Algorithm and Problem Solving Lab

Project
(15B17CI471)
Black Jack Game

Submitted By:

1. Deepak Kumar Goyal (9920103125)
2. Sarthak Pant (9920103126)
3. Aastha Garg (9920103139)

Submitted To:

Dr.Shikha K Mehta

Department of CSE/IT

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY UNIVERSITY,NOIDA

MAY 2022

Page 2: **Table of Contents**

BLACK JACK:-

Problem Statement:

Designing a virtual simulator of multi-user operated blackjack card game consisting of dealer, player and compiler assistance.

Reason/Motivation to choose the topic:

Curiosity to implement HUFFMAN CODING in the game of luck “BLACKJACK” .

Objective ,scope of the project and methodology:

At a blackjack table, the dealer faces five to nine playing positions from behind a semicircular table. Between one and eight standard 52-card decks are shuffled together. To start each round, players place bets in the "betting box" at each position. In jurisdictions allowing back betting, up to three players can be at each position. The player whose bet is at the front of the betting box controls the position, and the dealer consults the controlling player for playing decisions; the other bettors "play behind". A player can usually control or bet in as many boxes as desired at a single table, but an individual cannot play on more than one table at a time or place multiple bets within a single box. In many U.S. casinos, players are limited to playing one to three positions at a table.

The dealer deals from their left ("first base") to their far right ("third base"). Each box gets an initial hand of two cards visible to the people playing on it. The dealer's hand gets its first card face up, and, in "hole card" games, immediately gets a second card face down (the hole card), which the dealer peeks at but only reveals when it makes the dealer's hand a blackjack. Hole card games are sometimes played on tables with a small mirror or electronic sensor used to peek securely at the hole card. In European casinos, "no hole card" games are prevalent; the dealer's second card is not drawn until the players have played their hands.

Dealers deal the cards from one or two handheld decks, from a dealer's shoe, or from a shuffling machine. Single cards are dealt to each wagered-on position clockwise from the dealer's left, followed by a single card to the dealer, followed by an additional card to each of the positions in play. The players' initial cards may be dealt face up or face down (more common in single-deck games).

The object of the game is to win money by creating card totals higher than those of the dealer's hand but not exceeding 21, or by stopping at a total in the hope that the dealer will bust. On their turn, players choose to "hit" (take a card), "stand" (end their turn and stop without taking a card), "double" (double their wager, take a single card, and finish), "split" (if the two cards have the same value, separate them to make two hands), or "surrender" (give up a half-bet and retire from the game).

Number cards count as their number, the jack, queen, and king ("face cards" or "pictures") count as 10, and aces count as either 1 or 11 according to the player's choice. If the total exceeds 21 points, it busts, and all bets on it immediately lose.

After the boxes have finished playing, the dealer's hand is resolved by drawing cards until the hand achieves a total of 17 or higher (a dealer total of 17 including an ace valued as 11, also known as a "soft 17", must be drawn to in some games and must stand in others). The dealer never doubles, splits, or surrenders. If the dealer busts, all remaining player hands win. If the dealer does not bust, each remaining bet wins if its hand is higher than the dealer's and loses if it is lower.

A player total of 21 on the first two cards is a "natural" or "blackjack," and the player wins immediately unless the dealer also has one, in which case the hand ties. In the case of a tie ("push" or "standoff"), bets are returned without adjustment. But a blackjack beats any hand that is not a blackjack, even one with a value of 21.

Wins are paid out at even money, except for player blackjacks, which are traditionally paid out at 3 to 2 odds. Many casinos today pay blackjacks at less than 3:2. This is common in single-deck blackjack games.[12]

Blackjack games usually offer a side bet called insurance, which may be placed when the dealer's face up card is an ace. Additional side bets, such as "Dealer Match" which pays when the player's cards match the dealer's up card, are also sometimes available.

