Министерство образования и науки РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |  |  |
| --- | --- | --- | --- |
| Факультет | | | *Информационных технологий и компьютерных систем* |
|  | | |  |
| Кафедра | *Информатики и вычислительной техники* | | |
|  | |  | |
| Дисциплина | | ***Программирование*** | |

**ЗАДАНИЕ**

**на выполнение курсового проекта**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студенту (ке) | Шкаброву Даниле Андреевичу | | Группа | **ПИ-** |
|  | фамилия, имя, отчество полностью | |  | 182 |
|  |  | |  |  |
| Направление (специальность) | | *09.03.03 – Прикладная информатика* | | |
|  | | код, наименование | | |

|  |  |
| --- | --- |
| Тема проекта | Разработка электронной записной книжки и программы-игры |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Срок сдачи проекта (работы) на кафедру | « 25 » 05 |  | 20 | 19 | г. |

|  |  |
| --- | --- |
| Исходные данные к проекту (работе) | Разработка электронной «Записной книжки» |
| с возможностью поиска данных и программы-игры «Покер» | |
|  | |

|  |
| --- |
| Содержание пояснительной записки (перечень подлежащих разработке вопросов) |
| ***Введение. 1. Постановка задач на проектирование. 2. Методика решения. 3. Разработка алгоритмов. 4. Особенности разработки программы. 5. Инструкция пользователя. 6. Результаты тестирования. Приложение (текст программы)*** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Перечень графического материала с указанием основных чертежей | | | | | | | | | | | | | | | | | | | | | |
| и (или) иллюстративного материала | | | | | | | ***Укрупненная схема алгоритма,*** | | | | | | | | | | | | | |
| ***схемы алгоритмов для функции*** | | | | | | | | | | | | | | | | | | | | |
| Методическая литература и иные информационные источники | | | | | | | | | | | | | |  | | | | | | |
| 1. *Павловская, Т. А. С/С++. Программирование на языке высокого уровня / Т. А. Павловская. – СПб.: Питер, 2012. – 544с.* | | | | | | | | | | | | | | | | | | | | |
| 1. *Восходящее и нисходящее программирование: метод. указания / ОмГТУ ; сост. О. П. Шафеева.– Омск : Изд-во ОмГТУ, 2015. – 1 эл. опт. диск (CD-ROM).* 2. *Программирование на языке Си: Метод. указания / Сост. О. П. Шафеева, Ю.Г. Каворина, Шукурова Г.С.: Изд-во ОмГТУ, 2008. – 72 с.* 3. *Системы программирования: Метод. указания / Сост. О.П. Шафеева, И.А. Волчкова, С.А.Гончаров. − Омск: ОмГТУ, 2012. – 32 с.* 4. *Шафеева, О.П. Технологии программирования. С++. Учебное пособие. – Омск, Изд-во ОмГТУ. 2007. – 80 с.* | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | |
| Дата выдачи задания | « | *5* | | | » | *февраля* | | | | 20 | | 19 | г. | | | | | | | | | | |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Руководитель |  |  | *К.т.н., доцент Шафеева О.П.* |  |  | |  | подпись |  | ученая степень, звание, ФИО |  | дата | | Зав. кафедрой |  |  | *Д.т.н., профессор Потапов В.И.* |  |  | |  | подпись |  | ученая степень, звание, ФИО |  | дата | | | | | | | | | | | | | | | | | | | | | | | | | |
| Задание принял к  исполнению студент (ка) | | |  | | | | |  | | | **« 5** » 02 | | | |  | 20 | 19 | г. |
|  | | | | подпись | | | | |  | | | | | |  |  | | |  |  | | |

Министерство образования и науки РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |
| --- | --- |
| Факультет (институт) | *Информационных технологий и компьютерных систем* |
|  |  |
| Кафедра | *Информатики и вычислительной техники* |
|  |  |

**КУРСОВОЙ ПРОЕКТ**

|  |  |
| --- | --- |
| по дисциплине | ***Программирование*** |
|  |  |
| на тему | Разработка электронной записной книжки и программы-игры |

**Пояснительная записка**

|  |  |
| --- | --- |
| Шифр проекта | **020-КП-09.03.03-28-ПЗ** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Студента (ки) | | | | **Шкаброва Данилы Андреевича** | | | | | | |
|  |  |  |  | | | | фамилия, имя, отчество полностью | | | | | | |
|  |  |  | Курс | ***1*** |  | | | Группа | | **ПИ-182** | | |  |
|  |  |  |  |  | |  | | |  |  | | |  |
|  |  |  | Направление (специальность) | | | | | | | *09.03.03 –* | | | |
|  |  |  | *Прикладная информатика* | | | | | | | | | | |
|  |  |  | код, наименование | | | | | | | | | | |
|  |  |  | Руководитель | | | | *к.т.н., доцент* | | | | | | |
|  |  |  |  | | | | ученая степень, звание | | | | | | |
|  |  |  | *Шафеева О.П.* | | | | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | | | | |
|  |  |  | Выполнил (а) | | | |  | | | | | | |
|  |  |  |  | | | | дата, подпись студента (ки) | | | | | | |
|  |  |  | К защите | | | |  | | | | | | |
|  |  |  |  | | | | дата, подпись руководителя | | | | | | |
|  |  |  | Выполнение и подготовка к защите КП (КР) | | | | | Защита КП (КР) | | | | Итоговый рейтинг | |
|  |  |  |  | | | | |  | | | |  | |
|  |  |  | Проект (работа) защищен (а) с оценкой | | | | | | | |  | | |

Омск 2019

Министерство образования и науки РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Омский государственный технический университет»

**ОТЗЫВ**

**на курсовой проект (работу)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Факультет | | | | | | | *Информационных технологий и компьютерных систем* | | | | | | | | | | |
|  | | | | | | |  | | | | | | | | | | |
| Кафедра | | | *Информатики и вычислительной техники* | | | | | | | | | | | | | | |
|  | | |  | | | | | | | | | | | | | | |
| Дисциплина | | | | ***Программирование*** | | | | | | | | | | | | | |
|  | | | |  | | | | | | | | | | | | | |
| Тема | **Разработка электронной записной книжки и программы-игры** | | | | | | | | | | | | | | | | |
|  | |  | | | | | | | | | | | | | | | |
| Студент | | | **Шкабров Данила Андреевич** | | | | | | | | | | | | | | |
|  | | | фамилия, имя, отчество полностью | | | | | | | | | | | | | | |
| Курс | ***1*** | | | | Группа | | | **ПИ-**182 |  | | | | | | | | |
|  | |  | | |  | | |  |  | | | | | | | | |
| Руководитель | | | | | *к.т.н., доцент Шафеева О.П.* | | | | | | | | | | | | |
|  | | | | | ученая степень, звание, ФИО | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | |
| **Содержание отзыва** | | | | | | | | | | | | | | | | | |
| Спроектирована укрупненная схема алгоритма для проекта реализации | | | | | | | | | | | | | | | | | |
| базы данных «Записная книжка» и звуковой мультипликационной игры | | | | | | | | | | | | | | | | | |
| «Покер» | | | | | | | | | | | | | | | | | |
| Детализированы и разработаны схемы алгоритмов для отдельных функций и | | | | | | | | | | | | | | | | | |
| подзадач. Разработана и отлажена программа, реализующая следующие | | | | | | | | | | | | | | | | | |
| подзадачи: | | | | | | | | | | | | | | | | | |
| - Покер | | | | | | | | | | | | | | | | | |
| - Анимация | | | | | | | | | | | | | | | | | |
| - Меню (графическое, текстовое, в несколько уровней); | | | | | | | | | | | | | | | | | |
| - База данных; | | | | | | | | | | | | | | | | | |
| - Информация об авторе; | | | | | | | | | | | | | | | | | |
| - Тестирование готового программного продукта; | | | | | | | | | | | | | | | | | |
| Оформлена пояснительная записка. | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | |
| **Рейтинговые баллы за выполнение и подготовку к защите курсового проекта (работы)** | | | | | | | | | | | | |  | | | |
| **Заключение о допуске к защите** | | | | | | | | | |  | | | | | | |
|  | | | | | | | | | | | | | | | | | |
| **Руководитель** | | | | | |  | | | | | Дата | « » | | 20 |  | г. | |
|  | | | | | | подпись | | | | |  |  | |  |  |  | |

# Аннотация

К курсовому проекту на тему: «Разработка электронной записной книжки и программы-игры».

Пояснительная записка к курсовому проекту содержит 241 с., 64 рис.

Составлена укрупнённая схема алгоритма для проекта составления электронной записной книжки и программы-игры «Покер». Детализированы и разработаны схемы алгоритмов для отдельных функций и подзадач. Разработана и отлажена программа, реализующая следующие подзадачи:

* информация об авторе;
* электронная записная книжка;
* Программа-игра «Покер»;
* графическая заставка;
* тестирование готового программного продукта.

Ключевые слова: С\С++, электронная база данных, игровая программа, анимация, справочник.

# 

Оглавление

[1. Введение 4](#_Toc9793153)

[2. Анализ задания и постановка задач на проектирование 5](#_Toc9793154)

[3. Разработка алгоритмов 7](#_Toc9793155)

[3.1. Разработка схемы алгоритма функции «main» 7](#_Toc9793156)

[3.2. Разработка схемы алгоритма меню 7](#_Toc9793158)

[3.3. Разработка схемы алгоритма база данных «Записная книжка» 9](#_Toc9793159)

[3.3.1. Разработка схемы алгоритма меню базы данных 10](#_Toc9793160)

[3.3.2. Разработка схемы алгоритма удаления из базы данных 11](#_Toc9793161)

[3.3.3. Разработка схемы алгоритма вывода всей информации 12](#_Toc9793162)

[3.3.4. Разработка схемы алгоритма для поиска человека 13](#_Toc9793163)

[3.3.5. Разработка схемы алгоритма добавления в базу данных 14](#_Toc9793164)

[3.3.6. Разработка схемы алгоритма сохранения информации 15](#_Toc9793165)

[3.4. Разработка схемы алгоритма об авторе 16](#_Toc9793166)

[3.5. Разработка схемы алгоритма анимация 17](#_Toc9793167)

[3.6. Разработка схемы алгоритма главной функции покера 18](#_Toc9793168)

[3.6.1. Разработка схемы алгоритма событий окна 19](#_Toc9793169)

[3.6.2. Разработка схемы алгоритма события нажатия левой кнопкой мыши по окну 19](#_Toc9793170)

[3.6.3. Разработка схемы алгоритма события нажатие крестика 21](#_Toc9793171)

[3.6.4. Разработка схемы алгоритма событий во время игры 21](#_Toc9793172)

[3.6.5. Разработка схемы алгоритма событий во время игры 22](#_Toc9793173)

[3.6.6. Разработка схемы алгоритма проверка тройной комбинации 23](#_Toc9793174)

[3.6.7. Разработка схемы алгоритма проверка двойной комбинации 24](#_Toc9793175)

[3.6.8. Разработка схемы алгоритма проверка комбинации Каре 24](#_Toc9793176)

[3.6.9. Разработка схемы алгоритма проверка комбинации Стрит Флеш 25](#_Toc9793177)

[3.6.10. Разработка схемы алгоритма проверка комбинации Стрит 26](#_Toc9793178)

[3.6.11. Разработка схемы алгоритма проверка комбинации Флеш 26](#_Toc9793179)

[3.6.12. Разработка схемы алгоритма проверка комбинации Высшая карта 27](#_Toc9793180)

[3.6.13. Разработка схемы алгоритма проверка комбинации Фул Хаус 27](#_Toc9793181)

[3.6.14. Разработка схемы алгоритма сброса карт 28](#_Toc9793182)

[3.6.15. Разработка схемы алгоритма удержать 28](#_Toc9793183)

[3.6.16. Разработка схемы алгоритма ставка 29](#_Toc9793184)

[3.6.17. Разработка схемы алгоритма перерисовка 29](#_Toc9793185)

[3.6.18. Разработка схемы алгоритма рисование полей 30](#_Toc9793186)

[3.6.19. Разработка схемы алгоритма проверка победы 31](#_Toc9793187)

[3.6.20. Разработка схемы алгоритма проверка победы 32](#_Toc9793188)

[4. Особенности разработки программы 33](#_Toc9793189)

[5. Инструкция пользователю 34](#_Toc9793191)

[6. Результаты тестирования 47](#_Toc9793191)

[Заключение 58](#_Toc9793191)

[Приложение 59](#_Toc9793277)

## **Введение**

Целью выполнения курсового проекта является закрепление теоретических знаний и отработка практических навыков работы на компьютере.

Для реализации данной цели необходимо решить следующие задачи:

* разработка программы-игры «Покер»;
* разработка базы данных «Записная книжка»;

В настоящее время хранение информации - важный и неотъемлемый процесс программирования. В данной работе хранение информации рассматривается на примере электронной записной книжки. Использование справочников или базы данных для хранения информации очень актуально, ведь компьютер позволяет сохранять, считывать, использовать и изменять информацию с помощью простых действий.

Второй задачей данного курсового проекта является создание игровой программы «Покер». «Покер» - представляет собой карточную игру, цель которой — выиграть ставки, собрав как можно более высокую покерную комбинацию, используя 2, или 5 карт, или вынудив всех соперников прекратить участвовать в игре. Она развивает стратегические навыки игрока, зрительную память, умственные способности, улучшает память и интуицию.

Данная курсовая работа выполнена на языке программирования высокого уровня С++ с использованием среды CodeBlocks. На сегодняшний день язык С++ считается одним из совершенных и сложных языков программирования.

## **Анализ задания и постановка задач на проектирование**

Согласно заданию на курсовую работу проект должен содержать:

* база данных «Записная книжка»;
* программа-игра «Покер»;
* анимация;
* информация об авторе;
* функция выхода;

Для решения этих задач необходимо разработать специальное меню, которое будет запускаться из главной функции после компиляции проекта, оно будет содержать в себе все перечисленные ранее пункты.

База данных «Записная книжка» - представляет собой базу данных, содержащую некоторую информацию о людях и должна выполнять следующие операции:

* добавление – добавляет человека в базу данных;
* вывод – выводит таблицу со всей информацией справочника анимацию;
* удаление - удаляет человека из базы;
* поиск – производит поиск по имени, фамилии, отчеству, адресу, номеру дома, номеру телефона дню рождения, месяцу рождения и году рождения среди элементов справочника и выводит искомую информацию о человеке;

Игровая программа «Покер» - представляет собой карточную игру, цель которой — выиграть ставки, собрав как можно более высокую покерную комбинацию, используя 2, или 5 карт, или вынудив всех соперников прекратить участвовать в игре.

В игре должно быть несколько стадий:

1. Раздача 2 карт каждому игроку на руки.
2. Происходят 1-ые ставки игровых единиц.
3. Открываются первые 3 карты на столе.
4. Происходят 2-ые ставки игровых единиц.
5. Открывается 4 карта на столе.
6. Происходят 3-ьи и последние ставки игровых единиц.
7. Открывается 5 карта на столе.
8. Происходит проверка всех возможных комбинаций.
9. Сравниваются все полученные комбинации.
10. Определяется победитель по лучшей комбинации.
11. Происходит повтор предыдущих пунктов 1-10, пока ваша сумма не станет меньше 0, вы проиграете, и игра будет окончена, или вы наберете больше 4000 и выиграете, и игра будет окончена.
12. Возвращается игровая валюта в исходную и происходит повторение всех предыдущих пунктов.

После закрытия окна или нажав кнопку «Esc» игра будет завершена и произойдет возращение в главное меню.

Анимация – рисование пиксельного кота, его движений, который летит по звёздному небу, оставляя за собой след радуги.

Информация об авторе - содержит информацию о разработчике данного курсового проекта.

## **Разработка алгоритмов**

## 3.1. Разработка схемы алгоритма функции «main»

В главной функции «main» будет изменяться размер окна и вызываться подпрограмма меню. Схема алгоритма главной функции представлена на рисунке 1.

main.cpp



Рисунок 1 – Общая схема алгоритма главной функции

## 3.2. Разработка схемы алгоритма меню

Для перехода между задачами требуется меню. Оно нам необходимо чтобы любой пользователь мог выбрать одну из пяти задач из пункта меню. Изначально будет выделена одна из подзадач другим цветом, слева и справа стрелочками «=> подзадача<=». В зависимости от нажатой стрелочки («вверх» или «вниз») будет изменяться, выделения подзадач. При нажатии кнопки «Enter» будет запущена одна из подзадач, которая была выделена перед нажатием. В меню будет 5 подзадач:

* База Данных;
* Покер;
* Анимация;
* Об авторе;
* Выход;

Меню будет реализовано с помощью отдельной подпрограммы, схема алгоритма которой показана на рисунке 2.

main.cpp



Рисунок 2 - Схема алгоритма меню

## 3.3. Разработка схемы алгоритма база данных «Записная книжка»

В подпрограмме база данных, будет предложение пользователю ввести файл, с которого будет считываться информация. Потом из введенного файла считается информация и запишется в 1-ую структуру. Файл закроется и вся информация из 1-ой структуры перезапишется во 2-ую. После будет вызвано меню базы данных. При возвращении будет вызываться подпрограмма сохранения, в которой вся измененная информация будет записана. Схема алгоритма подпрограммы, которой представлена на рисунке 3.

Baza\_Dannih.h

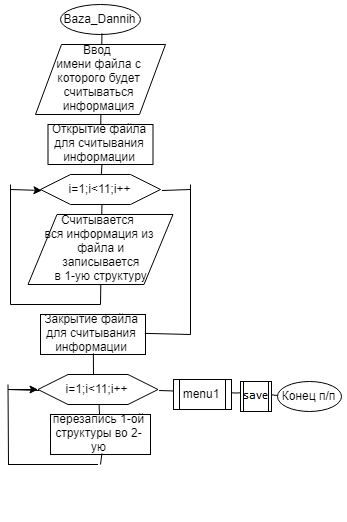


Рисунок 3 - Схема алгоритма базы данных

## 3.3.1. Разработка схемы алгоритма меню базы данных

Подпрограмма меню базы данных с возможностью выбора других функций, (работает так же, как и основное меню) содержащее:

* удаление людей из базы данных;
* добавление в неё;
* поиск по одному из девяти предложенных пунктов;
* возвращение в прошлое меню;

Схема алгоритма подпрограммы, которой представлена на рисунке 4. Baza\_Dannih.h

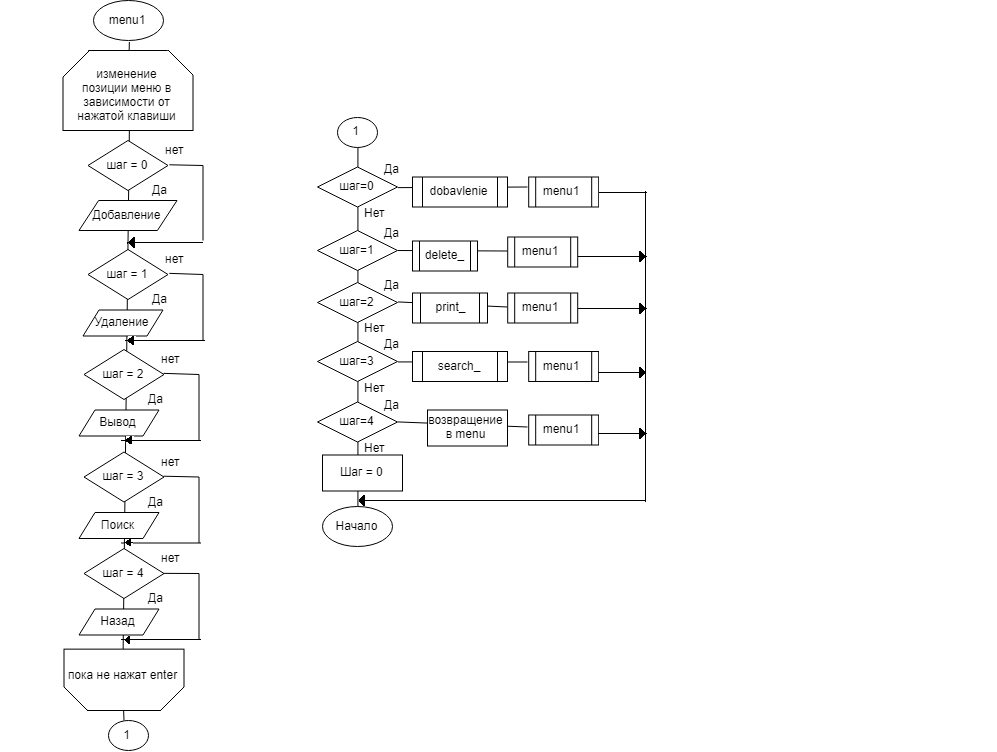


Рисунок 4-Схема алгоритма меню базы данных

## 3.3.2. Разработка схемы алгоритма удаления из базы данных

В подпрограмме удаления изначально будет выводиться таблица всех людей, которые есть в базе данных. И по номеру можно будет удалить человека из неё, схема алгоритма подпрограммы, которой представлена на рисунке 5.

