer[2];

**Function for each combination:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Rank** | **Combination** | **Description** | **Function** |
| 1 | High card | Highest value card | is\_highcard() |
| 2 | Pair | Two cards of same value | is\_onepair() |
| 3 | Two pairs | Two different pairs | is\_twopair() |
| 4 | Three of a kind | Three cards of the same value | is\_threeofkind() |
| 5 | Straight | All five cards in consecutive value order | is\_straight() |
| 6 | Flush | All five cards having the same suit | is\_flush() |
| 7 | Full house | Three of a kind and a Pair | is\_fullhouse() |
| 8 | Four of a kind | Four cards of the same value | is\_fourofkind() |
| 9 | Straight flush | All five cards in consecutive value order, with the same suit | is\_straight\_flush() |
| 10 | Royal Flush | Ten, Jack, Queen, King and Ace in the same suit | is\_royal\_flush() |

**Sudo Code:**

Read the input file:

For each Line in input file;

Start loop:

Populate gameplayer[0..1]. gamecard[0..4].value ;

Populate gameplayer[0..1]. gamecard[0..4].suit;

Sort gameplayer[0..1]. gamecard[0..4].value; // Descnding order

Call **Combination function**

Populate gameplayer[0..1].rank;

Populate gameplayer[0..1].reducedList[0..4];

Compare rank;

If rank of Playe1 == Player ==2

Compare the gameplayer[0].reducedList[0..4] and gameplayer[1].reducedList[0..4]

Update player1 / Player2 counter corrodingly

End loop

Print Result;