Components and Data Structures used:

1. Linked List
2. Hash Map
3. File Handling
4. Huffman Encoding

Hardware & Software requirements:

1. IDE with properly setup development environment
2. Intel pentium dual core or any other AMD equivalent

Contribution that the project will be able to make:

Our Project concludes that if multiple users desire to play our blackjack game, one should proceed with the login page. after entering password and username, the user would be redirected to the start otherwise he/she needs to signup and then proceed with login start.Our signup page is encrypted.After the initiation of the game, each user would be awarded with 1400 karmas and it is the decision of HomeGame function to recognize an action which would be hit or stand.The dealer hand is the one standing opposite to the user and in case the dealer wins, the card count is greater than 21 or equal to it. either way, the user is redirected to login/initiation of the game

Implementation And Code Output:

```
#include<iostream>
#include<vector>
#include<fstream>
#include <unordered_map>
#include<map>
#include<random>
#include<time.h>
#include<stdlib.h>
#include<bits/stdc++.h>
using namespace std;
int Karmas=1400;
const int MAX=47;
string username;
int dealernumber=0,usernumber=0,tdealer,tuser;
vector<int> card={2,3,4,5,6,7,8,9,10,11};
char alphabet[MAX] = { 'a', 'b', 'c', 'd', 'e', 'f', 'g',
                       'h', 'i', 'j', 'k', 'l', 'm', 'n',
                       'o', 'p', 'q', 'r', 's', 't', 'u',
                       'v', 'w', 'x', 'y', 'z'};

vector<int> frequency ={14810,2715,4943,7874,21912,4200,3693,
                        10795,13318,188,1257,7253,4761,12666,14003,0
                        ,3316,205,10977,11450,16587,5246,2019,3819,315,128};
```

```
vector<string> encode(26,"");
struct Node {
    int f, i;
    Node *l, *r;
    Node(int f, char i=-1, Node*l=NULL, Node*r=NULL)
        : f(f), i(i), l(l), r(r) {
    }
};
struct NodeCmp {
    bool operator()(const Node* a, const Node* b) {
        return a->f > b->f;
    }
};
void deleteNodeR(Node* node) {
    if (!node) return;
    deleteNodeR(node->l);
    deleteNodeR(node->r);
    delete node;
}
void printR(Node* n, string& s, vector<string>& R) {
    if (n->i > -1) R.push_back(s);
    else {
        s.push_back('0');
        printR(n->l, s, R);
        s.pop_back();
        s.push_back('1');
        printR(n->r, s, R);
        s.pop_back();
    }
}
vector<string> huffmanCodes(string S,vector<int> f,int N)
{
    priority_queue<Node*, vector<Node*>, NodeCmp> Q;
    for (int i = 0; i < N; i++)
        Q.push(new Node(f[i], i));
    while (Q.size() > 1) {
        auto l = Q.top(); Q.pop();
        auto r = Q.top(); Q.pop();
```

```
        Q.push(new Node(l->f + r->f, -1, l, r));
    }
    vector<string> R;
    string s;
    printR(Q.top(), s, R);
    deleteNodeR(Q.top());
    return R;
}
class node{
    public:
    node* next;
    int karmas;
    string username;
    bool quit;
    int card_total;
    node(string user,int kar, bool qu,int card){
        next=NULL;
        karmas=kar;
        username=user;
        quit=qu;
        card_total=card;
    }
};

void insert(node* &head,string user,int kar, bool qu,int card){
    node* temp= new node(user,kar, qu,card);
    if(head==NULL){
        temp->next=temp;
        head=temp;
        return;
    }
    node* t=head;
    while(t->next!=head){
        t=t->next;
    }
    t->next=temp;
    temp->next=head;
    head=temp;
```

```
}

void deletes(node* &head,string name){
    node* temp=head;
    // node* a=NULL;
    while(temp->next->username==name){
        temp=temp->next;
    }
    node* t=temp->next;
    temp->next=t->next;
    free(t);
}
void display(node* head){
    node* temp=NULL;
    while(temp->next!=head){
        cout<<temp->username<<" ";
    }
}

node* multiuser=NULL;

class hand;
class Game;
class Screen;
class logindetails{
    public:
          string getpassowrd(string password){
              int siz=30;
              string s="";
              for(int i=password.size()*4;i<60;++i){
                  char a=alphabet[rand()%MAX];
                  // cout<<a;
                  s=s+a;
              }
              return s;
          }
```

```
        string gethashpassword(string passwords){
            vector<pair<char,string>> hash_map(26);
            // for(int i=0;i<26;++i)
            unordered_map<char,string> umap;
            // cout<<"hello"<<endl;
            for(int i=0;i<26;++i){
                char x= 'a'+i;
                umap[x]=encode[i];
                // cout<<1;
            }
            string g="";
            int n= passwords.size();
            for(int i=0;i<n;++i){
                string x=umap[passwords[i]];
                g=g+x;
                // cout<<"check";
            }
            // cout<<"i was here";
            // cout<<g;
            return g;
        }
    int Login(){
                    // user name check weather exist or not;
    // if not add if yes go for passowrd or say already exist
    fstream  fil;

    fil.open("logindetains.txt",ios::out|ios::in);


cout<<"\n\t\t\t_____________________________________________________________________
_____________\n";
    cout<<"\n\t\t\t|\t\t\t\t\tLOGIN___PAGE\t\t\t\t\t  |";

cout<<"\n\t\t\t_____________________________________________________________________
_____________\n";


    string password;
```

```
    string checkusername,checkpassword;
    cout<<"\n\t\t\t\t\t\t\t\tENTER USERNAME\n\n";
    cout<<"\t\t\t\t\t\t\t\t";
    cin>>username;

cout<<"\t\t\t_____
_____\n";

    cout<<"\n\t\t\t\t\t\t\t\tENTER PASSWORD\n\n";
    cout<<"\t\t\t\t\t\t\t\t";
    cin>>password;

cout<<"\t\t\t_____
_____\n";

    int t=1;
    int flag=1;//1 for not getiing user name
    while(getline(fil,checkusername)){
      if(t%3==1){
        if(checkusername==username){
          // flag=0;
          string x="";
          getline(fil,x);
          getline(fil,checkpassword);
          // string
          password=gethashpassword(password);
          password+=x;
          if(password==checkpassword){
            // cout<<"\tPassowrd matched\n";
            flag=0;
            break;
          }

          t+=2;
          break;
        }
        else{
```

```
                t+=3;
            }
        }
        else{
            continue;
        }
    }
    fil.close();
    if(flag==0){
        cout<<"\n\t\t\t\t\t\t\tLOGIN CREDENTIAL MATCHED\n";

cout<<"\n\t\t\t_____
_____\n";

        fstream kar;
        kar.open(username+".txt",ios::out|ios::in);
        kar>>Karmas;
        kar.close();
        // WelcomeScreen();
    }
    else{
    //
cout<<"\n\t\t_____
_____\n";

        cout<<"\n\n\t\t\t\t\t\t\tLOGIN CREDENTIAL INVALID\n\n\t\t\t\t\t\tTRY SIGNING UP
(PRESS 0 TO SIGN UP ELSE 1) \n";
        int a;
        cout<<"\n\t\t\t\t\t\t\t";
        cin>>a;
        if(a==0)
        signup();
        // clrscr();
        else
          return 0;
        //  Welcomepage;
        // system("CLS");
```

```
cout<<"\t\t_____
_____\n";


    }
    return 0;

  }

  int signup(){

cout<<"\n\t\t\t_____
_____\n";
    cout<<"\n\t\t\t|\t\t\t\t\tSIGNUP___PAGE\t\t\t\t\t  |";

cout<<"\n\t\t\t_____
_____\n";

                    fstream fi;
    fi.open("logindetains.txt",ios::out | ios::in | ios::app);
    string password,cu,cp; //cu confirm username,confirm passowd
    cout<<"\n\t\t\t\t\t\t\t\tENTER USERNAME\n\n";
    cout<<"\t\t\t\t\t\t\t\t";
    cin>>username;

cout<<"\t\t\t_____
_____\n";

    cout<<"\n\t\t\t\t\t\t\t\tENTER PASSWORD\n\n";
    cout<<"\t\t\t\t\t\t\t\t";
    cin>>password;

cout<<"\t\t\t_____
_____\n";
    cout<<"\n\t\t\t\t\t\t\t\tCONFIRM PASSWORD\n\n";
    cout<<"\t\t\t\t\t\t\t\t";
    cin>>cp;
```

```
cout<<"\t\t\t_____
_____\n";


    int t=1;
    int flag=1;
    if(password!=cp){
        flag=0;
    }
    else