Baza\_Dannih.h

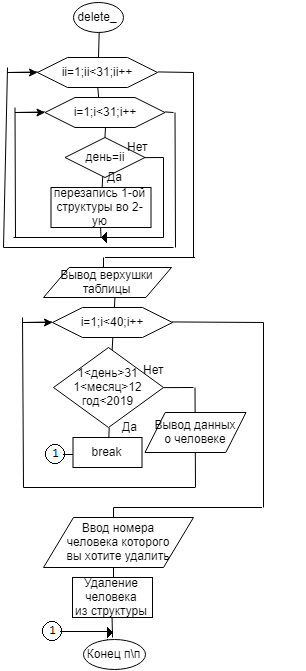


Рисунок 5 – Схема алгоритма удаления человека из базы данных

## 3.3.3. Разработка схемы алгоритма вывода всей информации

Для вывода всей информации о людях, которые есть в базе данных, используется таблица, в которую запишутся все люди из 2-ой структуры и выведется на экран. Схема алгоритма подпрограммы представлена на рисунке 6

Baza\_Dannih.h

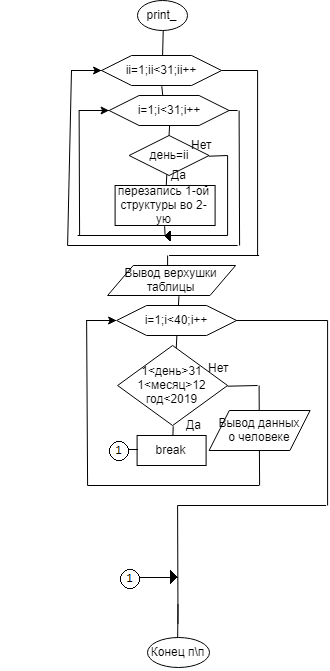


Рисунок 6 – Схема алгоритма подпрограммы, выводящей всю информацию справочника на экран

## 3.3.4. Разработка схемы алгоритма для поиска человека

Подпрограмма базы данных поиска будет искать людей по одному из критериев:

* Имени;
* Фамилии;
* Отчеству;
* Адресу;
* Номеру дома;
* Номеру телефона;
* Дню рождения;
* Месяцу рождения;
* Году рождения;

Среди элементов справочника и выводить искомую информацию о человеке, схема алгоритма подпрограммы представлена на рисунке 7.

Baza\_Dannih.h

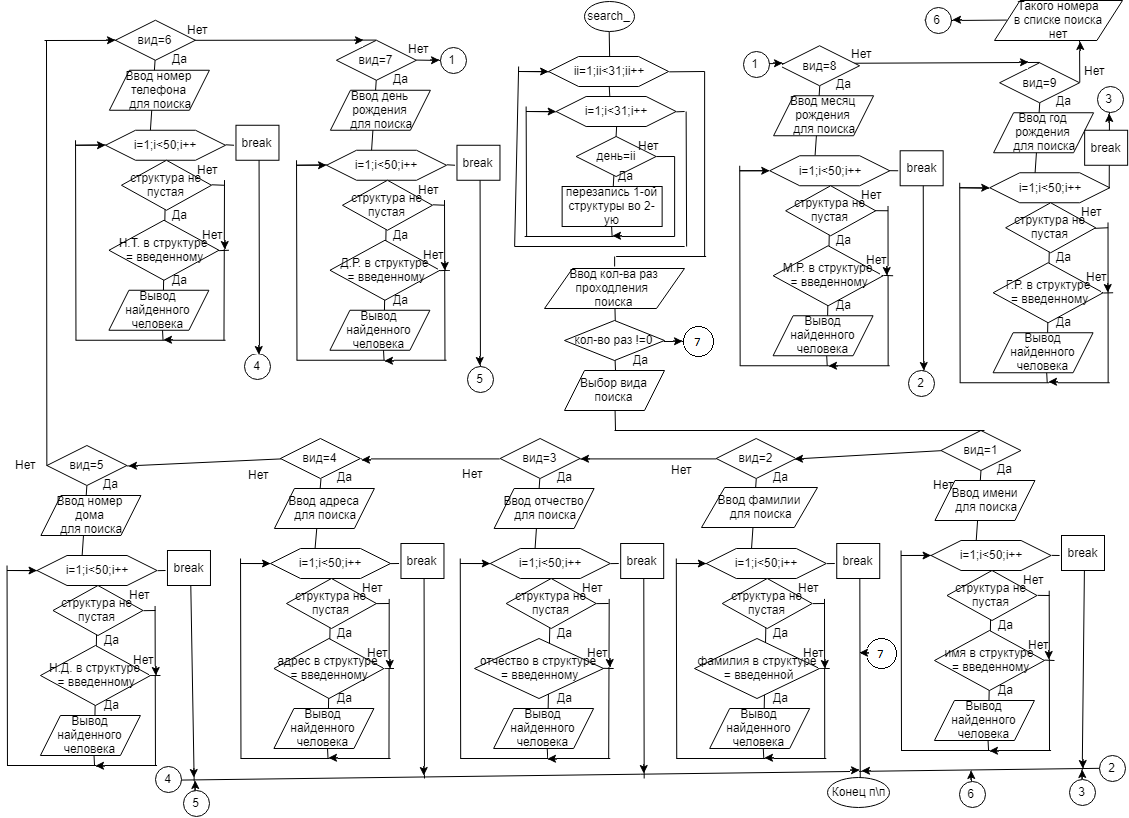


Рисунок. 7 – Схема алгоритма подпрограммы поиск

## 3.3.5. Разработка схемы алгоритма добавления в базу данных

Подпрограмма добавления человека в базу данных, заполнив пункты информации о человеке:

* фамилия;
* имя;
* отчество;
* улицу;
* номер дома;
* номер телефона;
* день рождения;
* месяц рождения;
* год рождения;

Будет записана во 2-ую структуру. Схема алгоритма подпрограммы представлена на рисунке 8.

Baza\_Dannih.h

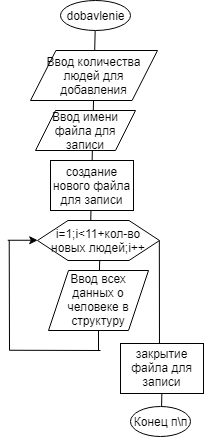


Рисунок 8 – Схема алгоритма подпрограммы добавления

## 3.3.6. Разработка схемы алгоритма сохранения информации

Функция сохранения всей информации, которую мы изменяли в течение всей работы с базой данной, будет записана в некоторый файл, который пользователь назовет. Схема алгоритма подпрограммы, которой представлена на рисунке 9.

Baza\_Dannih.h

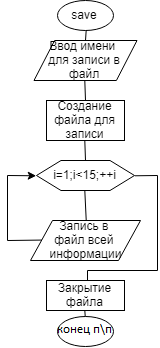


Рисунок 9 – Схема алгоритма подпрограммы сохранения

## 3.4. Разработка схемы алгоритма об авторе

При выборе подпрограммы «Об авторе» в пунктах главного меню, на экран выведется информация о нем. Схема алгоритма подпрограммы об авторе представлена на рисунке 10.

main.cpp

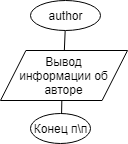


Рисунок 10 – Схема алгоритма вывода информации об авторе

## 3.5. Разработка схемы алгоритма анимация

Подпрограмма анимация представляет собой рисование пиксельного кота, его движений, который летит по звёздному небу, оставляя за собой след радуги. Схема алгоритма подпрограммы показана на рисунке 11.

main.cpp

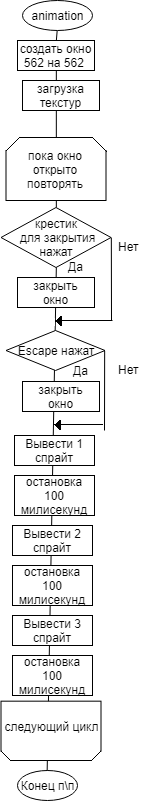


Рисунок 11 – Схема алгоритма подпрограммы анимация

## 3.6. Разработка схемы алгоритма главной функции покера

Подпрограмма главная функция покера включает в себя инициализацию и её проверку, вызов подпрограммы событий с окном, подпрограммы событий игры, подпрограмму перерисовки игрового процесса и подпрограмму выхода. Схема алгоритма подпрограммы представлена на рисунке 12.

CApp.cpp



Рисунок 12 – Схема алгоритма подпрограммы главной функции для покера

## 3.6.1. Разработка схемы алгоритма событий окна

Подпрограмма всех возможных событий, которые можно отправить в создаваемое окно для покера. Схема алгоритма подпрограммы, которой представлена на рисунке 13.

CEvent.cpp

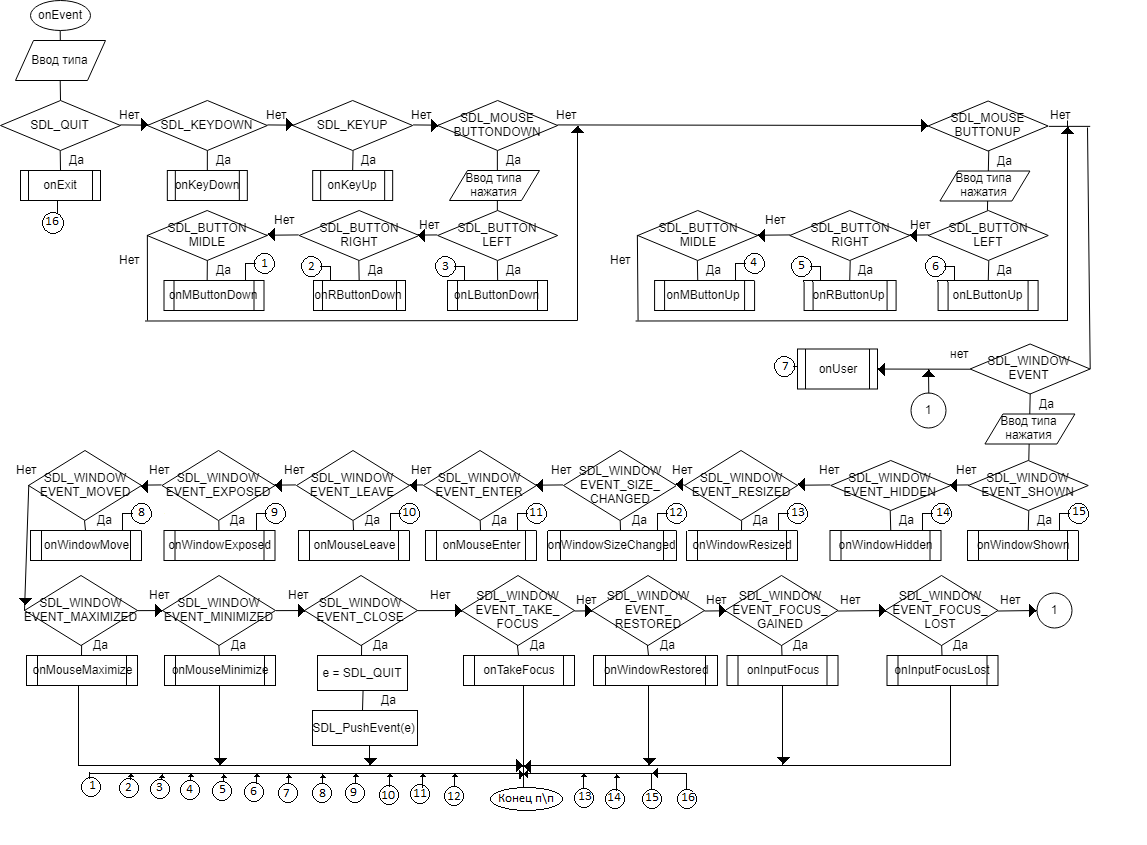


Рисунок 13 – Схема алгоритма событий для покера

## 3.6.2. Разработка схемы алгоритма события нажатия левой кнопкой мыши по окну

Схема алгоритма подпрограммы события нажатия левой кнопкой мыши по создаваемому окну для покера, для считывания какой-либо кнопки, если в такую попадает курсор мыши, представлена на рисунке 14.

CApp\_OnEvent.cpp

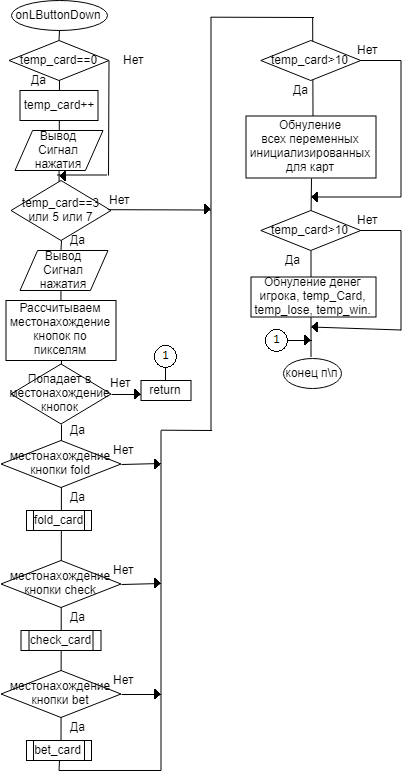


Рисунок 14 – схема алгоритма подпрограммы события нажатия левой

## 3.6.3. Разработка схемы алгоритма события нажатие крестика

Подпрограмма события – нажатие крестика для закрытия у создаваемого окна для покера, которая возвращает нулевое значение для главной переменной функции, чтобы началось обнуление всех переменных для выхода из главной функции покера. Схема алгоритма подпрограммы, которой представлена на рисунке 15.

CApp\_OnEvent.cpp



Рисунок 15 – схема алгоритма подпрограммы события – нажатие крестика

## 3.6.4. Разработка схемы алгоритма событий во время игры

Подпрограммы происходящих событий во время игры «Покер», включает в себя постепенную раздачу случайных карт всем игрокам. Схема алгоритма подпрограммы, которой представлена на рисунке 16.

CApp\_onLoop.cpp

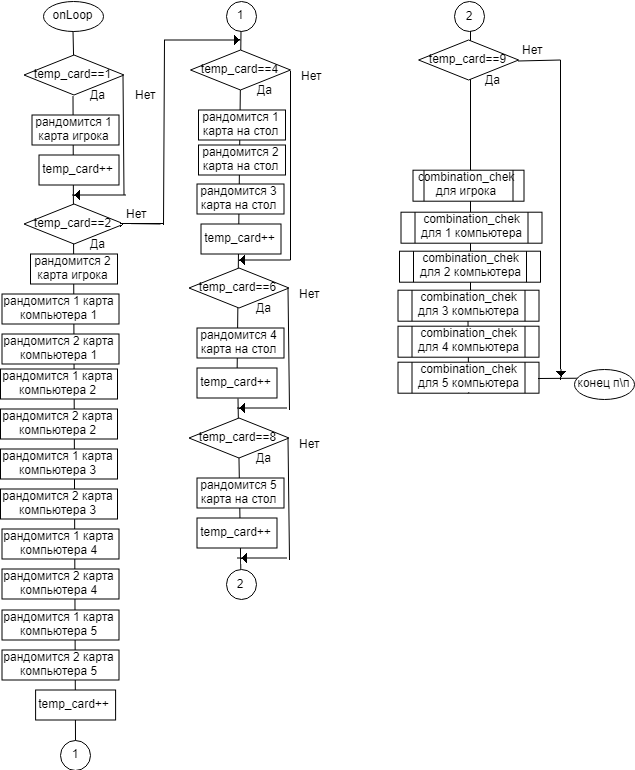


Рисунок 16 – схема алгоритма подпрограммы происходящих событий

## 3.6.5. Разработка схемы алгоритма событий во время игры

Подпрограммы проверка всех комбинаций, включает в себя проверку всех возможных комбинаций, которые могут быть в игре «Покер» у игроков. Схема алгоритмов подпрограммы, которой представлена на рисунке 17.

CApp.cpp



Рисунок 17 – схема алгоритма подпрограммы проверка комбинаций

## 3.6.6. Разработка схемы алгоритма проверка тройной комбинации

Подпрограмма проверка тройной комбинации игроков во время игры. Схема алгоритма подпрограммы, которой представлена на рисунке 18.

CApp.cpp

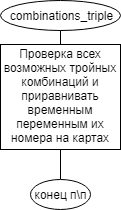


Рисунок 18 – схема алгоритма подпрограммы проверка тройной комбинации

## 3.6.7. Разработка схемы алгоритма проверка двойной комбинации

Схема алгоритма подпрограммы проверка двойной комбинации у игроков во время игры «Покер» представлена на рисунке 19.

CApp.cpp

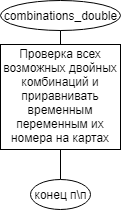


Рисунок 19 – схема алгоритма подпрограммы проверка двойной комбинации

## 3.6.8. Разработка схемы алгоритма проверка комбинации Каре

Схема алгоритма подпрограммы проверка комбинации Каре у игроков во время игры «Покер» представлена на рисунке 20.

CApp.cpp

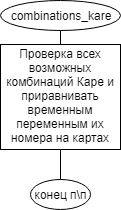


Рисунок 20 – схема алгоритма подпрограммы проверка комбинации Каре

## 3.6.9. Разработка схемы алгоритма проверка комбинации Стрит Флеш

Схема алгоритма подпрограммы проверка комбинации Стрит Флеш у игроков во время игры «Покер» представлена на рисунке 21.

CApp.cpp



Рисунок 21 – схема алгоритма подпрограммы проверка комбинации Стрит Флеш

## 3.6.10. Разработка схемы алгоритма проверка комбинации Стрит

Подпрограмма проверка комбинации Стрит у игроков во время игры. Схема алгоритма подпрограммы, которой представлена на рисунке 22.

CApp.cpp

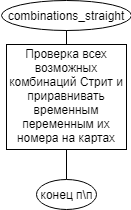


Рисунок 22 – схема алгоритма подпрограммы проверка комбинации Стрит

## 3.6.11. Разработка схемы алгоритма проверка комбинации Флеш

Подпрограмма проверка комбинации Флеш у игроков во время игры Схема алгоритма подпрограммы, которой представлена на рисунке 23.

CApp.cpp

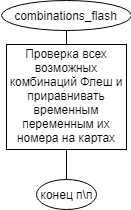


Рисунок 23 – схема алгоритма подпрограммы проверка комбинации Флеш

## 3.6.12. Разработка схемы алгоритма проверка комбинации Высшая карта

Подпрограмма проверка комбинации Высшая карта у игроков во время игры. Схема алгоритма подпрограммы, которой представлена на рисунке 24.

CApp.cpp

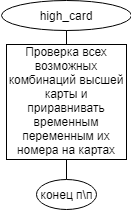


Рисунок 24 – схема алгоритма подпрограммы проверка комбинации Высшая карта

## 3.6.13. Разработка схемы алгоритма проверка комбинации Фул Хаус

Подпрограммы проверка комбинации Фул Хаус у игроков во время игры. Схема алгоритма подпрограммы, которой представлена на рисунке 25.

CApp.cpp

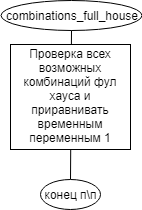


Рисунок 25 – схема алгоритма подпрограммы проверка комбинации

Фул Хаус

## 3.6.14. Разработка схемы алгоритма сброса карт

Подпрограмма сброс – сброса карт у игроков во время игры, по нажатию на кнопку «Flod» в правой нижней части экрана. Схема алгоритма подпрограммы, которой представлена на рисунке 26.

CApp.cpp



Рисунок 26 – схема алгоритма подпрограммы сброс

## 3.6.15. Разработка схемы алгоритма удержать

Подпрограмма удержать – удержать ставку, поставив 100 игровых единиц во время игры, по нажатию кнопки «Check» в правой нижней части окна. Схема алгоритма подпрограммы, которой представлена на рисунке 27.

CApp.cpp



Рисунок 27 – схема алгоритма подпрограммы удержать

## 3.6.16. Разработка схемы алгоритма ставка

Подпрограмма ставка – поднять ставку, поставив 200 игровых единиц во время игры, по нажатию кнопки «Bet» в правой нижней части окна. Схема алгоритма подпрограммы, которой представлена на рисунке 28.

CApp.cpp



Рисунок 28 – схема алгоритма подпрограммы ставка

## 3.6.17. Разработка схемы алгоритма перерисовка

Подпрограмма перерисовка – отрисовка всех изображений выведенных на окно «Покер», в течение всей игры. Схема алгоритма подпрограммы, которой представлена на рисунке 29.

Capp\_onRender.cpp



Рисунок 29 – схема алгоритма подпрограммы перерисовка

## 3.6.18. Разработка схемы алгоритма рисование полей

Подпрограмма рисование полей – отрисовка всех изображений выведенных на окно «Покер», в течение всей игры. Схема алгоритма подпрограммы, которой представлена на рисунке 30.

Capp\_onRender.cpp

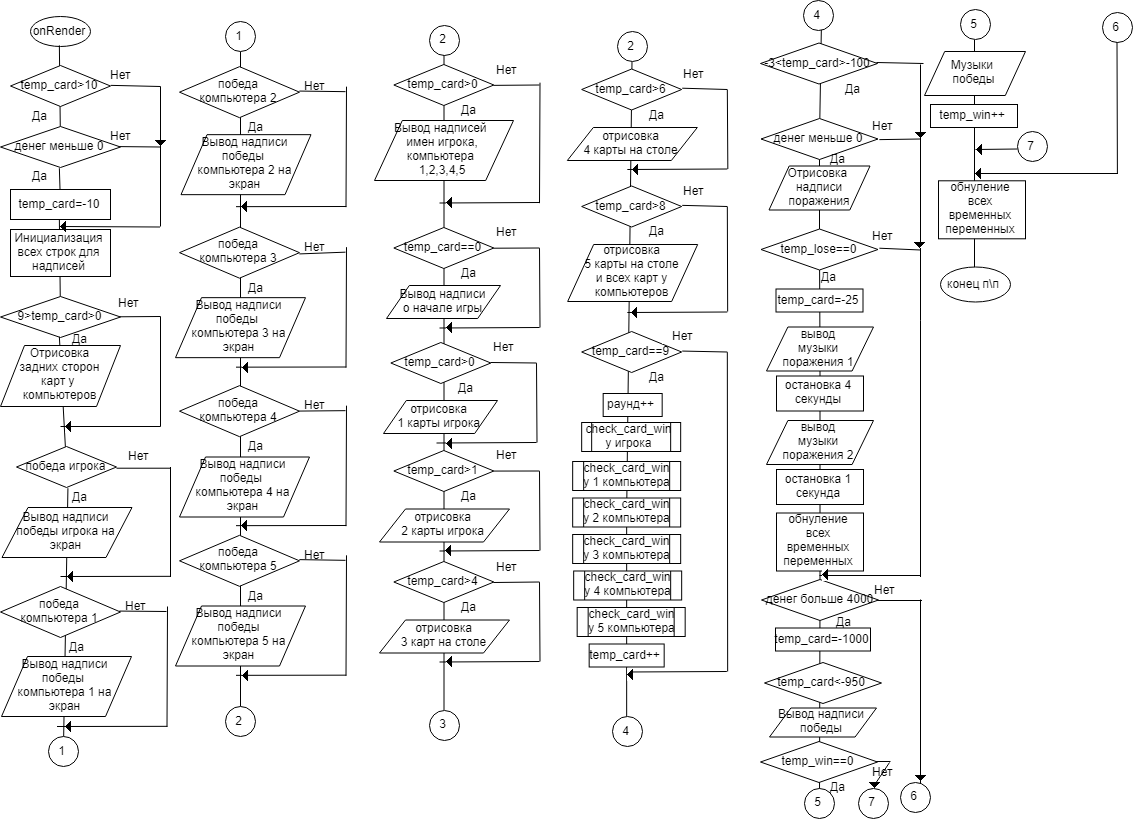


Рисунок 30 – схема алгоритма подпрограммы рисование полей

## 3.6.19. Разработка схемы алгоритма проверка победы

Подпрограмма проверка победы – сравнивание всех комбинаций игроков друг с другом, и по правилам покера, произойдет выбор победителя. Схема алгоритма подпрограммы, которой представлена на рисунке 31.

CApp.cpp



Рисунок 31 – схема алгоритма подпрограммы проверка победы

## 3.6.20. Разработка схемы алгоритма проверка победы

Подпрограмма выхода – обнуление всех загруженных текстур, музыки, форматов текста и тд. Схема алгоритма подпрограммы, которой представлена на рисунке 32.

Capp\_onQuit.cpp



Рисунок 32 – схема алгоритма подпрограммы выхода

## **Особенности разработки программы**

Для реализации всех алгоритмов был использован язык C++, среда разработки CodeBlocks и библиотека SDL\_2. Программа разработана в строгом соответствии со схемами алгоритмов. В программе был использован выбор пунктов в меню с клавиатуры, и запись всей информации базы данных в файлы. Вся информация программы находится в 14-ти разных файлах. Программа состоит из 8 276‬ строк кода.

## **Инструкция пользователю**

Для запуска программы необходимо запустить файл kursachproject.exe.

В открывшемся окне выбрать стрелками на клавиатуре один из пунктов меню, чтобы выбрать нужный пункт нажмите Enter.

Для работы с базой данных «Записная книжка» необходимо выбрать соответствующий пункт меню и нажать Enter. Как представлено на рисунке 33.

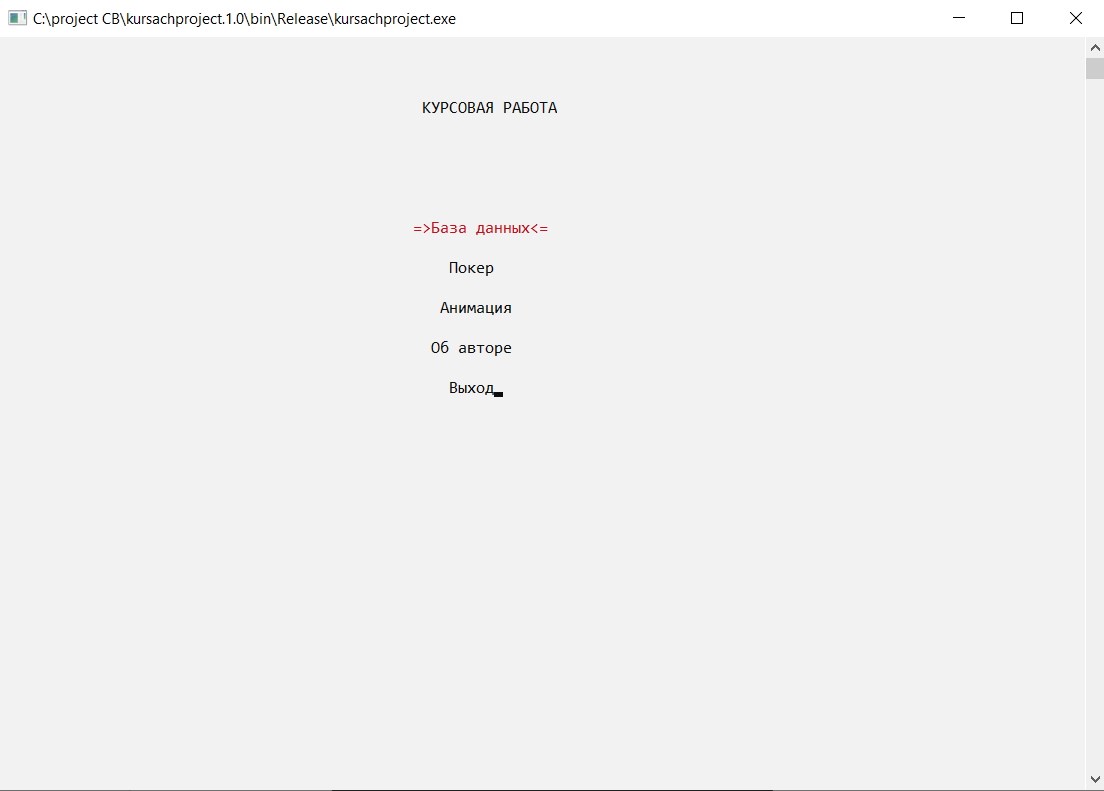


Рисунок 33 – основное меню

Далее необходимо ввести имя файла, с которого будет считываться информация о людях, как показано на рисунке 34.

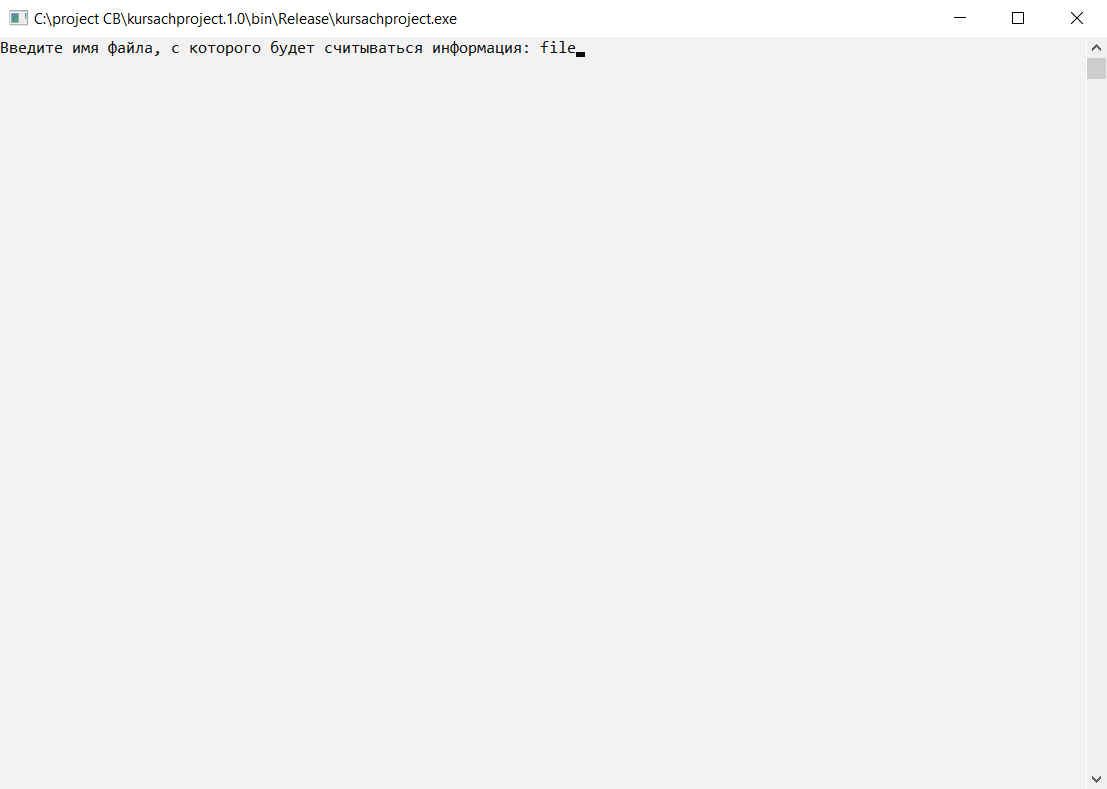


Рисунок 34 – ввод названия файла

В подменю базы данных нужно навестись стрелочками на нужный вам пункт и нажать Enter. Как на рисунке 35.

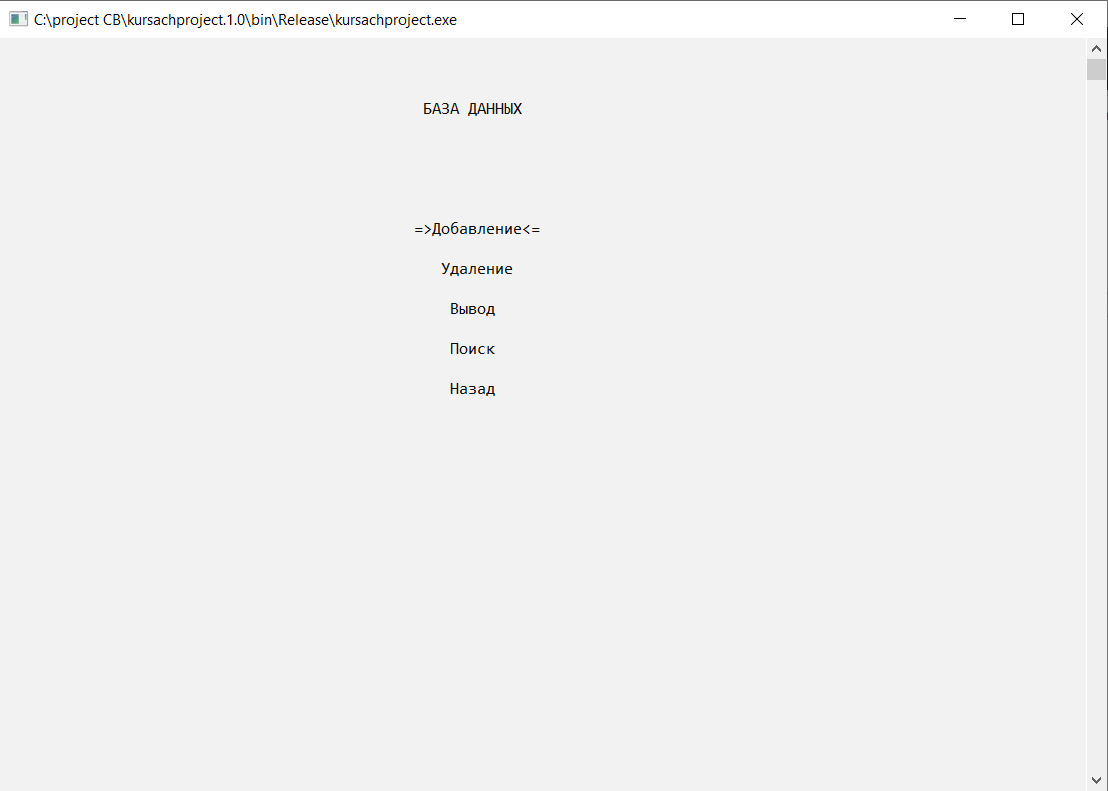


Рисунок 35 – подменю базы данных

В подменю добавление по базе данных необходимо ввести количество людей, которых хотите внести в базу данных, ввести имя файла в который они будут записаны, и написать подробную информацию о них. Как представлено на рисунке 36.

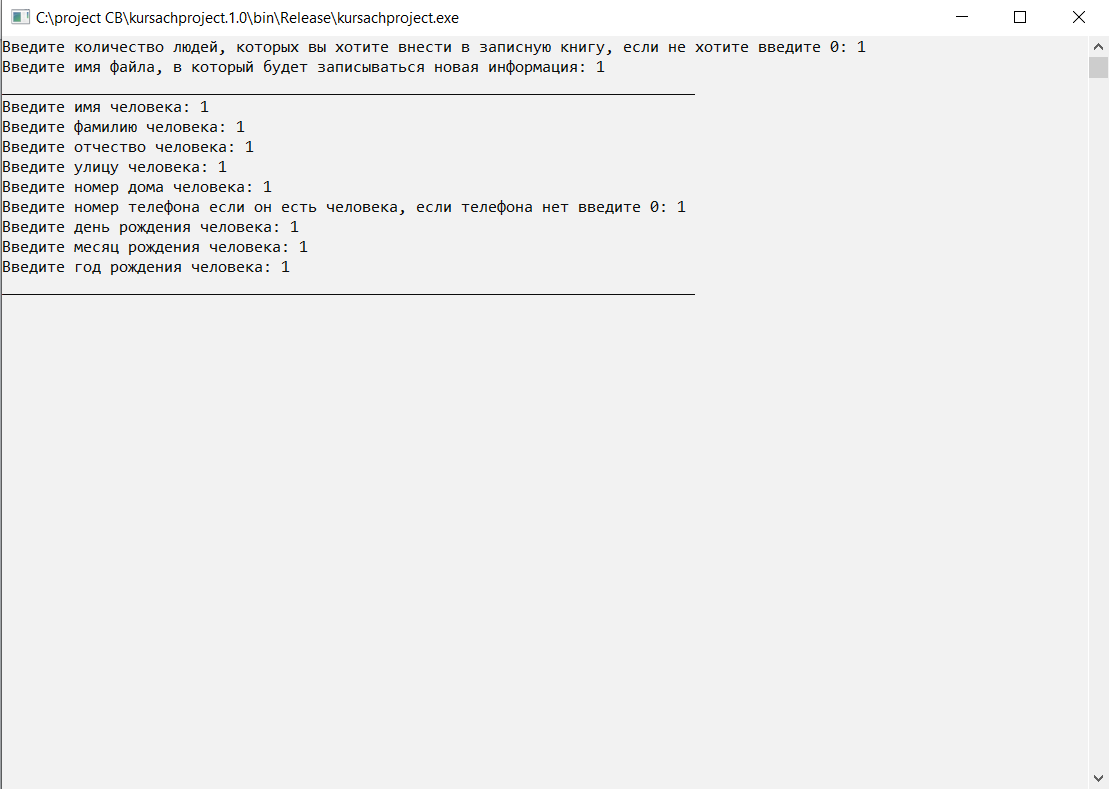


Рисунок 36 – пример работы добавления в базу данных

В подменю удаления по базе данных необходимо вписать номер человека из выведенной таблицы. Как представлено на рисунке 37.

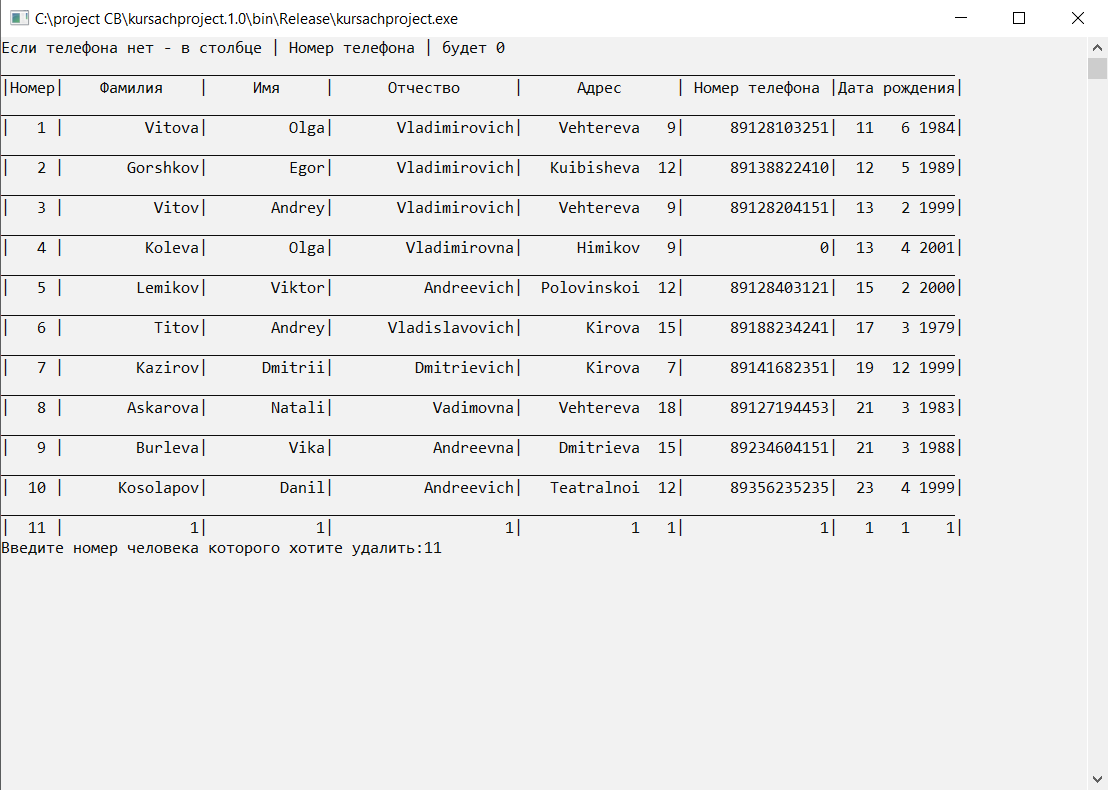


Рисунок 37 – пример работы удаления из базы данных

В подменю вывода всей информации о людях, будут выведены все, кто есть в базе данных. Как представлено на рисунке 38.

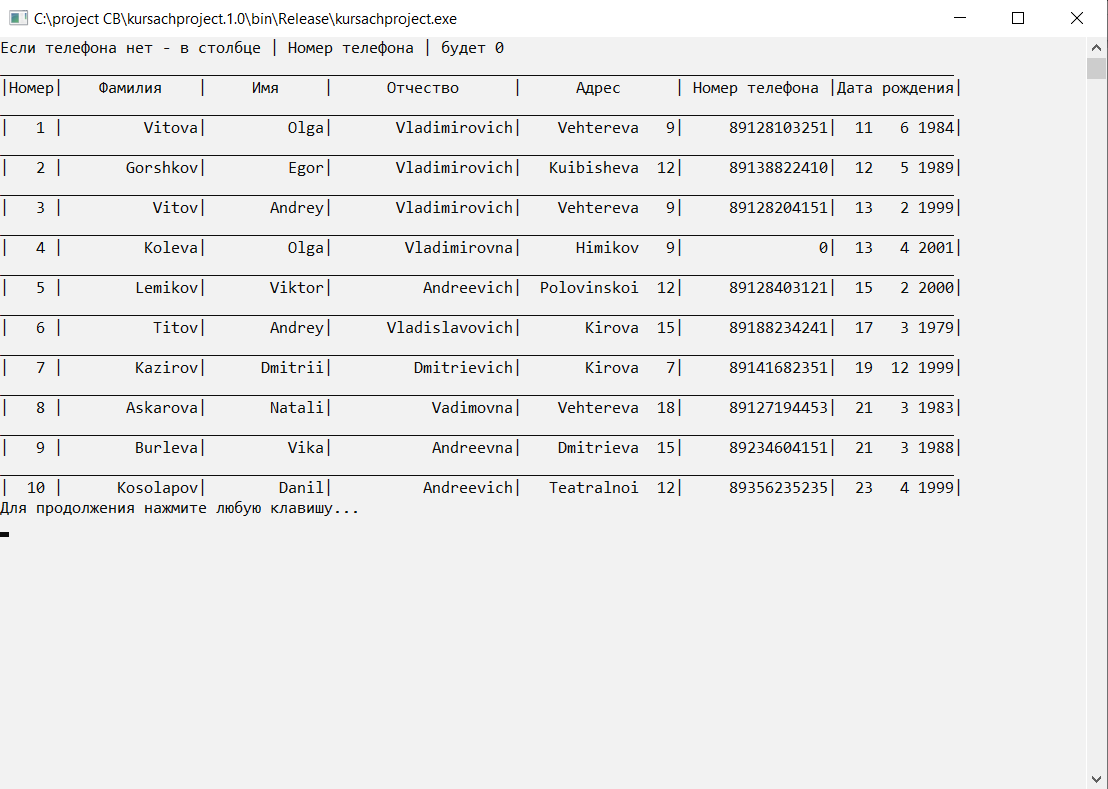


Рисунок 38 – пример работы вывода базы данных

В подменю поиска по базе данных необходимо ввести количество раз прохождения поиска и нажать Enter, выбрать ключевое слово для поиска, ввести его номер также и нажать Enter. Как представлено на рисунке 39.