    while(getline(fi,cu)){
        if(t==1){
            if(username==cu){
                flag=0;//username same at flag =0
                break;
            }
            t=2;
        }
        else{
            t=1;
        }
    }

    fi.close();

    fi.open("logindetains.txt",ios::out | ios::app);



    if(flag==1){
        fi<<"\n"<<username<<"\n";
        // fi<<password;

        string support_password=getpassowrd(password);
        // cout<<support_password.size();
        support_password=gethashpassword(support_password);
        fi<<support_password<<"\n";
```

```
          string  password_temp=gethashpassword(password);
          // cout<<password_temp;
          string final_password=password_temp+support_password;
          fi<<final_password;
          fi.close();
          cout<<"\n\t\t\t\t\t\t\t\tUSER CREATED\n\n";

cout<<"\t\t\t_____
_____\n";

          ofstream makefileuser;
          makefileuser.open(username+".txt");
          makefileuser<<1400;
          makefileuser.close();
       }
       else{
         fi.close();
         cout<<"\n\t\t\t\t\t\t\t\n INCORRECT CREDENTIALS\n\n";

cout<<"\t\t\t_____
_____\n";
       }
       // WelcomeScreen()
       return 0;
   }
};
class hand:public logindetails{
   public:
   // const int b=0;
   int Dealers_Hand(int a){
       // int t;
       if(a==1){

cout<<"\n_____
_____\n";
       dealernumber=0;
       usernumber=0;
```

```
      //
cout<<"\n*****************************************************************************************
**********************\n";
      cout<<"\n\n\t\t\t\t\t\t\t\tThis is dealers Hand\t";
      // srand(time(0));
      tdealer=rand()%10;
      dealernumber+=card[tdealer];
      if(tdealer+dealernumber<=17){
         cout<<"\t"<<card[tdealer];

      }
      cout<<"!";
      return 0;
      }
      else{
         while(dealernumber<=17){

cout<<"\n_____________________________________________________________________________
_____________________\n";

           //
cout<<"\n*****************************************************************************************
**********************\n";
            //
cout<<"\n*****************************************************************************************
**********************\n";

            cout<<"\n\t\t\t\t\t\t\t\tThis is dealers Hand\t";
            // srand(time(0));

            tdealer=rand()%10;
            dealernumber+=card[tdealer];
            cout<<card[tdealer];

         }
         return 0;
      }
   }
```

```
  int UserHand(int n){
     node* temp=multiuser;
     int t=0;
     while(t<n){
        if(temp->quit==false){

cout<<"\n_____
_____\n";

cout<<"\n**********************************************************************************
**********\n";
            cout<<"\n\tThis is "<<temp->username<<" hand\t";
            // srand(time(0));

            tuser=rand()%10;
            temp->card_total+=card[tuser];
            // temp->card_total=usernumber;

            cout<<"\n\tYour new card is \t"<<card[tuser]<<"\n";
            cout<<"\n\tYour  previous total was \t"<<temp->card_total-card[tuser]<<"\n";

            // cout<<"\n\tYour total card is \t"<<temp->card_total<<"\n";


            int a;

            cout<<"\n"<<temp->username<<"\tHIT press 1 and for STAND 2\t";
               cin>>a;
               if(a!=1){
                  // srand(time(0))
                  temp->quit=true;
                  t++;
               }
               if(temp->card_total>21){
                  temp->quit=true;
                  t++;
               }
```

```
        }
        // cout<<"lol";
        temp=temp->next;
      }
      return 0;
   }
};

class Game:public hand{
   public:


   int Home_game(){



      //
cout<<"\n*****************************************************************************************
**********************\n";
      // cout<<"\nYou have 1400 Karmas\n";