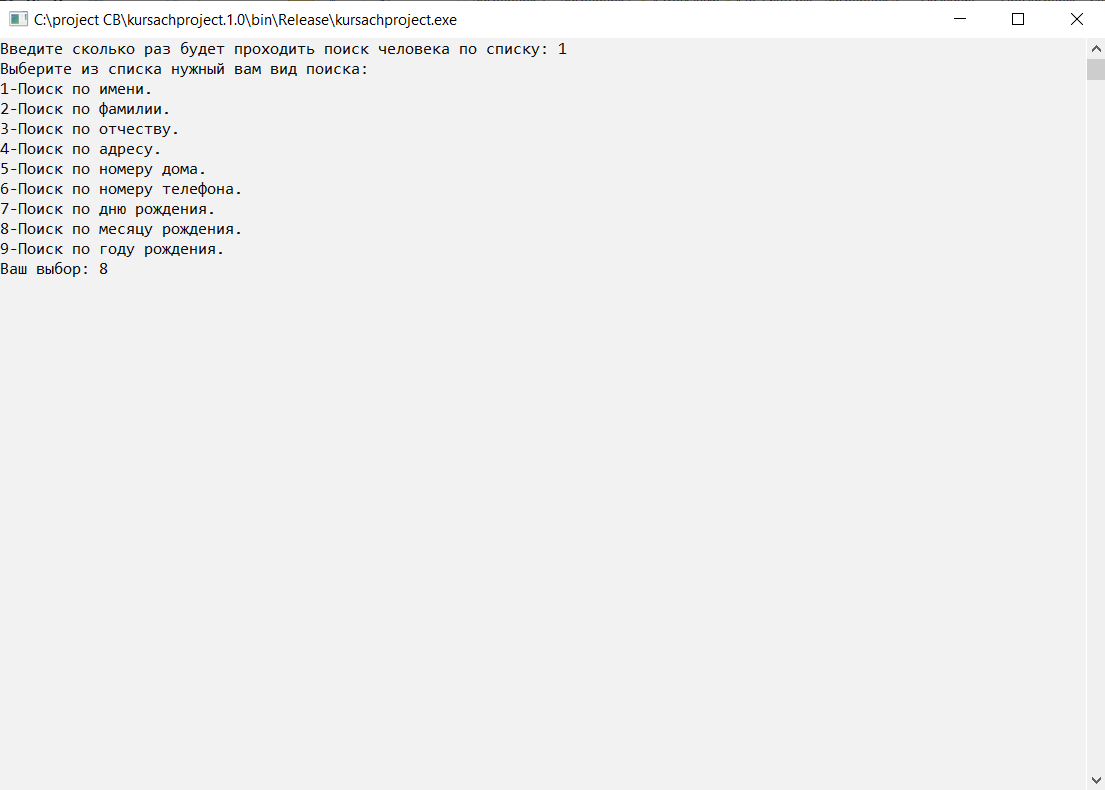


Рисунок 39 – пример работы поиска по базе данных

После этого нужно ввести поисковый запрос и нажать Enter, как представлено на рисунке 40.

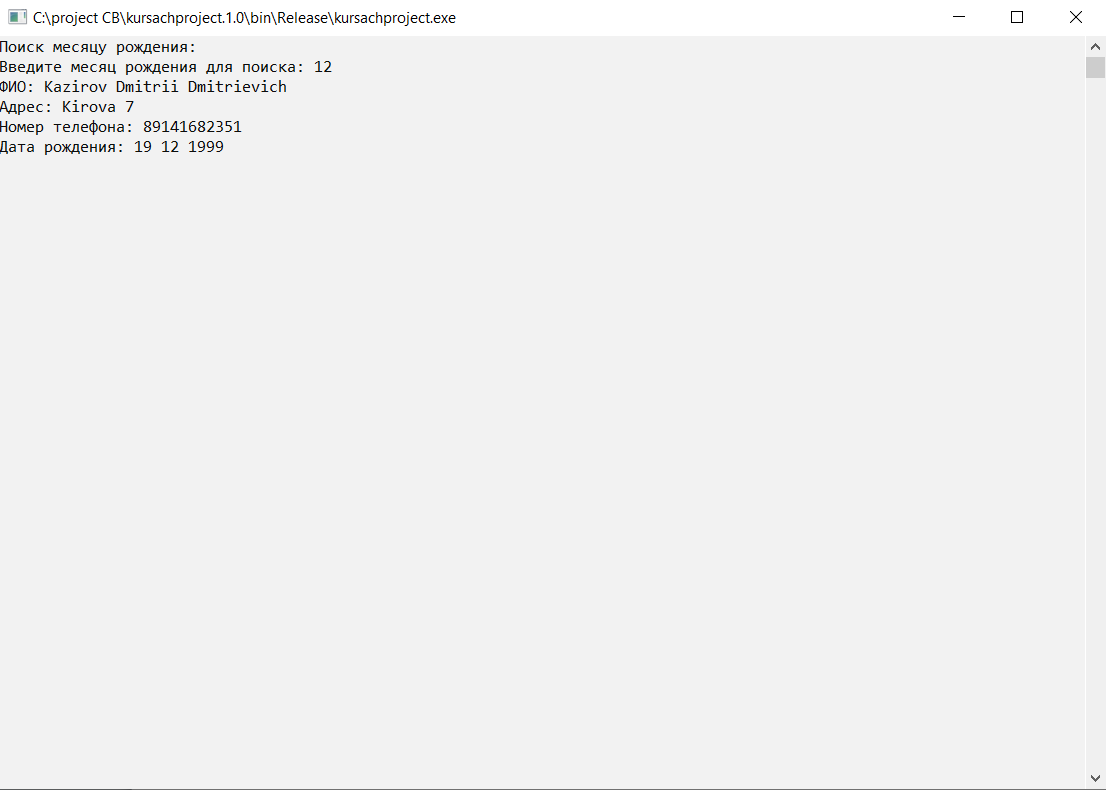


Рисунок 40 – пример работы поиска по базе данных

После окончания работы с базой данных наведитесь в подменю клавишу «Назад» и нажмите Enter, потом введите имя файла для сохранения и также нажмите Enter. Как представлено на рисунке 41.

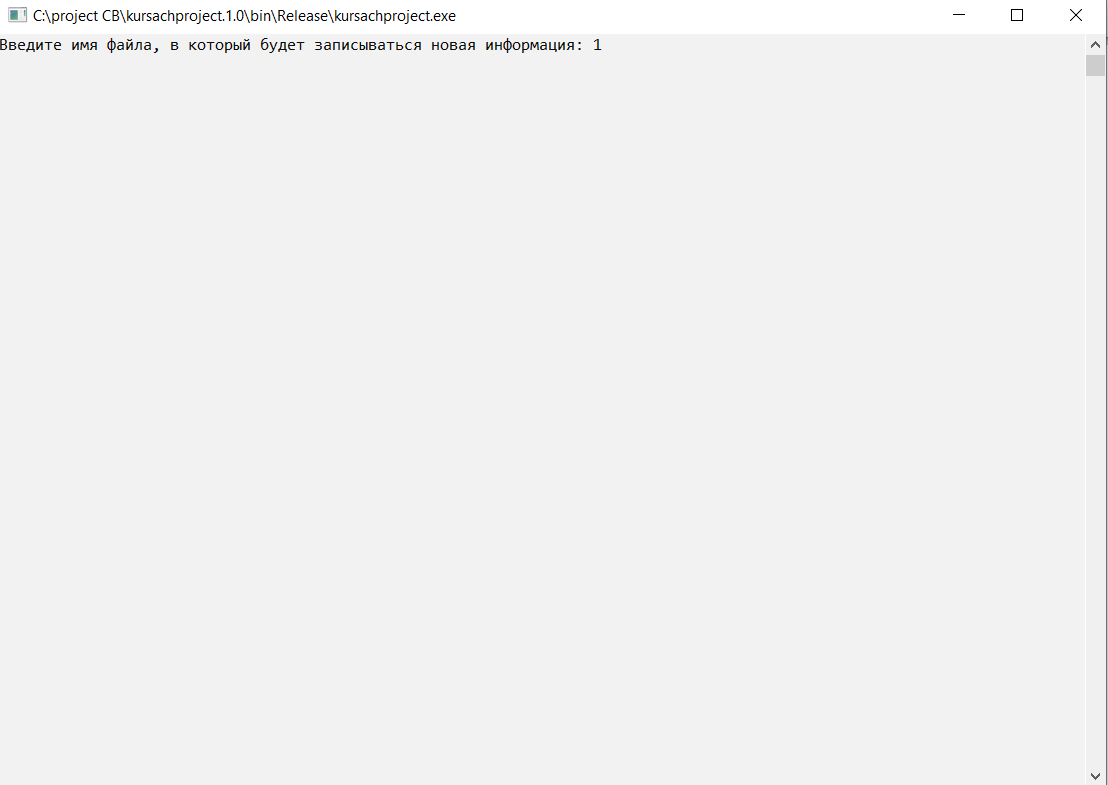


Рисунок 41 – пример работы сохранения базы данных

Чтобы открыть игру, выберите в основном меню подзадачу «Покер». После этого выведется отдельное окно с игрой. Управление осуществляется нажатиями левой кнопки мыши. Чтобы выйти нажмите Esc или нажмите на крестик для закрытия окна. Как это представлено на рисунке 42.



Рисунок 42 – пример работы игры «Покер»

Нажмите левой кнопкой мыши по окну, для начала игры, как это представлено на рисунке 43.



Рисунок 43 – пример работы игры «Покер»

Всем игрокам раздали карты. У нас есть выбор сбросить, удержать ставку (100) или поднять ставку (200). На примере удержали ставку и на столе открыли 3 карты. Как представлено на рисунке 44.



Рисунок 44 – пример работы игры «Покер»

Далее на примере удерживают ставку еще 2 раза, и открываются последние 2 карты, и определяется победитель по лучшей комбинации. Как представлено на рисунке 45.



Рисунок 45 – пример работы игры «Покер»

Победителем на примере стал 4 компьютер. У него сложилась двойная пара из 2 вольтов и 2 дам. Данные обо всех комбинациях игроков выводятся в консоль, как представлено на рисунке 46.

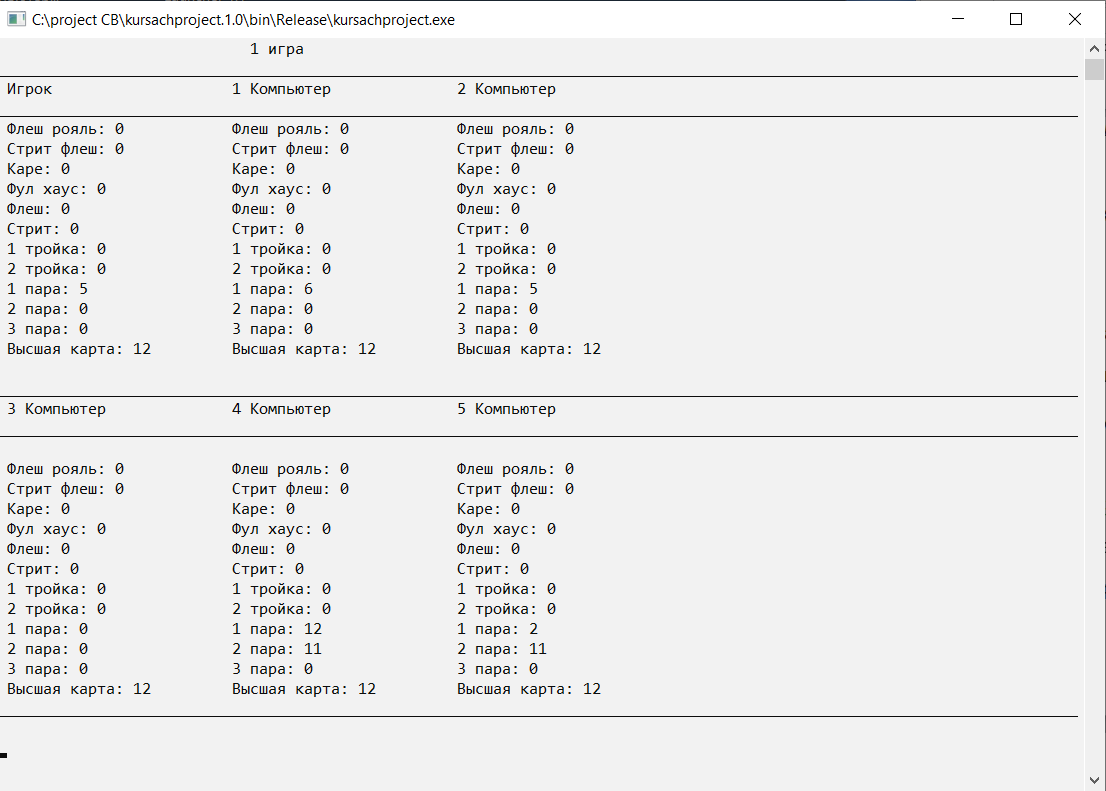


Рисунок 46 – пример работы вывода всех комбинаций в консоль игры

Выбрав пункт «Анимация» из основного меню, пользователь увидит анимацию движения пиксельного кота, который летит по звездному небу и оставляет за собой след радуги. Для закрытия анимации, нажмите «Esc» или закройте окно с анимацей. Пример работы анимации представлен на рисунке 47.

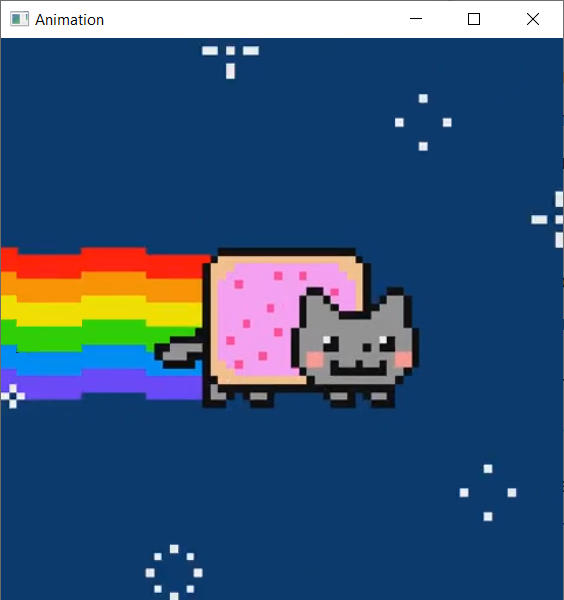


Рисунок 47 – пример работы анимации

Выбрав пункт «Об авторе» из основного меню, пользователь увидит информацию о разработчике данного курсового проекта, как представлено на рисунке 48.

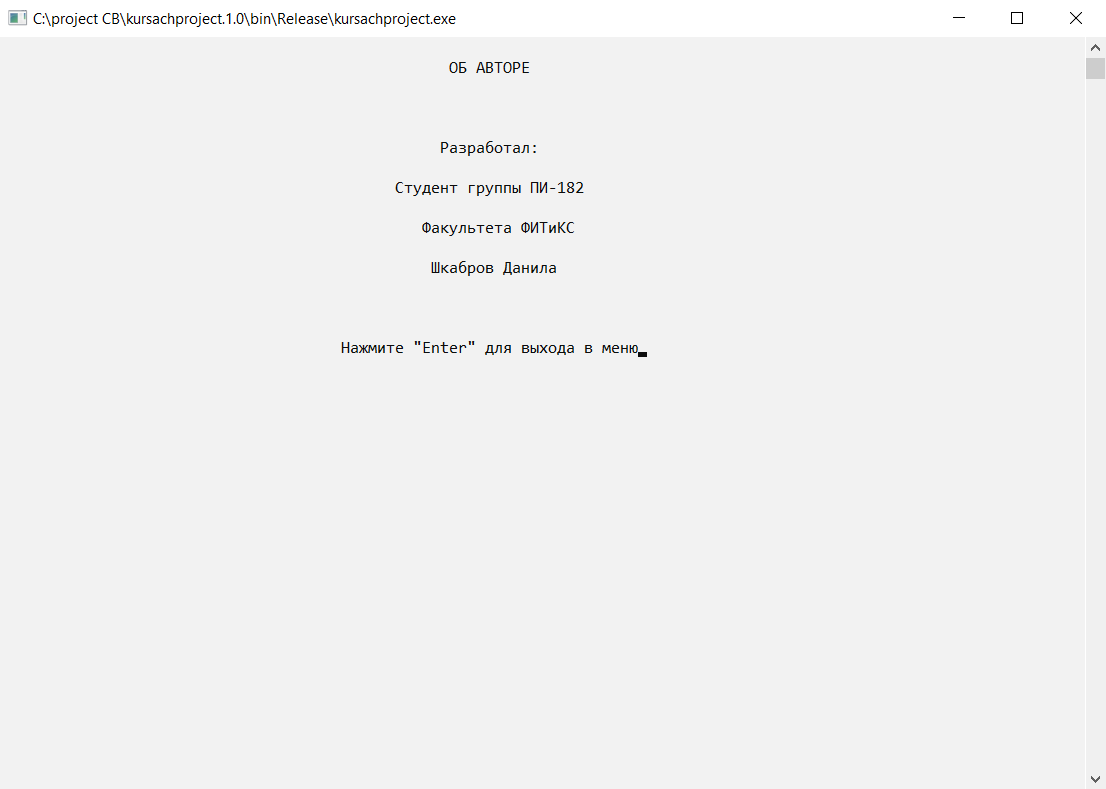


Рисунок 48 – информация об авторе

Последний пункт – это «Выход», при нажатии на который происходит выход из программы.

## **Результаты тестирования**

Программа была протестирована во всех режимах и результаты тестирования были сведены в таблице 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Режим | Вводимые  данные | Ожидаемый  результат | Результат |
| Ввод данных в БД | 10 людей их его данные | 10 людей их его данные | 10 людей их его данные |
| Поиск данных в БД | Человек 1 | Человек 1 найден | Человек 1 найден |
| Добавление данных в БД | Человек 11 и его данные | Человек 11 и его данные | Человек 11 и его данные |
| Удаление данных из БД | 1 человек | 1 человек удален | 1 человек удален |

Таблица 1 – Результаты тестирования программы

Результат тестирования основного меню представлен на рисунке 49.

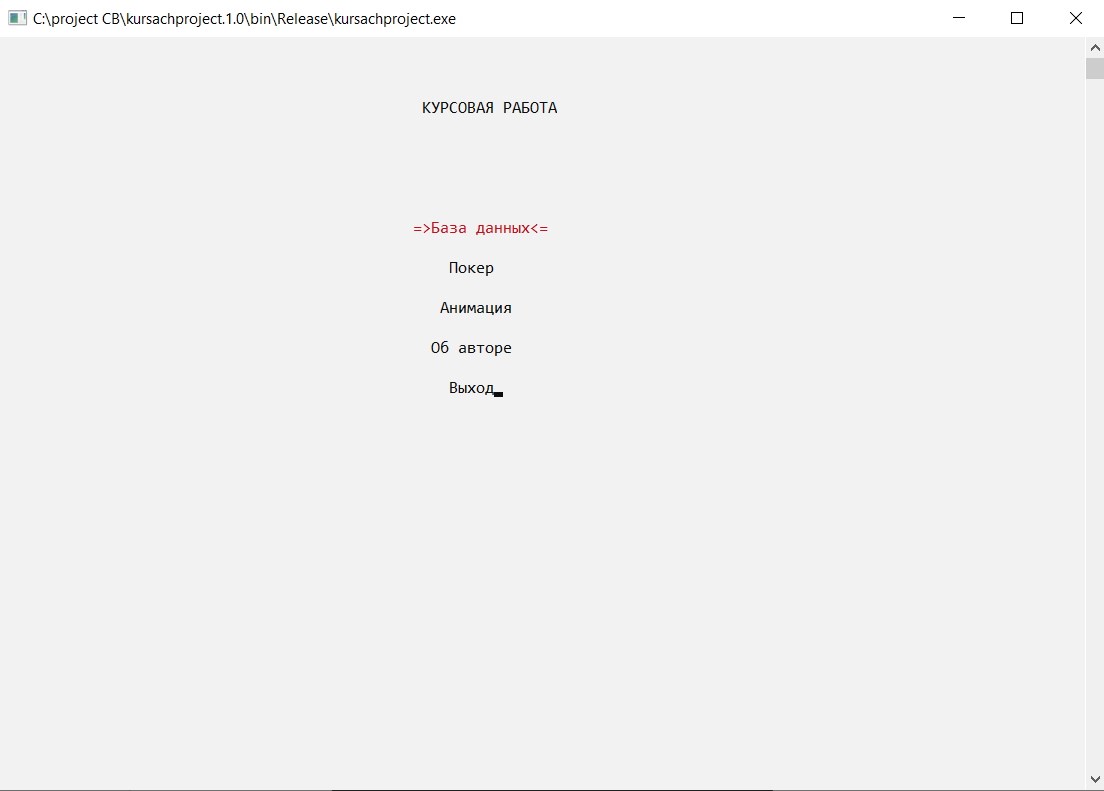


Рисунок 49 – тестирование основного меню

Результат тестирования ввода имени файла, с которого будет считываться информация, представлен на рисунке 50.

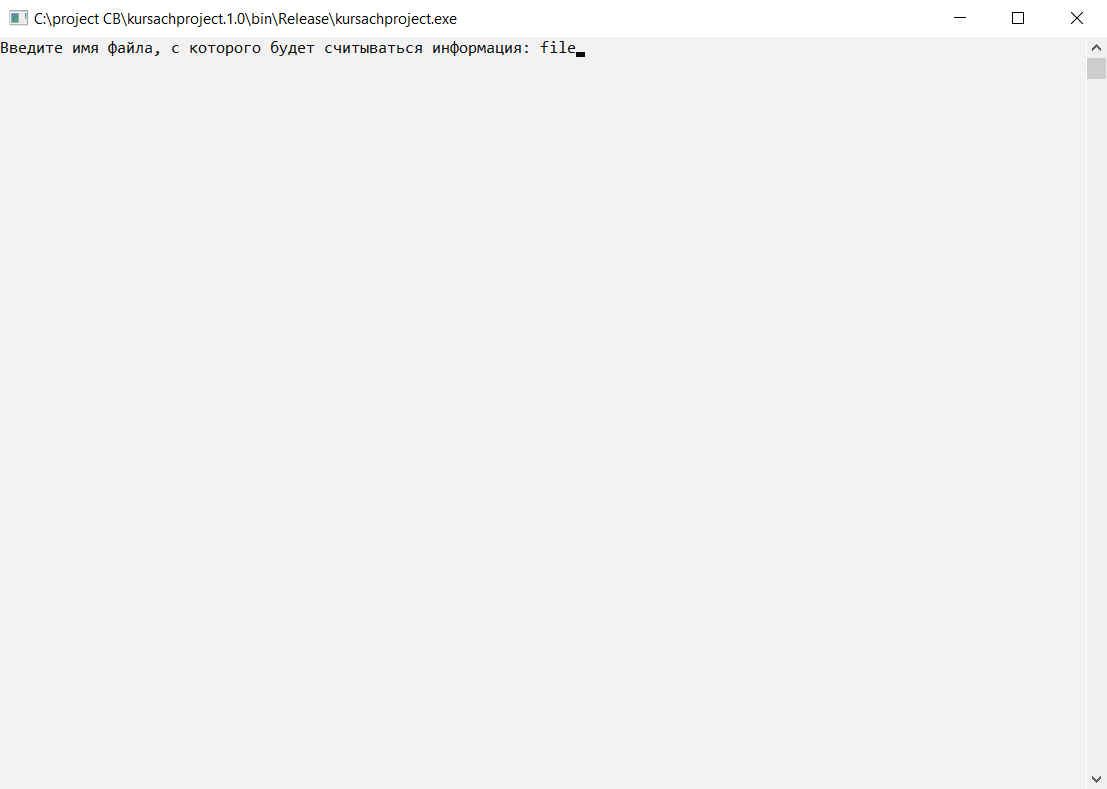


Рисунок 50 – тестирование ввода имени файла

Результат тестирования подменю, который представлен на рисунке 51.

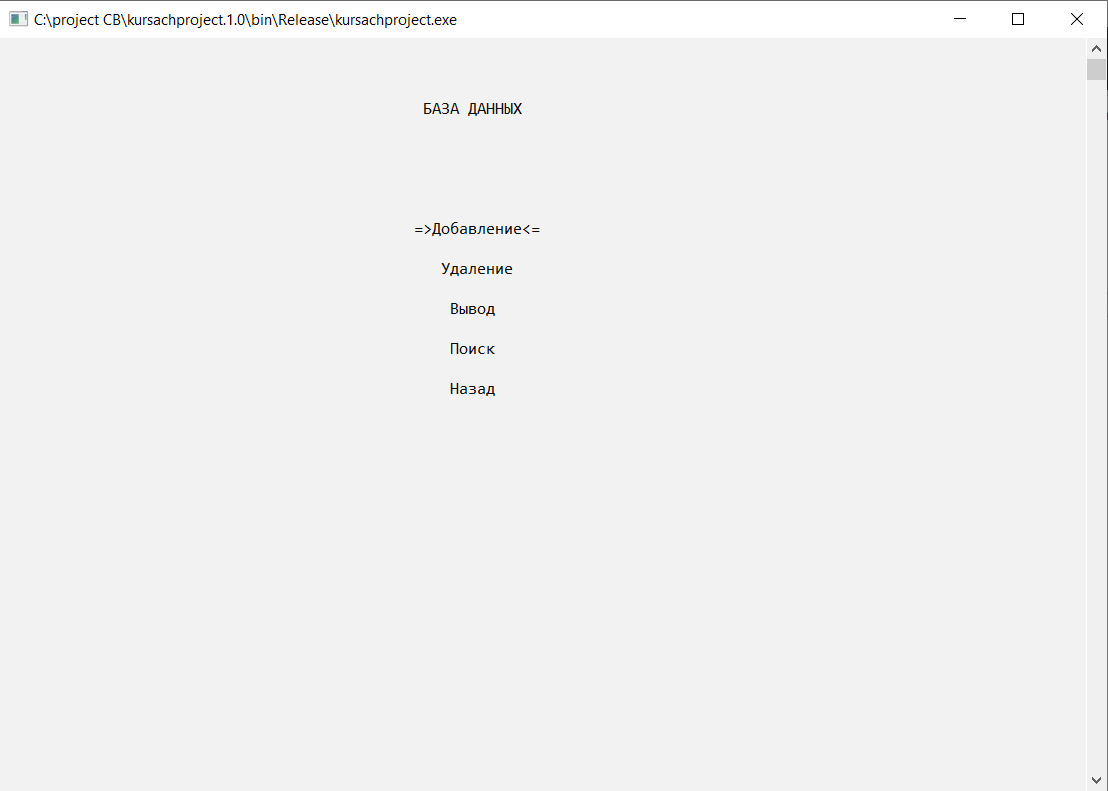


Рисунок 51 – тестирование подменю базы данных

Результат тестирования добавления человека в базу данных, представлен на рисунке 52.

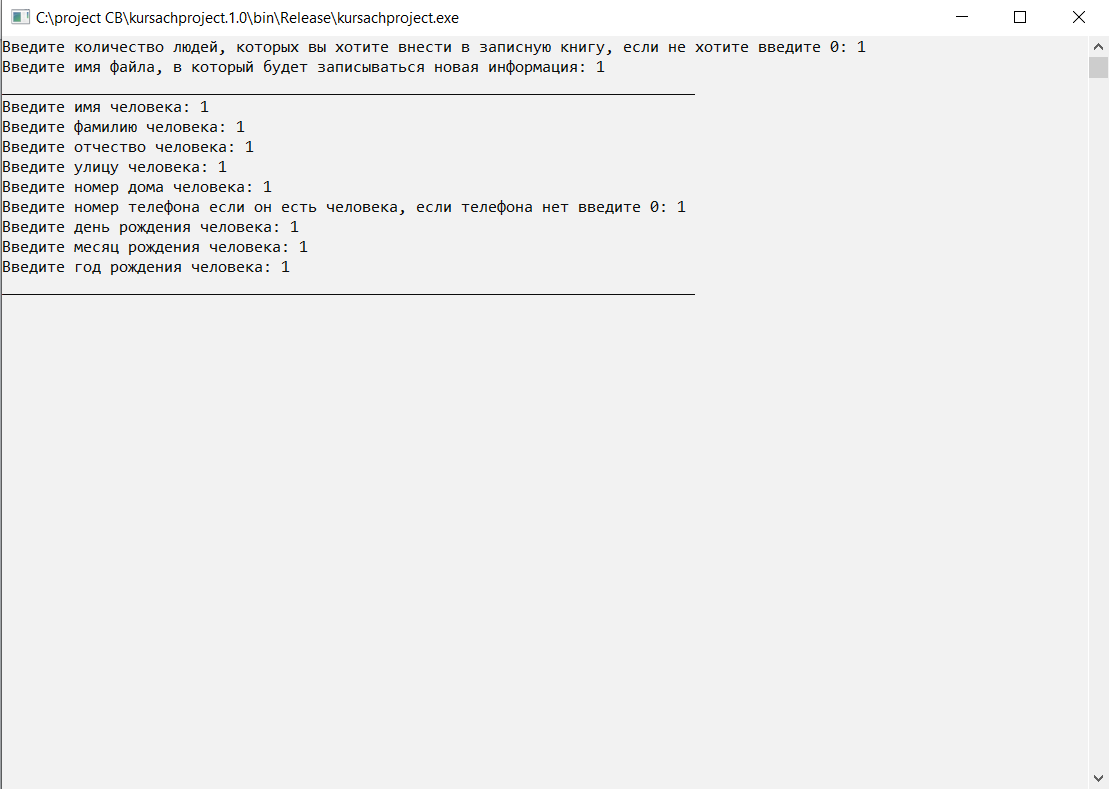


Рисунок 52 – тестирование работы добавления в базу данных

Результат тестирования удаления человека из базы данных, представлен на рисунке 53.

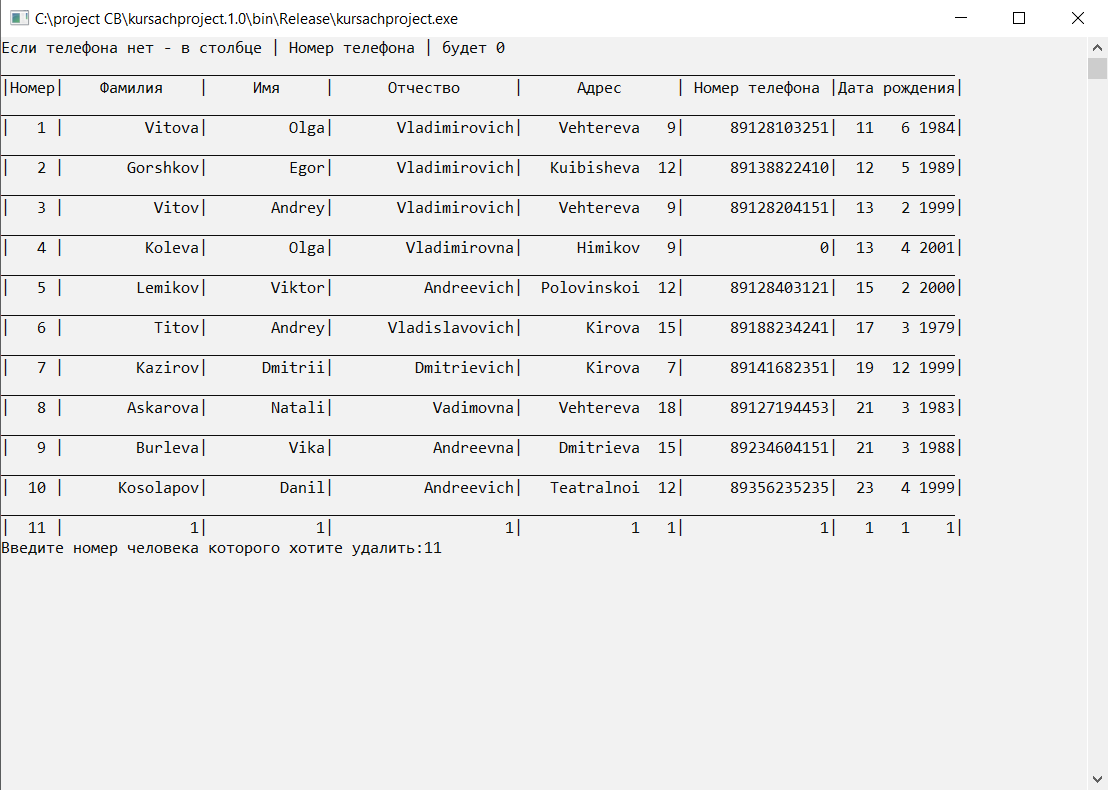


Рисунок 53 – тестирование работы удаления из базы данных

Результат тестирования вывода всей информации о людях, представлен на рисунке 54.

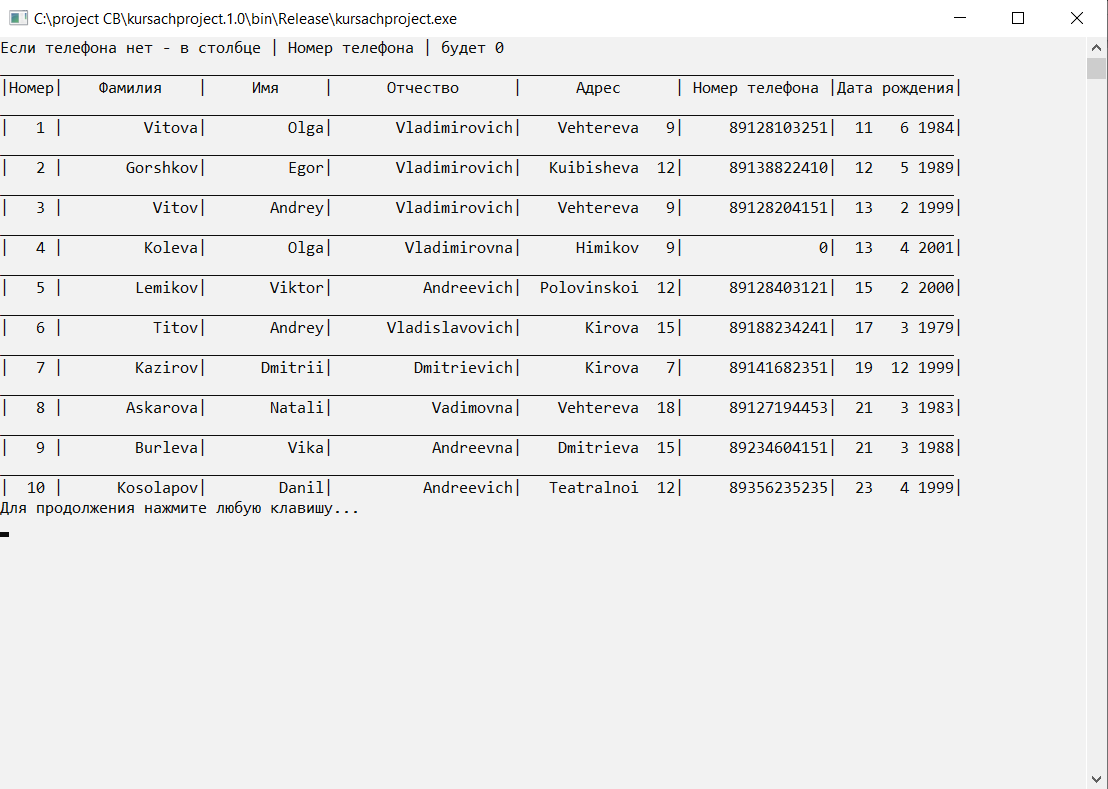


Рисунок 54 – тестирование работы вывода базы данных

Результат тестирования поиска информации о людях, представлен на рисунке 55 и рисунке 56.

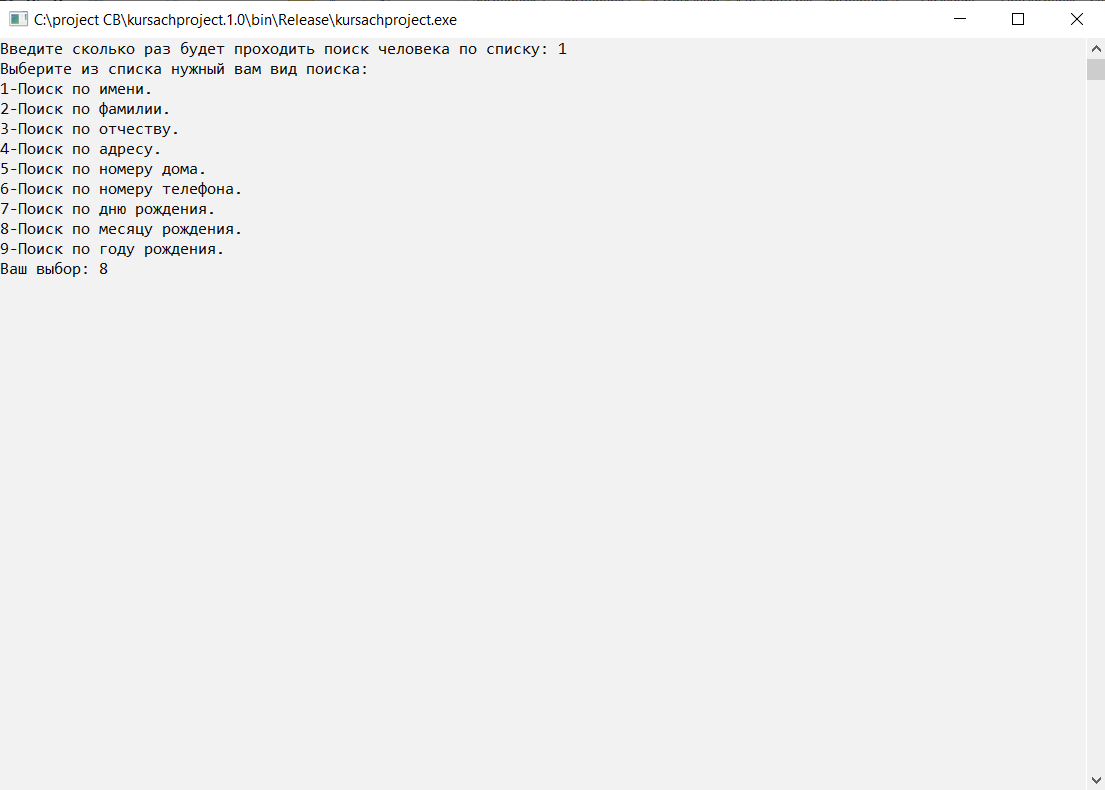


Рисунок 55 – тестирование работы поиска по базе данных

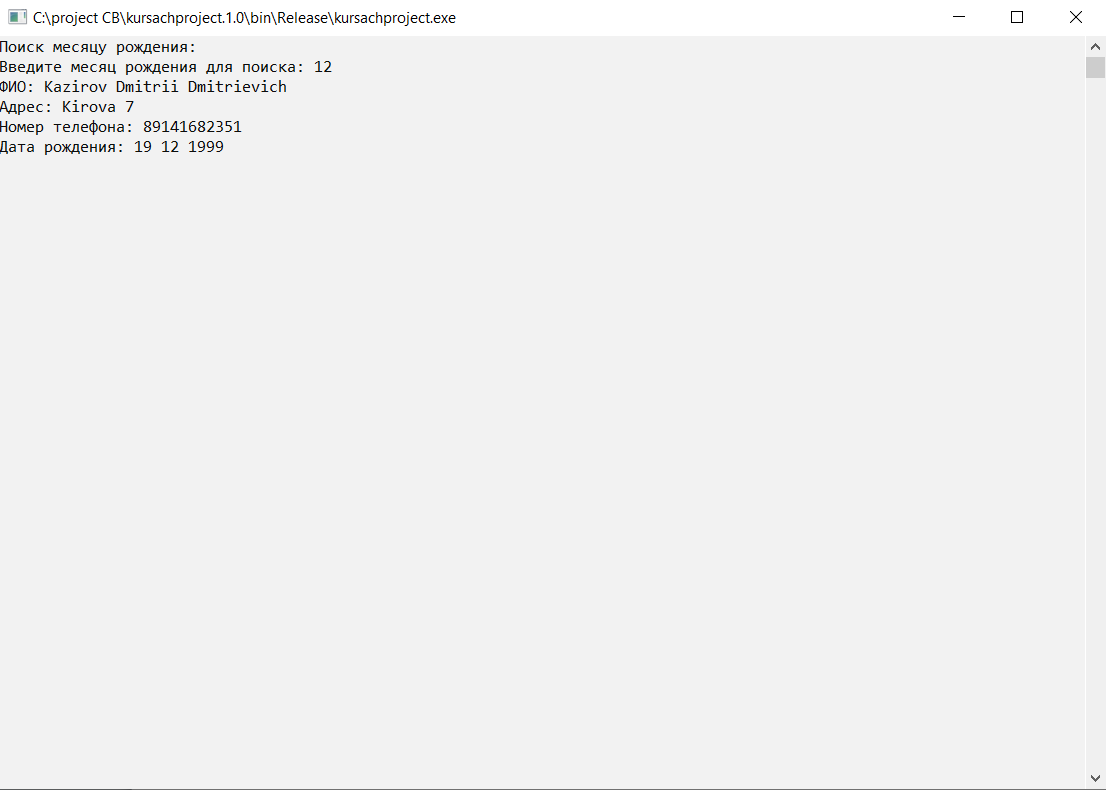


Рисунок 56 – тестирование работы поиска по базе данных

Результат тестирования сохранения в файл всей информации из базы данных, представлен на рисунке 57.