      // cout<<"Enter userna"
      int num;
      cout<<"ENTER THE NUMBER OF USER YOU WANT TO PLAY WITH\t";
      cin>>num;
      int flag=0;

      for(int i=0;i<num;++i){
         Login();

cout<<"\n_____________________________________________________________________________________
_____________________________________\n";

cout<<"\n_____________________________________________________________________________________
_____________________________________\n";
         fstream fil;
         fil.open(username+".txt",ios::out);
         // if(fil.is_open()){
```

```
        //     cout<<"\n opened";
        // }
        // else
        // cout<<"\nnot opened";
        // while(flag==0){
        // cout<<"\nDealers hand\n";
        cout<<"\nyour current karmas are\t"<<Karmas;
        if(Karmas==0){
            cout<<"\nYOUR KARMAS ARE 0 SO YOU ARE REQUESTED TO CONTACT ONE OF
OUR DEV AND PAY THEM 1Cr(negotiable) TO MAKE IT AGAIN 1400 BOOMER!!!!\n";
            return 0;
        }
        insert(multiuser,username,Karmas,false,0);
        fil.close();

    // }
    }
    Dealers_Hand(1);
    UserHand(num);
    // cout<<"notworking1";
    Dealers_Hand(2);
    // display(multiuser);
    // cout<<"notworking2";
    node* temp=multiuser;
    int tep=num;
    while(num--){
        if(temp->card_total >dealernumber && temp->card_total<=21 )
        {


            cout<<"\n\n\t\t\t\t\t\t\t\tYOU won!!!";
            temp->karmas=temp->karmas+ 20;
            // fil<<Karmas;
            //   cout<<"\n\n\t\t\t\t\t\t\t\tIF YOU WANT TO UPDATE YOUR KARMAS LOGIN OR
SIGHUP";
            cout<<"\n\n\t\t\t\t\t\t\t\t"<<temp->username<<" Your karmas are
\t"<<temp->karmas;
            // cout<<"\nWANT TO CONTINUE IF YES TYPE 0 ELSE 1\t";
```

```
                // cin>>flag;
                usernumber=0;
                dealernumber=0;
            }
            else
            {
                cout<<"\n\n\t\t\t\t\t\t\t\t\t"<<temp->username<<"YOU Lose!!";
                temp->karmas=temp->karmas-50;
                // fil<<Karmas;
                cout<<"\n\n\t\t\t\t\t\t\t\t\tYour karmas are \t"<<temp->karmas;
                // cout<<"\nWANT TO CONTINUE IF YES TYPE 0 ELSE 1\t";
                // cin>>flag;
            }
            temp=temp->next;
        }
        // cout<<"lol";

            temp=multiuser;

            while(tep--){
                // cout<<"saing..";
                fstream fil;
                fil.open(temp->username+".txt",ios::out | ios::in);
                fil<<temp->karmas;
                temp=temp->next;
                fil.close();
            }
            // deletes(multiuser,temp->username);




cout<<"\n_____
_____\n";

cout<<"\n_____
_____\n";
```

```
    return 0;

    }


};

class Screen:public Game{


    public:



    void Rules(){
          // cout << "_____*" << endl;

cout<<"\n\t\t\t_____
_____\n";
          cout<<"\n\t\t\t|\t\t\t\t\tRULES||INSTRUCTION\t\t\t\t\t   |";

cout<<"\n\t\t\t_____
_____\n";

                                cout << "\n\t\t\t|\t\t\t\t# How to play the game of Blackjack. ";
                                cout << "\n\t\t\t|\t\t\t\tThere are two players: a dealer, ";
                                cout << "\n\t\t\t|\t\t\t\tplayed by a computer, ";
                                cout << "\n\t\t\t|\t\t\t\tand a player, played by you. ";
                                cout << "\n\t\t\t|\t\t\t\tThe game will be played as many ";
                                cout << "\n\t\t\t|\t\t\t\trounds as the player can or wants, ";
                                cout << "\n\t\t\t|\t\t\t\tand the winner is determined ";
                                cout << "\n\t\t\t|\t\t\t\teach round.";
                                cout << "\n\t\t\t|\t\t\t\tYou, the player, start with 1400 points and ";
                                cout << "\n\t\t\t|\t\t\t\tcan bet at least     each round. ";