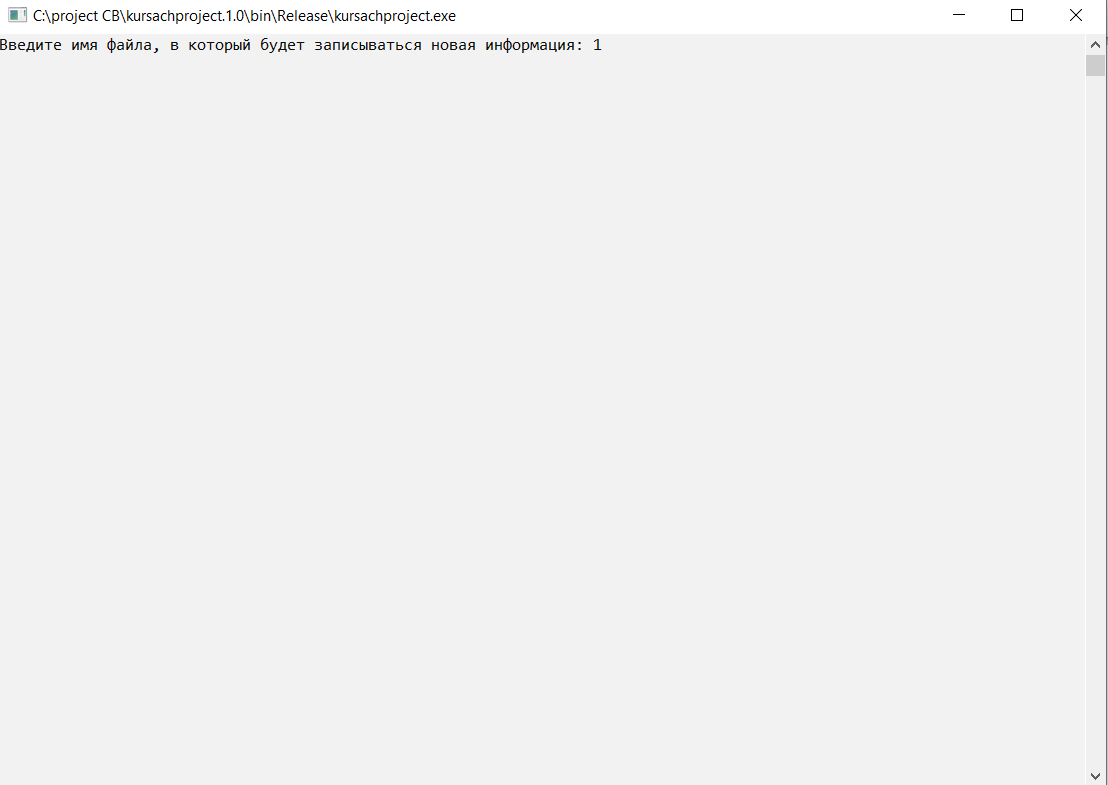


Рисунок 57 – тестирование работы сохранения базы данных

Результат тестирования игры «Покер», представлен с рисунка 58 по рисунок 61.



Рисунок 58 – тестирование работы игры «Покер»



Рисунок 59 – тестирование работы игры «Покер»



Рисунок 60 – тестирование работы игры «Покер»



Рисунок 61 – тестирование работы игры «Покер»

Результат тестирования вывода всех комбинаций у игроков, представлен на рисунке 62.

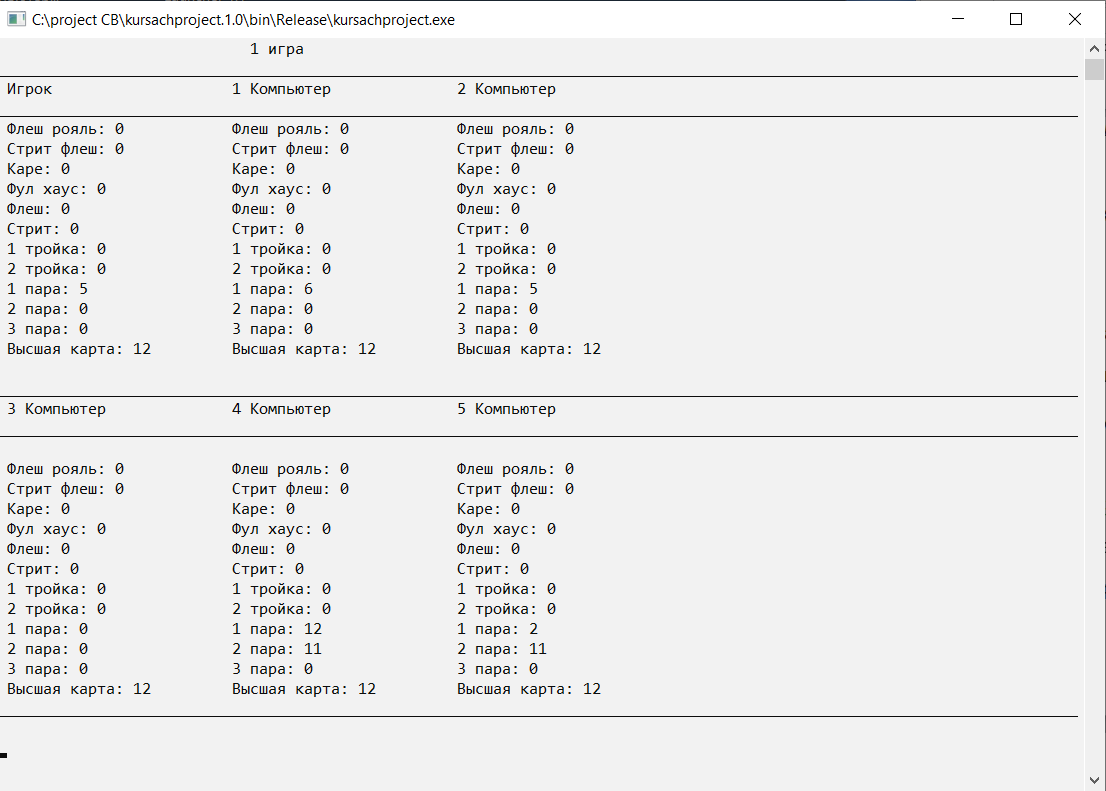


Рисунок 62 – тестирование работы вывода всех комбинаций в консоль игры

Результат тестирования анимации, представлен на рисунке 63.

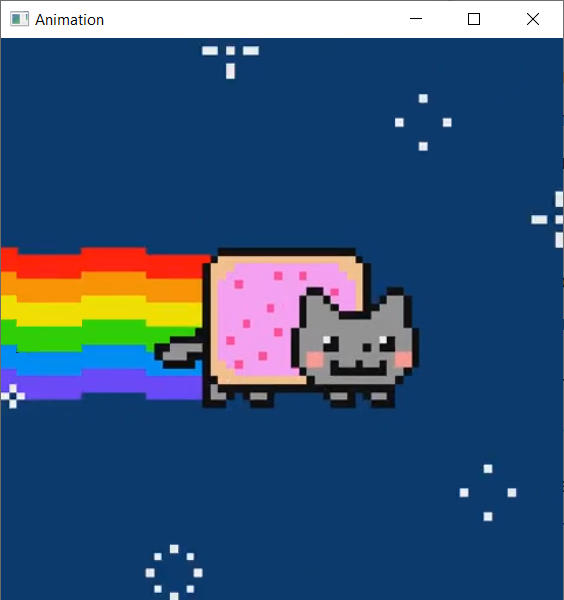


Рисунок 63 – тестирование работы анимации

Результат тестирования подзадачи «Об авторе», представлен на рисунке 64.

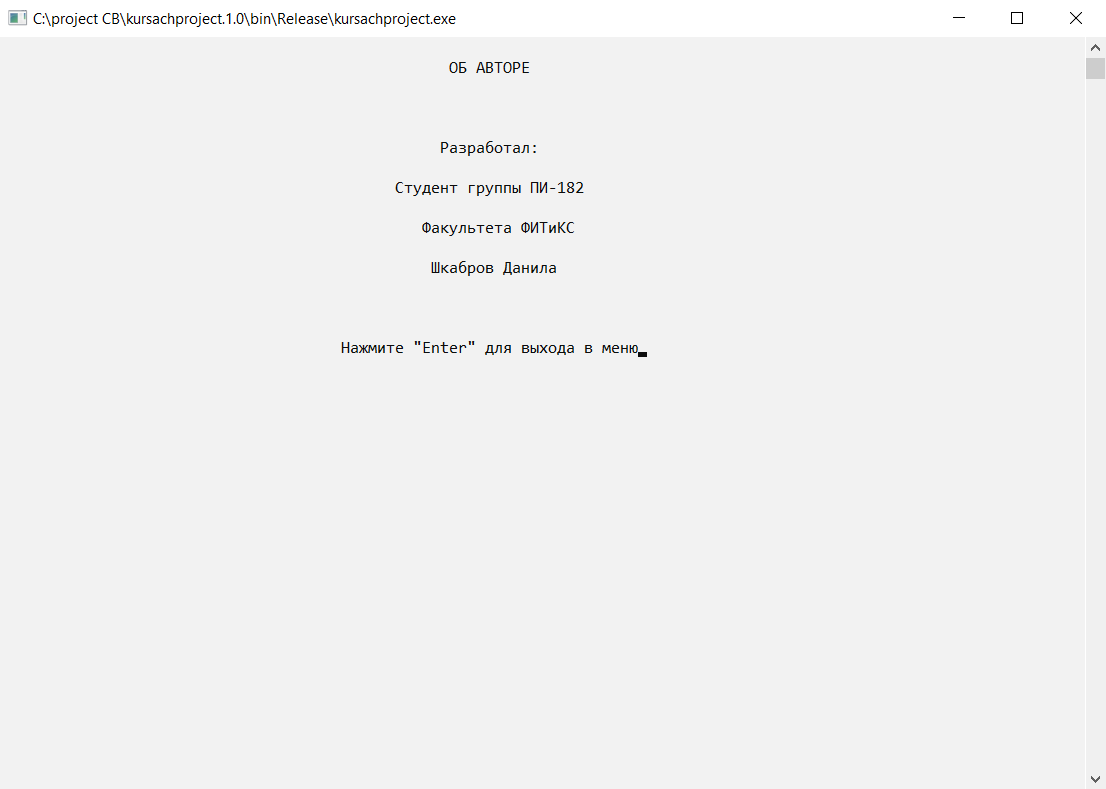


Рисунок 64 – тестирование информации об авторе

**Заключение**

В результате выполнения курсового проекта, были разработаны алгоритмы и отлажены программы:

* Электронная «Записная книжка»
* Программа-игра;
* Анимация;
* Сведения об авторе;
* Выход.

Достоинства программы:

* При поиске в базе данных информация о людях отсортирована по дню, месяцу и году рождения по возрастанию;
* Красивая визуализация игры «Покер»;
* Продуманная до мелочей игра «Покер».

Недостатки:

* База данных чувствительна к регистру.

Игровая программа «Покер» полностью проработана, и может быть использована людьми, для развлечения и улучшения стратегических и логических навыков. Разработанная база данных работает и может быть применена на практике.

Приложение

**main.cpp:**

#include "CApp.h"

#include "Baza\_Dannih.h"

#include <math.h>

#include <locale.h>

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include <conio.h>

#include <graphics.h>

#include <sstream>

#include <windows.h>

#include <winbgim.h>

#include <limits.h>

#include <SFML/Graphics.hpp>

char Colour[3];

using namespace std;

enum ConsoleColor

{

Black = 0,

Blue = 1,

Green = 2,

Cyan = 3,

Red = 4,

Magenta = 5,

Brown = 6,

LightGray = 7,

DarkGray = 8,

LightBlue = 9,

LightGreen = 10,

LightCyan = 11,

LightRed = 12,

LightMagenta = 13,

Yellow = 14,

White = 15

};

void XY(int x, int y)

{

COORD a = { x, y };

HANDLE form = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleCursorPosition(form, a);

}

void Color(int text, int background)

{

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hStdOut, (WORD)((background << 4) | text));

}

void SetColour(char R, char G, char B)

{

Colour[0] = R;

Colour[1] = G;

Colour[2] = B;

}

void menu();

int main();

void animation()

{

srand( time( NULL ) );

sf::RenderWindow window(sf::VideoMode(562, 562), "Animation");

sf::Texture t1;

t1.loadFromFile("Animation/Animation1.bmp");

sf::Texture t2;

t2.loadFromFile("Animation/Animation2.bmp");

sf::Texture t3;

t3.loadFromFile("Animation/Animation3.bmp");

sf::Sprite s;

sf::Sprite s1;

sf::Sprite s2;

s.setTexture(t1);

s1.setTexture(t2);

s2.setTexture(t3);

while (window.isOpen())

{

sf::Event event;

while (window.pollEvent(event))

{

if (event.type == sf::Event::Closed)

window.close();

if (event.key.code == sf::Keyboard::Escape)

window.close();

}

window.draw(s);

window.display();

Sleep(100);

window.draw(s1);

window.display();

Sleep(100);

window.draw(s2);

window.display();

Sleep(100);

}

menu();

}

void author()

{

system("cls");

XY(50, 1);

Color(0, 15);//цвет букв, задний фон

printf("ОБ АВТОРЕ");

XY(49, 5);

printf("Разработал:");

XY(44, 7);

printf("Студент группы ПИ-182");

XY(47, 9);

printf("Факультета ФИТиКС");

XY(48, 11);

printf("Шкабров Данила");

XY(38, 15);

printf("Нажмите \"Enter\" для выхода в меню");

}

void menu()

{

int n;

int key = 0;//начальное положение ключа

system("cls");

do

{

Color(0, 15);

system("cls");

key = (key + 5) % 5;

XY(47, 3);

printf("КУРСОВАЯ РАБОТА");

if (key == 0)

{

XY(46, 9);

Color(4, 15);

printf("=>База данных<=");

}

else

{

XY(48, 9);

Color(0, 15);

printf("База данных");

}

if (key == 1)

{

XY(48, 11);

Color(4, 15);

printf("=>Покер<=");

}

else

{

XY(50, 11);

Color(0, 15);

printf("Покер");

}

if (key == 3)

{

XY(46, 15);

Color(4, 15);

printf("=>Об авторе<=");

}

else

{

XY(48, 15);

Color(0, 15);

printf("Об авторе");

}

if (key == 2)

{

XY(47, 13);

Color(4, 15);

printf("=>Анимация<=");

}

else

{

XY(49, 13);

Color(0, 15);

printf("Анимация");

}

if (key == 4)

{

XY(48, 17);

Color(4, 15);

printf("=>Выход<=");

}

else

{

XY(50, 17);

Color(0, 15);

printf("Выход");

}

{

n = \_getch();

if (n == 80) key++;

if (n == 72) key--;

}

}

while (n != 13);

switch (key)

{

case 0:

{

system("cls");

Baza\_Dannih();

\_getch();

menu();

break;

}

case 1:

{

system("cls");

CApp game;

game.onExecute();

menu();

break;

}

case 2:

{

system("cls");

animation();

menu();

break;

}

case 3:

{

system("cls");

author();

\_getch();

menu();

break;

}

{

}

case 4:

{

system("cls");

exit(1);

menu();

break;

}

}

key = 0;

}

int main(int argc,char\*\* argv)

{

srand( time( NULL ) );

setlocale(LC\_ALL, "Russian");

HWND window\_header = GetConsoleWindow();

SetWindowPos(window\_header, HWND\_TOP, 250, 100, 900, 640, NULL);

menu();

return 0;

}

**CApp.cpp:**

#include "CApp.h"

//Конструктор

CApp::CApp()

{

running = true;

window = NULL;

screen = NULL;

background = NULL;

//--------------CARDS--------------------

cart\_2\_spades = NULL;

cart\_2\_clubs = NULL;

cart\_2\_hearts = NULL;

cart\_2\_diamonds = NULL;

cart\_3\_spades = NULL;

cart\_3\_clubs = NULL;

cart\_3\_hearts = NULL;

cart\_3\_diamonds = NULL;

cart\_4\_spades = NULL;

cart\_4\_clubs = NULL;

cart\_4\_hearts = NULL;

cart\_4\_diamonds = NULL;

cart\_5\_spades = NULL;

cart\_5\_clubs = NULL;

cart\_5\_hearts = NULL;

cart\_5\_diamonds = NULL;

cart\_6\_spades = NULL;

cart\_6\_clubs = NULL;

cart\_6\_hearts = NULL;

cart\_6\_diamonds = NULL;

cart\_7\_spades = NULL;

cart\_7\_clubs = NULL;

cart\_7\_hearts = NULL;

cart\_7\_diamonds = NULL;

cart\_8\_spades = NULL;

cart\_8\_clubs = NULL;

cart\_8\_hearts = NULL;

cart\_8\_diamonds = NULL;

cart\_9\_spades = NULL;

cart\_9\_clubs = NULL;

cart\_9\_hearts = NULL;

cart\_9\_diamonds = NULL;

cart\_10\_spades = NULL;

cart\_10\_clubs = NULL;

cart\_10\_hearts = NULL;

cart\_10\_diamonds = NULL;

cart\_J\_spades = NULL;

cart\_J\_clubs = NULL;

cart\_J\_hearts = NULL;

cart\_J\_diamonds = NULL;

cart\_Q\_spades = NULL;

cart\_Q\_clubs = NULL;

cart\_Q\_hearts = NULL;

cart\_Q\_diamonds = NULL;

cart\_K\_spades = NULL;

cart\_K\_clubs = NULL;

cart\_K\_hearts = NULL;

cart\_K\_diamonds = NULL;

cart\_A\_spades = NULL;

cart\_A\_clubs = NULL;

cart\_A\_hearts = NULL;

cart\_A\_diamonds = NULL;

Back\_Card = NULL;

//--------------CARDS--------------------

snd\_draw = NULL;

snd\_lose = NULL;

snd\_place = NULL;

snd\_win = NULL;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Диструктор

CApp::~CApp()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Главная функция

int CApp::onExecute()

{

if(!onInit())

return -1;

SDL\_Event e;

while(running)

{

while(SDL\_PollEvent(&e))

{

onEvent(&e);

}

onLoop();

onRender();

}

onQuit();

return 0;

}

//--------------CARDS--------------------

void CApp::fold\_card()

{

check\_temp\_card\_straight=0;

check\_temp\_high\_card=0;

temp\_card=0;

//--------------player--------------------

check\_flush\_royalP=0;

check\_straight\_flushP=0;

check\_kareP=0;

check\_tripleP=0;

check\_triple2P=0;

check\_doubleP=0;

check\_double2P=0;

check\_double3P=0;

check\_straighP=0;

check\_flushP=0;

check\_full\_houseP=0;

check\_highP=0;

straight\_tempP=11;

//--------------player(END)--------------

}

//--------------CARDS--------------------

void CApp::check\_card()

{

money-=100;

moneybox+=600;

temp\_card++;

}

//--------------CARDS--------------------

void CApp::bet\_card()

{

money-=200;

moneybox+=1200;

temp\_card++;

}

//--------------CARDS--------------------

void CApp::combinations\_double(int card1, int card2, int \*check\_double, int \*check\_double2, int \*check\_double3)

{

if(\*check\_double==0)

{

if((queue\_cards[card1] == queue\_cards[0] || queue\_cards[card1] == queue\_cards[1] || queue\_cards[card1] == queue\_cards[2] || queue\_cards[card1] == queue\_cards[3]) && (queue\_cards[card2] == queue\_cards[0] || queue\_cards[card2] == queue\_cards[1] || queue\_cards[card2] == queue\_cards[2] || queue\_cards[card2] == queue\_cards[3]))

{

\*check\_double=2;

}

if((queue\_cards[card1] == queue\_cards[4] || queue\_cards[card1] == queue\_cards[5] || queue\_cards[card1] == queue\_cards[6] || queue\_cards[card1] == queue\_cards[7]) && (queue\_cards[card2] == queue\_cards[4] || queue\_cards[card2] == queue\_cards[5] || queue\_cards[card2] == queue\_cards[6] || queue\_cards[card2] == queue\_cards[7]))

{

\*check\_double=3;

}

if((queue\_cards[card1] == queue\_cards[8] || queue\_cards[card1] == queue\_cards[9] || queue\_cards[card1] == queue\_cards[10] || queue\_cards[card1] == queue\_cards[11]) && (queue\_cards[card2] == queue\_cards[8] || queue\_cards[card2] == queue\_cards[9] || queue\_cards[card2] == queue\_cards[10] || queue\_cards[card2] == queue\_cards[11]))

{

\*check\_double=4;

}

if((queue\_cards[card1] == queue\_cards[12] || queue\_cards[card1] == queue\_cards[13] || queue\_cards[card1] == queue\_cards[14] || queue\_cards[card1] == queue\_cards[15]) && (queue\_cards[card2] == queue\_cards[12] || queue\_cards[card2] == queue\_cards[13] || queue\_cards[card2] == queue\_cards[14] || queue\_cards[card2] == queue\_cards[15]))

{

\*check\_double=5;

}

if((queue\_cards[card1] == queue\_cards[16] || queue\_cards[card1] == queue\_cards[17] || queue\_cards[card1] == queue\_cards[18] || queue\_cards[card1] == queue\_cards[19]) && (queue\_cards[card2] == queue\_cards[16] || queue\_cards[card2] == queue\_cards[17] || queue\_cards[card2] == queue\_cards[18] || queue\_cards[card2] == queue\_cards[19]))

{

\*check\_double=6;

}

if((queue\_cards[card1] == queue\_cards[20] || queue\_cards[card1] == queue\_cards[21] || queue\_cards[card1] == queue\_cards[22] || queue\_cards[card1] == queue\_cards[23]) && (queue\_cards[card2] == queue\_cards[20] || queue\_cards[card2] == queue\_cards[21] || queue\_cards[card2] == queue\_cards[22] || queue\_cards[card2] == queue\_cards[23]))

{

\*check\_double=7;

}

if((queue\_cards[card1] == queue\_cards[24] || queue\_cards[card1] == queue\_cards[25] || queue\_cards[card1] == queue\_cards[26] || queue\_cards[card1] == queue\_cards[27]) && (queue\_cards[card2] == queue\_cards[24] || queue\_cards[card2] == queue\_cards[25] || queue\_cards[card2] == queue\_cards[26] || queue\_cards[card2] == queue\_cards[27]))

{

\*check\_double=8;

}

if((queue\_cards[card1] == queue\_cards[28] || queue\_cards[card1] == queue\_cards[29] || queue\_cards[card1] == queue\_cards[30] || queue\_cards[card1] == queue\_cards[31]) && (queue\_cards[card2] == queue\_cards[28] || queue\_cards[card2] == queue\_cards[29] || queue\_cards[card2] == queue\_cards[30] || queue\_cards[card2] == queue\_cards[31]))

{

\*check\_double=9;

}

if((queue\_cards[card1] == queue\_cards[32] || queue\_cards[card1] == queue\_cards[33] || queue\_cards[card1] == queue\_cards[34] || queue\_cards[card1] == queue\_cards[35]) && (queue\_cards[card2] == queue\_cards[32] || queue\_cards[card2] == queue\_cards[33] || queue\_cards[card2] == queue\_cards[34] || queue\_cards[card2] == queue\_cards[35]))

{

\*check\_double=10;

}

if((queue\_cards[card1] == queue\_cards[36] || queue\_cards[card1] == queue\_cards[37] || queue\_cards[card1] == queue\_cards[38] || queue\_cards[card1] == queue\_cards[39]) && (queue\_cards[card2] == queue\_cards[36] || queue\_cards[card2] == queue\_cards[37] || queue\_cards[card2] == queue\_cards[38] || queue\_cards[card2] == queue\_cards[39]))

{

\*check\_double=11;

}

if((queue\_cards[card1] == queue\_cards[40] || queue\_cards[card1] == queue\_cards[41] || queue\_cards[card1] == queue\_cards[42] || queue\_cards[card1] == queue\_cards[43]) && (queue\_cards[card2] == queue\_cards[40] || queue\_cards[card2] == queue\_cards[41] || queue\_cards[card2] == queue\_cards[42] || queue\_cards[card2] == queue\_cards[43]))

{

\*check\_double=12;

}

if((queue\_cards[card1] == queue\_cards[44] || queue\_cards[card1] == queue\_cards[45] || queue\_cards[card1] == queue\_cards[46] || queue\_cards[card1] == queue\_cards[47]) && (queue\_cards[card2] == queue\_cards[44] || queue\_cards[card2] == queue\_cards[45] || queue\_cards[card2] == queue\_cards[46] || queue\_cards[card2] == queue\_cards[47]))

{

\*check\_double=13;

}

if((queue\_cards[card1] == queue\_cards[48] || queue\_cards[card1] == queue\_cards[49] || queue\_cards[card1] == queue\_cards[50] || queue\_cards[card1] == queue\_cards[51]) && (queue\_cards[card2] == queue\_cards[48] || queue\_cards[card2] == queue\_cards[49] || queue\_cards[card2] == queue\_cards[50] || queue\_cards[card2] == queue\_cards[51]))

{

\*check\_double=14;

}

}

else

{

if(\*check\_double2==0&&\*check\_double!=0)

{

if((queue\_cards[card1] == queue\_cards[0] || queue\_cards[card1] == queue\_cards[1] || queue\_cards[card1] == queue\_cards[2] || queue\_cards[card1] == queue\_cards[3]) && (queue\_cards[card2] == queue\_cards[0] || queue\_cards[card2] == queue\_cards[1] || queue\_cards[card2] == queue\_cards[2] || queue\_cards[card2] == queue\_cards[3]))

{

\*check\_double2=2;

}

if((queue\_cards[card1] == queue\_cards[4] || queue\_cards[card1] == queue\_cards[5] || queue\_cards[card1] == queue\_cards[6] || queue\_cards[card1] == queue\_cards[7]) && (queue\_cards[card2] == queue\_cards[4] || queue\_cards[card2] == queue\_cards[5] || queue\_cards[card2] == queue\_cards[6] || queue\_cards[card2] == queue\_cards[7]))

{

\*check\_double2=3;

}

if((queue\_cards[card1] == queue\_cards[8] || queue\_cards[card1] == queue\_cards[9] || queue\_cards[card1] == queue\_cards[10] || queue\_cards[card1] == queue\_cards[11]) && (queue\_cards[card2] == queue\_cards[8] || queue\_cards[card2] == queue\_cards[9] || queue\_cards[card2] == queue\_cards[10] || queue\_cards[card2] == queue\_cards[11]))

{

\*check\_double2=4;

}

if((queue\_cards[card1] == queue\_cards[12] || queue\_cards[card1] == queue\_cards[13] || queue\_cards[card1] == queue\_cards[14] || queue\_cards[card1] == queue\_cards[15]) && (queue\_cards[card2] == queue\_cards[12] || queue\_cards[card2] == queue\_cards[13] || queue\_cards[card2] == queue\_cards[14] || queue\_cards[card2] == queue\_cards[15]))

{

\*check\_double2=5;

}

if((queue\_cards[card1] == queue\_cards[16] || queue\_cards[card1] == queue\_cards[17] || queue\_cards[card1] == queue\_cards[18] || queue\_cards[card1] == queue\_cards[19]) && (queue\_cards[card2] == queue\_cards[16] || queue\_cards[card2] == queue\_cards[17] || queue\_cards[card2] == queue\_cards[18] || queue\_cards[card2] == queue\_cards[19]))

{

\*check\_double2=6;

}

if((queue\_cards[card1] == queue\_cards[20] || queue\_cards[card1] == queue\_cards[21] || queue\_cards[card1] == queue\_cards[22] || queue\_cards[card1] == queue\_cards[23]) && (queue\_cards[card2] == queue\_cards[20] || queue\_cards[card2] == queue\_cards[21] || queue\_cards[card2] == queue\_cards[22] || queue\_cards[card2] == queue\_cards[23]))

{

\*check\_double2=7;

}

if((queue\_cards[card1] == queue\_cards[24] || queue\_cards[card1] == queue\_cards[25] || queue\_cards[card1] == queue\_cards[26] || queue\_cards[card1] == queue\_cards[27]) && (queue\_cards[card2] == queue\_cards[24] || queue\_cards[card2] == queue\_cards[25] || queue\_cards[card2] == queue\_cards[26] || queue\_cards[card2] == queue\_cards[27]))

{

\*check\_double2=8;

}

if((queue\_cards[card1] == queue\_cards[28] || queue\_cards[card1] == queue\_cards[29] || queue\_cards[card1] == queue\_cards[30] || queue\_cards[card1] == queue\_cards[31]) && (queue\_cards[card2] == queue\_cards[28] || queue\_cards[card2] == queue\_cards[29] || queue\_cards[card2] == queue\_cards[30] || queue\_cards[card2] == queue\_cards[31]))

{

\*check\_double2=9;

}

if((queue\_cards[card1] == queue\_cards[32] || queue\_cards[card1] == queue\_cards[33] || queue\_cards[card1] == queue\_cards[34] || queue\_cards[card1] == queue\_cards[35]) && (queue\_cards[card2] == queue\_cards[32] || queue\_cards[card2] == queue\_cards[33] || queue\_cards[card2] == queue\_cards[34] || queue\_cards[card2] == queue\_cards[35]))

{

\*check\_double2=10;

}

if((queue\_cards[card1] == queue\_cards[36] || queue\_cards[card1] == queue\_cards[37] || queue\_cards[card1] == queue\_cards[38] || queue\_cards[card1] == queue\_cards[39]) && (queue\_cards[card2] == queue\_cards[36] || queue\_cards[card2] == queue\_cards[37] || queue\_cards[card2] == queue\_cards[38] || queue\_cards[card2] == queue\_cards[39]))

{

\*check\_double2=11;

}

if((queue\_cards[card1] == queue\_cards[40] || queue\_cards[card1] == queue\_cards[41] || queue\_cards[card1] == queue\_cards[42] || queue\_cards[card1] == queue\_cards[43]) && (queue\_cards[card2] == queue\_cards[40] || queue\_cards[card2] == queue\_cards[41] || queue\_cards[card2] == queue\_cards[42] || queue\_cards[card2] == queue\_cards[43]))

{

\*check\_double2=12;

}

if((queue\_cards[card1] == queue\_cards[44] || queue\_cards[card1] == queue\_cards[45] || queue\_cards[card1] == queue\_cards[46] || queue\_cards[card1] == queue\_cards[47]) && (queue\_cards[card2] == queue\_cards[44] || queue\_cards[card2] == queue\_cards[45] || queue\_cards[card2] == queue\_cards[46] || queue\_cards[card2] == queue\_cards[47]))

{

\*check\_double2=13;

}

if((queue\_cards[card1] == queue\_cards[48] || queue\_cards[card1] == queue\_cards[49] || queue\_cards[card1] == queue\_cards[50] || queue\_cards[card1] == queue\_cards[51]) && (queue\_cards[card2] == queue\_cards[48] || queue\_cards[card2] == queue\_cards[49] || queue\_cards[card2] == queue\_cards[50] || queue\_cards[card2] == queue\_cards[51]))

{

\*check\_double2=14;

}

}

else

{

if(\*check\_double3==0&&\*check\_double2!=0&&\*check\_double!=0)

{

if((queue\_cards[card1] == queue\_cards[0] || queue\_cards[card1] == queue\_cards[1] || queue\_cards[card1] == queue\_cards[2] || queue\_cards[card1] == queue\_cards[3]) && (queue\_cards[card2] == queue\_cards[0] || queue\_cards[card2] == queue\_cards[1] || queue\_cards[card2] == queue\_cards[2] || queue\_cards[card2] == queue\_cards[3]))

{

\*check\_double3=2;

}

if((queue\_cards[card1] == queue\_cards[4] || queue\_cards[card1] == queue\_cards[5] || queue\_cards[card1] == queue\_cards[6] || queue\_cards[card1] == queue\_cards[7]) && (queue\_cards[card2] == queue\_cards[4] || queue\_cards[card2] == queue\_cards[5] || queue\_cards[card2] == queue\_cards[6] || queue\_cards[card2] == queue\_cards[7]))

{

\*check\_double3=3;

}

if((queue\_cards[card1] == queue\_cards[8] || queue\_cards[card1] == queue\_cards[9] || queue\_cards[card1] == queue\_cards[10] || queue\_cards[card1] == queue\_cards[11]) && (queue\_cards[card2] == queue\_cards[8] || queue\_cards[card2] == queue\_cards[9] || queue\_cards[card2] == queue\_cards[10] || queue\_cards[card2] == queue\_cards[11]))

{

\*check\_double3=4;

}

if((queue\_cards[card1] == queue\_cards[12] || queue\_cards[card1] == queue\_cards[13] || queue\_cards[card1] == queue\_cards[14] || queue\_cards[card1] == queue\_cards[15]) && (queue\_cards[card2] == queue\_cards[12] || queue\_cards[card2] == queue\_cards[13] || queue\_cards[card2] == queue\_cards[14] || queue\_cards[card2] == queue\_cards[15]))

{

\*check\_double3=5;

}

if((queue\_cards[card1] == queue\_cards[16] || queue\_cards[card1] == queue\_cards[17] || queue\_cards[card1] == queue\_cards[18] || queue\_cards[card1] == queue\_cards[19]) && (queue\_cards[card2] == queue\_cards[16] || queue\_cards[card2] == queue\_cards[17] || queue\_cards[card2] == queue\_cards[18] || queue\_cards[card2] == queue\_cards[19]))

{

\*check\_double3=6;

}

if((queue\_cards[card1] == queue\_cards[20] || queue\_cards[card1] == queue\_cards[21] || queue\_cards[card1] == queue\_cards[22] || queue\_cards[card1] == queue\_cards[23]) && (queue\_cards[card2] == queue\_cards[20] || queue\_cards[card2] == queue\_cards[21] || queue\_cards[card2] == queue\_cards[22] || queue\_cards[card2] == queue\_cards[23]))

{

\*check\_double3=7;

}

if((queue\_cards[card1] == queue\_cards[24] || queue\_cards[card1] == queue\_cards[25] || queue\_cards[card1] == queue\_cards[26] || queue\_cards[card1] == queue\_cards[27]) && (queue\_cards[card2] == queue\_cards[24] || queue\_cards[card2] == queue\_cards[25] || queue\_cards[card2] == queue\_cards[26] || queue\_cards[card2] == queue\_cards[27]))

{

\*check\_double3=8;

}

if((queue\_cards[card1] == queue\_cards[28] || queue\_cards[card1] == queue\_cards[29] || queue\_cards[card1] == queue\_cards[30] || queue\_cards[card1] == queue\_cards[31]) && (queue\_cards[card2] == queue\_cards[28] || queue\_cards[card2] == queue\_cards[29] || queue\_cards[card2] == queue\_cards[30] || queue\_cards[card2] == queue\_cards[31]))

{

\*check\_double3=9;

}

if((queue\_cards[card1] == queue\_cards[32] || queue\_cards[card1] == queue\_cards[33] || queue\_cards[card1] == queue\_cards[34] || queue\_cards[card1] == queue\_cards[35]) && (queue\_cards[card2] == queue\_cards[32] || queue\_cards[card2] == queue\_cards[33] || queue\_cards[card2] == queue\_cards[34] || queue\_cards[card2] == queue\_cards[35]))

{

\*check\_double3=10;

}

if((queue\_cards[card1] == queue\_cards[36] || queue\_cards[card1] == queue\_cards[37] || queue\_cards[card1] == queue\_cards[38] || queue\_cards[card1] == queue\_cards[39]) && (queue\_cards[card2] == queue\_cards[36] || queue\_cards[card2] == queue\_cards[37] || queue\_cards[card2] == queue\_cards[38] || queue\_cards[card2] == queue\_cards[39]))

{

\*check\_double3=11;

}

if((queue\_cards[card1] == queue\_cards[40] || queue\_cards[card1] == queue\_cards[41] || queue\_cards[card1] == queue\_cards[42] || queue\_cards[card1] == queue\_cards[43]) && (queue\_cards[card2] == queue\_cards[40] || queue\_cards[card2] == queue\_cards[41] || queue\_cards[card2] == queue\_cards[42] || queue\_cards[card2] == queue\_cards[43]))

{

\*check\_double3=12;

}

if((queue\_cards[card1] == queue\_cards[44] || queue\_cards[card1] == queue\_cards[45] || queue\_cards[card1] == queue\_cards[46] || queue\_cards[card1] == queue\_cards[47]) && (queue\_cards[card2] == queue\_cards[44] || queue\_cards[card2] == queue\_cards[45] || queue\_cards[card2] == queue\_cards[46] || queue\_cards[card2] == queue\_cards[47]))

{

\*check\_double3=13;

}

if((queue\_cards[card1] == queue\_cards[48] || queue\_cards[card1] == queue\_cards[49] || queue\_cards[card1] == queue\_cards[50] || queue\_cards[card1] == queue\_cards[51]) && (queue\_cards[card2] == queue\_cards[48] || queue\_cards[card2] == queue\_cards[49] || queue\_cards[card2] == queue\_cards[50] || queue\_cards[card2] == queue\_cards[51]))

{

\*check\_double3=14;

}

}

}

}

if(\*check\_double==\*check\_double2)

\*check\_double2=0;

if(\*check\_double==\*check\_double3)

\*check\_double3=0;

if(\*check\_double2==\*check\_double3)

\*check\_double3=0;

}

//--------------CARDS--------------------

void CApp::combinations\_triple(int card1, int card2, int card3, int \*check\_triple, int \*check\_triple2)

{

if(\*check\_triple==0)

{

if((queue\_cards[card1] == queue\_cards[0] || queue\_cards[card1] == queue\_cards[1] || queue\_cards[card1] == queue\_cards[2] || queue\_cards[card1] == queue\_cards[3]) && (queue\_cards[card2] == queue\_cards[0] || queue\_cards[card2] == queue\_cards[1] || queue\_cards[card2] == queue\_cards[2] || queue\_cards[card2] == queue\_cards[3]) && (queue\_cards[card3] == queue\_cards[0] || queue\_cards[card3] == queue\_cards[1] || queue\_cards[card3] == queue\_cards[2] || queue\_cards[card3] == queue\_cards[3]))

{

\*check\_triple=2;

}

if((queue\_cards[card1] == queue\_cards[4] || queue\_cards[card1] == queue\_cards[5] || queue\_cards[card1] == queue\_cards[6] || queue\_cards[card1] == queue\_cards[7]) && (queue\_cards[card2] == queue\_cards[4] || queue\_cards[card2] == queue\_cards[5] || queue\_cards[card2] == queue\_cards[6] || queue\_cards[card2] == queue\_cards[7]) && (queue\_cards[card3] == queue\_cards[4] || queue\_cards[card3] == queue\_cards[5] || queue\_cards[card3] == queue\_cards[6] || queue\_cards[card3] == queue\_cards[7]))

{

\*check\_triple=3;

}

if((queue\_cards[card1] == queue\_cards[8] || queue\_cards[card1] == queue\_cards[9] || queue\_cards[card1] == queue\_cards[10] || queue\_cards[card1] == queue\_cards[11]) && (queue\_cards[card2] == queue\_cards[8] || queue\_cards[card2] == queue\_cards[9] || queue\_cards[card2] == queue\_cards[10] || queue\_cards[card2] == queue\_cards[11]) && (queue\_cards[card3] == queue\_cards[8] || queue\_cards[card3] == queue\_cards[9] || queue\_cards[card3] == queue\_cards[10] || queue\_cards[card3] == queue\_cards[11]))

{

\*check\_triple=4;

}

if((queue\_cards[card1] == queue\_cards[12] || queue\_cards[card1] == queue\_cards[13] || queue\_cards[card1] == queue\_cards[14] || queue\_cards[card1] == queue\_cards[15]) && (queue\_cards[card2] == queue\_cards[12] || queue\_cards[card2] == queue\_cards[13] || queue\_cards[card2] == queue\_cards[14] || queue\_cards[card2] == queue\_cards[15]) && (queue\_cards[card3] == queue\_cards[12] || queue\_cards[card3] == queue\_cards[13] || queue\_cards[card3] == queue\_cards[14] || queue\_cards[card3] == queue\_cards[15]))

{

\*check\_triple=5;

}

if((queue\_cards[card1] == queue\_cards[16] || queue\_cards[card1] == queue\_cards[17] || queue\_cards[card1] == queue\_cards[18] || queue\_cards[card1] == queue\_cards[19]) && (queue\_cards[card2] == queue\_cards[16] || queue\_cards[card2] == queue\_cards[17] || queue\_cards[card2] == queue\_cards[18] || queue\_cards[card2] == queue\_cards[19]) && (queue\_cards[card3] == queue\_cards[16] || queue\_cards[card3] == queue\_cards[17] || queue\_cards[card3] == queue\_cards[18] || queue\_cards[card3] == queue\_cards[19]))

{

\*check\_triple=6;

}

if((queue\_cards[card1] == queue\_cards[20] || queue\_cards[card1] == queue\_cards[21] || queue\_cards[card1] == queue\_cards[22] || queue\_cards[card1] == queue\_cards[23]) && (queue\_cards[card2] == queue\_cards[20] || queue\_cards[card2] == queue\_cards[21] || queue\_cards[card2] == queue\_cards[22] || queue\_cards[card2] == queue\_cards[23]) && (queue\_cards[card3] == queue\_cards[20] || queue\_cards[card3] == queue\_cards[21] || queue\_cards[card3] == queue\_cards[22] || queue\_cards[card3] == queue\_cards[23]))

{

\*check\_triple=7;

}

if((queue\_cards[card1] == queue\_cards[24] || queue\_cards[card1] == queue\_cards[25] || queue\_cards[card1] == queue\_cards[26] || queue\_cards[card1] == queue\_cards[27]) && (queue\_cards[card2] == queue\_cards[24] || queue\_cards[card2] == queue\_cards[25] || queue\_cards[card2] == queue\_cards[26] || queue\_cards[card2] == queue\_cards[27]) && (queue\_cards[card3] == queue\_cards[24] || queue\_cards[card3] == queue\_cards[25] || queue\_cards[card3] == queue\_cards[26] || queue\_cards[card3] == queue\_cards[27]))

{

\*check\_triple=8;

}

if((queue\_cards[card1] == queue\_cards[28] || queue\_cards[card1] == queue\_cards[29] || queue\_cards[card1] == queue\_cards[30] || queue\_cards[card1] == queue\_cards[31]) && (queue\_cards[card2] == queue\_cards[28] || queue\_cards[card2] == queue\_cards[29] || queue\_cards[card2] == queue\_cards[30] || queue\_cards[card2] == queue\_cards[31]) && (queue\_cards[card3] == queue\_cards[28] || queue\_cards[card3] == queue\_cards[29] || queue\_cards[card3] == queue\_cards[30] || queue\_cards[card3] == queue\_cards[31]))

{

\*check\_triple=9;

}

if((queue\_cards[card1] == queue\_cards[32] || queue\_cards[card1] == queue\_cards[33] || queue\_cards[card1] == queue\_cards[34] || queue\_cards[card1] == queue\_cards[35]) && (queue\_cards[card2] == queue\_cards[32] || queue\_cards[card2] == queue\_cards[33] || queue\_cards[card2] == queue\_cards[34] || queue\_cards[card2] == queue\_cards[35]) && (queue\_cards[card3] == queue\_cards[32] || queue\_cards[card3] == queue\_cards[33] || queue\_cards[card3] == queue\_cards[34] || queue\_cards[card3] == queue\_cards[35]))

{

\*check\_triple=10;

}

if((queue\_cards[card1] == queue\_cards[36] || queue\_cards[card1] == queue\_cards[37] || queue\_cards[card1] == queue\_cards[38] || queue\_cards[card1] == queue\_cards[39]) && (queue\_cards[card2] == queue\_cards[36] || queue\_cards[card2] == queue\_cards[37] || queue\_cards[card2] == queue\_cards[38] || queue\_cards[card2] == queue\_cards[39]) && (queue\_cards[card3] == queue\_cards[36] || queue\_cards[card3] == queue\_cards[37] || queue\_cards[card3] == queue\_cards[38] || queue\_cards[card3] == queue\_cards[39]))

{

\*check\_triple=11;