                                cout << "\n\t\t\t|\t\t\t\tThe maximum number of   a player can bet ";
```

```
                    cout << "\n\t\t\t|\t\t\tat each round is set at "  ;
                    cout << "\n\t\t\t|\t\t\there. The dealer is assumed to have ";
                    cout <<   "\n\t\t\t|\t\t\tin the beginning. ";
                    cout << "\n\t\t\t|\t\t\tIf either the player or the dealer loses all ";
                    cout << "\n\t\t\t|\t\t\t , the game ends.\n\n";
                    cout << "\n\t\t\t|\t\t\tAt each round, the objective of the player ";
                    cout << "\n\t\t\t|\t\t\tis to win the bet by creating a card total ";
                    cout << "\n\t\t\t|\t\t\tthat is higher than the value of ";
                    cout << "\n\t\t\t|\t\t\tthe dealer's hand, but not exceeding 21 ";
                    cout << "\n\t\t\t|\t\t\t(called, \"busting\"). ";
                    cout << "\n\t\t\t|\t\t\tThe value of a hand is determined by summing
";
                    cout << "\n\t\t\t|\t\t\tover values of all ";
                    cout << "\n\t\t\t|\t\t\tcards in a hand: 2~10 have the same values ";
                    cout << "\n\t\t\t|\t\t\tas the face values, ";
                    cout << "\n\t\t\t|\t\t\twhile J, Q, and K (face cards) are counted ";
                    cout << "\n\t\t\t|\t\t\tas 10 and an ace, A, ";
                    cout << "\n\t\t\t|\t\t\tcan be counted as 1 or 11. The suits of the ";
                    cout << "\n\t\t\t|\t\t\tcards don't have any meaning.";
                    cout << "\n\t\t\t|\t\t\tOnce the amount of the bet is chosen for ";
                    cout << "\n\t\t\t|\t\t\teach round, ";
                    cout << "\n\t\t\t|\t\t\ttwo cards are dealt at the beginning of the ";
                    cout << "\n\t\t\t|\t\t\tround: both cards of the player are revealed,";
                    cout << "\n\t\t\t|\t\t\twhile only one card is revealed for the ";
                    cout << "\n\t\t\t|\t\t\tdealer. ";
                    cout << "\n\t\t\t|\t\t\tThe player has two options: Hit or Stand.\n";
                    cout << "\n\t\t\t|\t\t\t(1) Hit: Take another card from the dealer.";
                    cout << "\n\t\t\t|\t\t\tIf the player's hand ";
                    cout << "\n\t\t\t|\t\t\tis not busted by exceeding 21, ";
                    cout << "\n\t\t\t|\t\t\tthe player has another chance ";
                    cout << "\n\t\t\t|\t\t\tto choose to hit or stand,\n\n";
                    cout << "\n\t\t\t|\t\t\t(2) Stand: Take no more card. ";
                    cout << "\n\t\t\t|\t\t\tThen, the player's value is ";
                    cout << "\n\t\t\t|\t\t\tdetermined by summing over all cards in ";
                    cout << "\n\t\t\t|\t\t\tthe hand (An ace, A, can be either 1 or 11, ";
                    cout << "\n\t\t\t|\t\t\twhichever is better).\n\n";
                    cout << "\n\t\t\t|\t\t\tIf the player gets busted by exceeding 21, ";
                    cout << "\n\t\t\t|\t\t\tthe dealer wins. If the player choose to ";
```

```
                        cout << "\n\t\t\t|\t\t\tstand at a value 21 or lower, ";
                        cout << "\n\t\t\t|\t\t\tthe dealer should hit until the value is ";
                        cout << "\n\t\t\t|\t\t\t17 or greater (the ace, A, is counted as 11 ";
                        cout << "\n\t\t\t|\t\t\tas long as the sum is less than 21, ";
                        cout << "\n\t\t\t|\t\t\teven when the sum becomes 17, which is ";
                        cout << "\n\t\t\t|\t\t\tcalled \"S17\" rule). ";
                        cout << "\n\t\t\t|\t\t\tIf the dealer gets busted, the player wins. ";
                        cout << "\n\t\t\t|\t\t\tIf both are not busted, ";
                        cout << "\n\t\t\t|\t\t\tthe winner is determined by comparing values;";
                        cout << "\n\t\t\t|\t\t\t the player wins ";
                        cout << "\n\t\t\t|\t\t\tif the player's value is greater, and the ";
                        cout << "\n\t\t\t|\t\t\tdealer wins if the dealer's value is greater.";
                        cout << "\n\t\t\t|\t\t\tIf tied, the bet is returned to the player.";
                        cout << "\n\t\t\t|\t\t\tIf the first two cards has the value ";
                        cout << "\n\t\t\t|\t\t\t21 by having an ace and ";
                        cout << "\n\t\t\t|\t\t\ta 10-valued card (10 or J or Q or K), ";
                        cout << "\n\t\t\t|\t\t\tit's called the \"Blackjack\" and ";
                        cout << "\n\t\t\t|\t\t\twins every hand except another blackjack (if ";
                        cout << "\n\t\t\t|\t\t\tboth get blackjacks, it's a tie).";
                        cout << "\n\t\t\t|\t\t\t# Card representation.";
                        cout << "\n\t\t\t|\t\t\tThe ranks: A (ace), 2, 3, 4, 5, 6, 7, 8, 9, ";
                        cout << "\n\t\t\t|\t\t\t10, J, Q, K.\n";
                        cout << "\n\t\t\t|\t\t\tThen, for example, A(s) stands for the spade ";
                        cout << "\n\t\t\t|\t\t\tace, 10(d) stands for the diamond 10, ";
                        cout << "\n\t\t\t|\t\t\tand Q(h) stands for the heart queen.";
                        cout << "\n\t\t\t|\t\t\t# Player inputs.";
                        cout << "\n\t\t\t|\t\t\tThe player can give inputs using keyboards ";
                        cout << "\n\t\t\t|\t\t\tat the prompt, and only the first character ";
                        cout << "\n\t\t\t|\t\t\t(excluding white spaces) of a line, followed ";
                        cout << "\n\t\t\t|\t\t\tby Enter, will be regarded as a valid input. ";
                        cout << "\n\t\t\t|\t\t\tPossible input characters are: n (new round),";
                        cout << "\n\t\t\t|\t\t\tr (rules), h (hit), s (stand), q (quit), ";
                        cout << "\n\t\t\t|\t\t\tand 1~5 (size of the bet, number of decks).\n";

cout<<"\n\t\t\t_____
_____\n";

        cout<<"\n\t\t\t|\t\t\tPRESS 1 FOR GOING BACK\n";
```

```
        cout<<"\n\t\t\t|\t\t\t\t";
        int a;
        cin>>a;
        if(a==1)
        WelcomeScreen();
    }



    void WelcomeScreen(){
        int choice;
        cout<<"\n\n\t\t\t\t\t\t\tWelcome to BLACKJACK game!\n\n      "<<endl;
        cout<<"\t\t\t-------------------------------------------------------------------------------"<<endl;
        cout<<"\t\t\t-------------------------------------------------------------------------------"<<endl;

cout<<"\t\t\t*******************************************************************************************
********"<<endl;
        cout<<"\t\t\t**\t\t\t\t\t1-Play Game\t\t\t\t\t**"<<endl<<"\t\t\t**\t\t\t\t\t2-Rules and
Instructions\t\t\t**"<<endl<<"\t\t\t**\t\t\t\t\t3-Login\t\t\t\t\t\t**"<<endl<<"\t\t\t**\t\t\t\t\t4-Si
gnup\t\t\t\t\t**"<<endl;

cout<<"\t\t\t*******************************************************************************************
********"<<endl;
        cout<<"\t\t\t-------------------------------------------------------------------------------"<<endl;
        cout<<"\t\t\t-------------------------------------------------------------------------------"<<endl;
        cout<<"\t\t\t**\t\t\t\t\tEnter your choice"<<endl;
        cout<<"\t\t\t  \t\t\t\t\t";
        cin>>choice;
        if(choice==1)
        {
            Home_game();
            WelcomeScreen();
        }
        else if(choice==2)
        {
            Rules();
        }
```

```
        else if(choice==3)
        {
            Login();
            WelcomeScreen();

        }
        else if(choice==4)
        {
            signup();
            WelcomeScreen();

        }
    }


};
int main(){
encode=huffmanCodes("abcdefghijklmnopqrstuvwxyz",frequency,26);
// for(auto it : encode){
//     cout<<it<<endl;
// }
Screen s;
s.WelcomeScreen();
}
```

SCREENSHOTS:

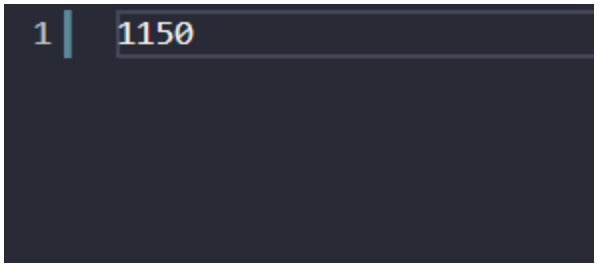
```
your current karmas are 1200
_____________________________________________________________________________________________

                                                        This is dealers Hand           3!
_____________________________________________________________________________________________
*********************************************************************************************************

       This is sarthak hand
       Your new card is        9

       Your  previous total was        0

sarthak HIT press 1 and for STAND 2     1

_____________________________________________________________________________________________
*********************************************************************************************************

       This is sarthak hand
       Your new card is        6

       Your  previous total was        9

sarthak HIT press 1 and for STAND 2     1

_____________________________________________________________________________________________
*********************************************************************************************************

       This is sarthak hand
       Your new card is        2

       Your  previous total was        15

sarthak HIT press 1 and for STAND 2     1

_____________________________________________________________________________________________
```

```
_____________________________________________________________________________________________
*********************************************************************************************************

        This is sarthak hand
        Your new card is        11

        Your  previous total was        17

 sarthak HIT press 1 and for STAND 2     1

_____________________________________________________________________________________________

                                                        This is dealers Hand    6
_____________________________________________________________________________________________

                                                        This is dealers Hand    10

                                                        sarthakYOU Lose!!

                                                        Your karmas are         1150
_____________________________________________________________________________________________

_____________________________________________________________________________________________
```

```
                        Welcome to BLACKJACK game!


--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
********************************************************************************
**                              1-Play Game                                   **
**                              2-Rules and Instructions                      **
**                              3-Login                                       **
**                              4-Signup                                      **
********************************************************************************
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
**                              Enter your choice
                                2


_____________________________________________________________________________

|                               RULES||INSTRUCTION                           |
_____________________________________________________________________________

|                       # How to play the game of Blackjack.
|                       There are two players: a dealer,
|                       played by a computer,
|                       and a player, played by you.
|                       The game will be played as many
|                       rounds as the player can or wants,
|                       and the winner is determined
|                       each round.
|                       You, the player, start with 1400 points and
|                       can bet at least     each round.
|                       The maximum number of   a player can bet
|                       at each round is set at
|                       here. The dealer is assumed to have
|                       in the beginning.
|                       If either the player or the dealer loses all
|                        , the game ends.


|                       At each round, the objective of the player
|                       is to win the bet by creating a card total
|                       that is higher than the value of
|                       the dealer's hand, but not exceeding 21
|                       (called, "busting").
|                       The value of a hand is determined by summing
```

```
|
|                         If the player gets busted by exceeding 21,
|                         the dealer wins. If the player choose to
|                         stand at a value 21 or lower,
|                         the dealer should hit until the value is
|                         17 or greater (the ace, A, is counted as 11
|                         as long as the sum is less than 21,
|                         even when the sum becomes 17, which is
|                         called "S17" rule).
|                         If the dealer gets busted, the player wins.
|                         If both are not busted,
|                         the winner is determined by comparing values;
|                          the player wins
|                         if the player's value is greater, and the
|                         dealer wins if the dealer's value is greater.
|                         If tied, the bet is returned to the player.
|                         If the first two cards has the value
|                         21 by having an ace and
|                         a 10-valued card (10 or J or Q or K),
|                         it's called the "Blackjack" and
|                         wins every hand except another blackjack (if
|                         both get blackjacks, it's a tie).
|                         # Card representation.
|                         The ranks: A (ace), 2, 3, 4, 5, 6, 7, 8, 9,
|                         10, J, Q, K.
|
|                         Then, for example, A(s) stands for the spade
|                         ace, 10(d) stands for the diamond 10,
|                         and Q(h) stands for the heart queen.
|                         # Player inputs.
|                         The player can give inputs using keyboards
|                         at the prompt, and only the first character
|                         (excluding white spaces) of a line, followed
|                         by Enter, will be regarded as a valid input.
|                         Possible input characters are: n (new round),
|                         r (rules), h (hit), s (stand), q (quit),
|                         and 1~5 (size of the bet, number of decks).
|
-------------------------------------------------------------------------
|
|                         PRESS 1 FOR GOING BACK
```


logindetains.txt
1 dempok
2 00010100110101110101001111001100001010001010001001110001010000110011100110110101010110000010010011111000001010110100111101100010111
3 000101000100010101001010110010100001000000101001101011110101001111001100001010001010001001110001010000110011100110110101010110000010010011111000001010110100111101100010111
4 aastha
5 00010100010011001100110001010110010110010001001001011010001010011010100101110000010100010001001111
6 00000000101010110001011000000010100010011001100010101100101100100010010011010001010011010100101110000010100010001001111
7 deepak
8 00010100110101110101001111001100001010001010001001110001010000110011100110110101010110000010010011111000001010110100111101100010111
9 000101000100010100100010100110010000001000000101001101011101001111001100010100010100011100001010000110011100110110101010110000010010011111000001010110100111101100


1
1150