}

if((queue\_cards[card1] == queue\_cards[40] || queue\_cards[card1] == queue\_cards[41] || queue\_cards[card1] == queue\_cards[42] || queue\_cards[card1] == queue\_cards[43]) && (queue\_cards[card2] == queue\_cards[40] || queue\_cards[card2] == queue\_cards[41] || queue\_cards[card2] == queue\_cards[42] || queue\_cards[card2] == queue\_cards[43]) && (queue\_cards[card3] == queue\_cards[40] || queue\_cards[card3] == queue\_cards[41] || queue\_cards[card3] == queue\_cards[42] || queue\_cards[card3] == queue\_cards[43]))

{

\*check\_triple=12;

}

if((queue\_cards[card1] == queue\_cards[44] || queue\_cards[card1] == queue\_cards[45] || queue\_cards[card1] == queue\_cards[46] || queue\_cards[card1] == queue\_cards[47]) && (queue\_cards[card2] == queue\_cards[44] || queue\_cards[card2] == queue\_cards[45] || queue\_cards[card2] == queue\_cards[46] || queue\_cards[card2] == queue\_cards[47]) && (queue\_cards[card3] == queue\_cards[44] || queue\_cards[card3] == queue\_cards[45] || queue\_cards[card3] == queue\_cards[46] || queue\_cards[card3] == queue\_cards[47]))

{

\*check\_triple=13;

}

if((queue\_cards[card1] == queue\_cards[48] || queue\_cards[card1] == queue\_cards[49] || queue\_cards[card1] == queue\_cards[50] || queue\_cards[card1] == queue\_cards[51]) && (queue\_cards[card2] == queue\_cards[48] || queue\_cards[card2] == queue\_cards[49] || queue\_cards[card2] == queue\_cards[50] || queue\_cards[card2] == queue\_cards[51]) && (queue\_cards[card3] == queue\_cards[48] || queue\_cards[card3] == queue\_cards[49] || queue\_cards[card3] == queue\_cards[50] || queue\_cards[card3] == queue\_cards[51]))

{

\*check\_triple=14;

}

}

else

{

if(\*check\_triple2==0)

{

if((queue\_cards[card1] == queue\_cards[0] || queue\_cards[card1] == queue\_cards[1] || queue\_cards[card1] == queue\_cards[2] || queue\_cards[card1] == queue\_cards[3]) && (queue\_cards[card2] == queue\_cards[0] || queue\_cards[card2] == queue\_cards[1] || queue\_cards[card2] == queue\_cards[2] || queue\_cards[card2] == queue\_cards[3]) && (queue\_cards[card3] == queue\_cards[0] || queue\_cards[card3] == queue\_cards[1] || queue\_cards[card3] == queue\_cards[2] || queue\_cards[card3] == queue\_cards[3]))

{

\*check\_triple2=2;

}

if((queue\_cards[card1] == queue\_cards[4] || queue\_cards[card1] == queue\_cards[5] || queue\_cards[card1] == queue\_cards[6] || queue\_cards[card1] == queue\_cards[7]) && (queue\_cards[card2] == queue\_cards[4] || queue\_cards[card2] == queue\_cards[5] || queue\_cards[card2] == queue\_cards[6] || queue\_cards[card2] == queue\_cards[7]) && (queue\_cards[card3] == queue\_cards[4] || queue\_cards[card3] == queue\_cards[5] || queue\_cards[card3] == queue\_cards[6] || queue\_cards[card3] == queue\_cards[7]))

{

\*check\_triple2=3;

}

if((queue\_cards[card1] == queue\_cards[8] || queue\_cards[card1] == queue\_cards[9] || queue\_cards[card1] == queue\_cards[10] || queue\_cards[card1] == queue\_cards[11]) && (queue\_cards[card2] == queue\_cards[8] || queue\_cards[card2] == queue\_cards[9] || queue\_cards[card2] == queue\_cards[10] || queue\_cards[card2] == queue\_cards[11]) && (queue\_cards[card3] == queue\_cards[8] || queue\_cards[card3] == queue\_cards[9] || queue\_cards[card3] == queue\_cards[10] || queue\_cards[card3] == queue\_cards[11]))

{

\*check\_triple2=4;

}

if((queue\_cards[card1] == queue\_cards[12] || queue\_cards[card1] == queue\_cards[13] || queue\_cards[card1] == queue\_cards[14] || queue\_cards[card1] == queue\_cards[15]) && (queue\_cards[card2] == queue\_cards[12] || queue\_cards[card2] == queue\_cards[13] || queue\_cards[card2] == queue\_cards[14] || queue\_cards[card2] == queue\_cards[15]) && (queue\_cards[card3] == queue\_cards[12] || queue\_cards[card3] == queue\_cards[13] || queue\_cards[card3] == queue\_cards[14] || queue\_cards[card3] == queue\_cards[15]))

{

\*check\_triple2=5;

}

if((queue\_cards[card1] == queue\_cards[16] || queue\_cards[card1] == queue\_cards[17] || queue\_cards[card1] == queue\_cards[18] || queue\_cards[card1] == queue\_cards[19]) && (queue\_cards[card2] == queue\_cards[16] || queue\_cards[card2] == queue\_cards[17] || queue\_cards[card2] == queue\_cards[18] || queue\_cards[card2] == queue\_cards[19]) && (queue\_cards[card3] == queue\_cards[16] || queue\_cards[card3] == queue\_cards[17] || queue\_cards[card3] == queue\_cards[18] || queue\_cards[card3] == queue\_cards[19]))

{

\*check\_triple2=6;

}

if((queue\_cards[card1] == queue\_cards[20] || queue\_cards[card1] == queue\_cards[21] || queue\_cards[card1] == queue\_cards[22] || queue\_cards[card1] == queue\_cards[23]) && (queue\_cards[card2] == queue\_cards[20] || queue\_cards[card2] == queue\_cards[21] || queue\_cards[card2] == queue\_cards[22] || queue\_cards[card2] == queue\_cards[23]) && (queue\_cards[card3] == queue\_cards[20] || queue\_cards[card3] == queue\_cards[21] || queue\_cards[card3] == queue\_cards[22] || queue\_cards[card3] == queue\_cards[23]))

{

\*check\_triple=7;

}

if((queue\_cards[card1] == queue\_cards[24] || queue\_cards[card1] == queue\_cards[25] || queue\_cards[card1] == queue\_cards[26] || queue\_cards[card1] == queue\_cards[27]) && (queue\_cards[card2] == queue\_cards[24] || queue\_cards[card2] == queue\_cards[25] || queue\_cards[card2] == queue\_cards[26] || queue\_cards[card2] == queue\_cards[27]) && (queue\_cards[card3] == queue\_cards[24] || queue\_cards[card3] == queue\_cards[25] || queue\_cards[card3] == queue\_cards[26] || queue\_cards[card3] == queue\_cards[27]))

{

\*check\_triple2=8;

}

if((queue\_cards[card1] == queue\_cards[28] || queue\_cards[card1] == queue\_cards[29] || queue\_cards[card1] == queue\_cards[30] || queue\_cards[card1] == queue\_cards[31]) && (queue\_cards[card2] == queue\_cards[28] || queue\_cards[card2] == queue\_cards[29] || queue\_cards[card2] == queue\_cards[30] || queue\_cards[card2] == queue\_cards[31]) && (queue\_cards[card3] == queue\_cards[28] || queue\_cards[card3] == queue\_cards[29] || queue\_cards[card3] == queue\_cards[30] || queue\_cards[card3] == queue\_cards[31]))

{

\*check\_triple2=9;

}

if((queue\_cards[card1] == queue\_cards[32] || queue\_cards[card1] == queue\_cards[33] || queue\_cards[card1] == queue\_cards[34] || queue\_cards[card1] == queue\_cards[35]) && (queue\_cards[card2] == queue\_cards[32] || queue\_cards[card2] == queue\_cards[33] || queue\_cards[card2] == queue\_cards[34] || queue\_cards[card2] == queue\_cards[35]) && (queue\_cards[card3] == queue\_cards[32] || queue\_cards[card3] == queue\_cards[33] || queue\_cards[card3] == queue\_cards[34] || queue\_cards[card3] == queue\_cards[35]))

{

\*check\_triple2=10;

}

if((queue\_cards[card1] == queue\_cards[36] || queue\_cards[card1] == queue\_cards[37] || queue\_cards[card1] == queue\_cards[38] || queue\_cards[card1] == queue\_cards[39]) && (queue\_cards[card2] == queue\_cards[36] || queue\_cards[card2] == queue\_cards[37] || queue\_cards[card2] == queue\_cards[38] || queue\_cards[card2] == queue\_cards[39]) && (queue\_cards[card3] == queue\_cards[36] || queue\_cards[card3] == queue\_cards[37] || queue\_cards[card3] == queue\_cards[38] || queue\_cards[card3] == queue\_cards[39]))

{

\*check\_triple2=11;

}

if((queue\_cards[card1] == queue\_cards[40] || queue\_cards[card1] == queue\_cards[41] || queue\_cards[card1] == queue\_cards[42] || queue\_cards[card1] == queue\_cards[43]) && (queue\_cards[card2] == queue\_cards[40] || queue\_cards[card2] == queue\_cards[41] || queue\_cards[card2] == queue\_cards[42] || queue\_cards[card2] == queue\_cards[43]) && (queue\_cards[card3] == queue\_cards[40] || queue\_cards[card3] == queue\_cards[41] || queue\_cards[card3] == queue\_cards[42] || queue\_cards[card3] == queue\_cards[43]))

{

\*check\_triple2=12;

}

if((queue\_cards[card1] == queue\_cards[44] || queue\_cards[card1] == queue\_cards[45] || queue\_cards[card1] == queue\_cards[46] || queue\_cards[card1] == queue\_cards[47]) && (queue\_cards[card2] == queue\_cards[44] || queue\_cards[card2] == queue\_cards[45] || queue\_cards[card2] == queue\_cards[46] || queue\_cards[card2] == queue\_cards[47]) && (queue\_cards[card3] == queue\_cards[44] || queue\_cards[card3] == queue\_cards[45] || queue\_cards[card3] == queue\_cards[46] || queue\_cards[card3] == queue\_cards[47]))

{

\*check\_triple2=13;

}

if((queue\_cards[card1] == queue\_cards[48] || queue\_cards[card1] == queue\_cards[49] || queue\_cards[card1] == queue\_cards[50] || queue\_cards[card1] == queue\_cards[51]) && (queue\_cards[card2] == queue\_cards[48] || queue\_cards[card2] == queue\_cards[49] || queue\_cards[card2] == queue\_cards[50] || queue\_cards[card2] == queue\_cards[51]) && (queue\_cards[card3] == queue\_cards[48] || queue\_cards[card3] == queue\_cards[49] || queue\_cards[card3] == queue\_cards[50] || queue\_cards[card3] == queue\_cards[51]))

{

\*check\_triple2=14;

}

}

}

if(\*check\_triple==\*check\_triple2)

\*check\_triple2=0;

}

//--------------CARDS--------------------

void CApp::combinations\_kare(int card1, int card2, int card3, int card4, int \*check\_kare)

{

if((queue\_cards[card1] == queue\_cards[0] || queue\_cards[card1] == queue\_cards[1] || queue\_cards[card1] == queue\_cards[2] || queue\_cards[card1] == queue\_cards[3]) && (queue\_cards[card2] == queue\_cards[0] || queue\_cards[card2] == queue\_cards[1] || queue\_cards[card2] == queue\_cards[2] || queue\_cards[card2] == queue\_cards[3]) && (queue\_cards[card3] == queue\_cards[0] || queue\_cards[card3] == queue\_cards[1] || queue\_cards[card3] == queue\_cards[2] || queue\_cards[card3] == queue\_cards[3]) && (queue\_cards[card4] == queue\_cards[0] || queue\_cards[card4] == queue\_cards[1] || queue\_cards[card4] == queue\_cards[2] || queue\_cards[card4] == queue\_cards[3]))

{

\*check\_kare=2;

}

if((queue\_cards[card1] == queue\_cards[4] || queue\_cards[card1] == queue\_cards[5] || queue\_cards[card1] == queue\_cards[6] || queue\_cards[card1] == queue\_cards[7]) && (queue\_cards[card2] == queue\_cards[4] || queue\_cards[card2] == queue\_cards[5] || queue\_cards[card2] == queue\_cards[6] || queue\_cards[card2] == queue\_cards[7]) && (queue\_cards[card3] == queue\_cards[4] || queue\_cards[card3] == queue\_cards[5] || queue\_cards[card3] == queue\_cards[6] || queue\_cards[card3] == queue\_cards[7]) && (queue\_cards[card4] == queue\_cards[4] || queue\_cards[card4] == queue\_cards[5] || queue\_cards[card4] == queue\_cards[6] || queue\_cards[card4] == queue\_cards[7]))

{

\*check\_kare=3;

}

if((queue\_cards[card1] == queue\_cards[8] || queue\_cards[card1] == queue\_cards[9] || queue\_cards[card1] == queue\_cards[10] || queue\_cards[card1] == queue\_cards[11]) && (queue\_cards[card2] == queue\_cards[8] || queue\_cards[card2] == queue\_cards[9] || queue\_cards[card2] == queue\_cards[10] || queue\_cards[card2] == queue\_cards[11]) && (queue\_cards[card3] == queue\_cards[8] || queue\_cards[card3] == queue\_cards[9] || queue\_cards[card3] == queue\_cards[10] || queue\_cards[card3] == queue\_cards[11]) && (queue\_cards[card4] == queue\_cards[8] || queue\_cards[card4] == queue\_cards[9] || queue\_cards[card4] == queue\_cards[10] || queue\_cards[card4] == queue\_cards[11]))

{

\*check\_kare=4;

}

if((queue\_cards[card1] == queue\_cards[12] || queue\_cards[card1] == queue\_cards[13] || queue\_cards[card1] == queue\_cards[14] || queue\_cards[card1] == queue\_cards[15]) && (queue\_cards[card2] == queue\_cards[12] || queue\_cards[card2] == queue\_cards[13] || queue\_cards[card2] == queue\_cards[14] || queue\_cards[card2] == queue\_cards[15]) && (queue\_cards[card3] == queue\_cards[12] || queue\_cards[card3] == queue\_cards[13] || queue\_cards[card3] == queue\_cards[14] || queue\_cards[card3] == queue\_cards[15]) && (queue\_cards[card4] == queue\_cards[12] || queue\_cards[card4] == queue\_cards[13] || queue\_cards[card4] == queue\_cards[14] || queue\_cards[card4] == queue\_cards[15]))

{

\*check\_kare=5;

}

if((queue\_cards[card1] == queue\_cards[16] || queue\_cards[card1] == queue\_cards[17] || queue\_cards[card1] == queue\_cards[18] || queue\_cards[card1] == queue\_cards[19]) && (queue\_cards[card2] == queue\_cards[16] || queue\_cards[card2] == queue\_cards[17] || queue\_cards[card2] == queue\_cards[18] || queue\_cards[card2] == queue\_cards[19]) && (queue\_cards[card3] == queue\_cards[16] || queue\_cards[card3] == queue\_cards[17] || queue\_cards[card3] == queue\_cards[18] || queue\_cards[card3] == queue\_cards[19]) && (queue\_cards[card4] == queue\_cards[16] || queue\_cards[card4] == queue\_cards[17] || queue\_cards[card4] == queue\_cards[18] || queue\_cards[card4] == queue\_cards[19]))

{

\*check\_kare=6;

}

if((queue\_cards[card1] == queue\_cards[20] || queue\_cards[card1] == queue\_cards[21] || queue\_cards[card1] == queue\_cards[22] || queue\_cards[card1] == queue\_cards[23]) && (queue\_cards[card2] == queue\_cards[20] || queue\_cards[card2] == queue\_cards[21] || queue\_cards[card2] == queue\_cards[22] || queue\_cards[card2] == queue\_cards[23]) && (queue\_cards[card3] == queue\_cards[20] || queue\_cards[card3] == queue\_cards[21] || queue\_cards[card3] == queue\_cards[22] || queue\_cards[card3] == queue\_cards[23]) && (queue\_cards[card4] == queue\_cards[20] || queue\_cards[card4] == queue\_cards[21] || queue\_cards[card4] == queue\_cards[22] || queue\_cards[card4] == queue\_cards[23]))

{

\*check\_kare=7;

}

if((queue\_cards[card1] == queue\_cards[24] || queue\_cards[card1] == queue\_cards[25] || queue\_cards[card1] == queue\_cards[26] || queue\_cards[card1] == queue\_cards[27]) && (queue\_cards[card2] == queue\_cards[24] || queue\_cards[card2] == queue\_cards[25] || queue\_cards[card2] == queue\_cards[26] || queue\_cards[card2] == queue\_cards[27]) && (queue\_cards[card3] == queue\_cards[24] || queue\_cards[card3] == queue\_cards[25] || queue\_cards[card3] == queue\_cards[26] || queue\_cards[card3] == queue\_cards[27]) && (queue\_cards[card4] == queue\_cards[24] || queue\_cards[card4] == queue\_cards[25] || queue\_cards[card4] == queue\_cards[26] || queue\_cards[card4] == queue\_cards[27]))

{

\*check\_kare=8;

}

if((queue\_cards[card1] == queue\_cards[28] || queue\_cards[card1] == queue\_cards[29] || queue\_cards[card1] == queue\_cards[30] || queue\_cards[card1] == queue\_cards[31]) && (queue\_cards[card2] == queue\_cards[28] || queue\_cards[card2] == queue\_cards[29] || queue\_cards[card2] == queue\_cards[30] || queue\_cards[card2] == queue\_cards[31]) && (queue\_cards[card3] == queue\_cards[28] || queue\_cards[card3] == queue\_cards[29] || queue\_cards[card3] == queue\_cards[30] || queue\_cards[card3] == queue\_cards[31]) && (queue\_cards[card4] == queue\_cards[28] || queue\_cards[card4] == queue\_cards[29] || queue\_cards[card4] == queue\_cards[30] || queue\_cards[card4] == queue\_cards[31]))

{

\*check\_kare=9;

}

if((queue\_cards[card1] == queue\_cards[32] || queue\_cards[card1] == queue\_cards[33] || queue\_cards[card1] == queue\_cards[34] || queue\_cards[card1] == queue\_cards[35]) && (queue\_cards[card2] == queue\_cards[32] || queue\_cards[card2] == queue\_cards[33] || queue\_cards[card2] == queue\_cards[34] || queue\_cards[card2] == queue\_cards[35]) && (queue\_cards[card3] == queue\_cards[32] || queue\_cards[card3] == queue\_cards[33] || queue\_cards[card3] == queue\_cards[34] || queue\_cards[card3] == queue\_cards[35]) && (queue\_cards[card4] == queue\_cards[32] || queue\_cards[card4] == queue\_cards[33] || queue\_cards[card4] == queue\_cards[34] || queue\_cards[card4] == queue\_cards[35]))

{

\*check\_kare=10;

}

if((queue\_cards[card1] == queue\_cards[36] || queue\_cards[card1] == queue\_cards[37] || queue\_cards[card1] == queue\_cards[38] || queue\_cards[card1] == queue\_cards[39]) && (queue\_cards[card2] == queue\_cards[36] || queue\_cards[card2] == queue\_cards[37] || queue\_cards[card2] == queue\_cards[38] || queue\_cards[card2] == queue\_cards[39]) && (queue\_cards[card3] == queue\_cards[36] || queue\_cards[card3] == queue\_cards[37] || queue\_cards[card3] == queue\_cards[38] || queue\_cards[card3] == queue\_cards[39]) && (queue\_cards[card4] == queue\_cards[36] || queue\_cards[card4] == queue\_cards[37] || queue\_cards[card4] == queue\_cards[38] || queue\_cards[card4] == queue\_cards[39]))

{

\*check\_kare=11;

}

if((queue\_cards[card1] == queue\_cards[40] || queue\_cards[card1] == queue\_cards[41] || queue\_cards[card1] == queue\_cards[42] || queue\_cards[card1] == queue\_cards[43]) && (queue\_cards[card2] == queue\_cards[40] || queue\_cards[card2] == queue\_cards[41] || queue\_cards[card2] == queue\_cards[42] || queue\_cards[card2] == queue\_cards[43]) && (queue\_cards[card3] == queue\_cards[40] || queue\_cards[card3] == queue\_cards[41] || queue\_cards[card3] == queue\_cards[42] || queue\_cards[card3] == queue\_cards[43]) && (queue\_cards[card4] == queue\_cards[40] || queue\_cards[card4] == queue\_cards[41] || queue\_cards[card4] == queue\_cards[42] || queue\_cards[card4] == queue\_cards[43]))

{

\*check\_kare=12;

}

if((queue\_cards[card1] == queue\_cards[44] || queue\_cards[card1] == queue\_cards[45] || queue\_cards[card1] == queue\_cards[46] || queue\_cards[card1] == queue\_cards[47]) && (queue\_cards[card2] == queue\_cards[44] || queue\_cards[card2] == queue\_cards[45] || queue\_cards[card2] == queue\_cards[46] || queue\_cards[card2] == queue\_cards[47]) && (queue\_cards[card3] == queue\_cards[44] || queue\_cards[card3] == queue\_cards[45] || queue\_cards[card3] == queue\_cards[46] || queue\_cards[card3] == queue\_cards[47]) && (queue\_cards[card4] == queue\_cards[44] || queue\_cards[card4] == queue\_cards[45] || queue\_cards[card4] == queue\_cards[46] || queue\_cards[card4] == queue\_cards[47]))

{

\*check\_kare=13;

}

if((queue\_cards[card1] == queue\_cards[48] || queue\_cards[card1] == queue\_cards[49] || queue\_cards[card1] == queue\_cards[50] || queue\_cards[card1] == queue\_cards[51]) && (queue\_cards[card2] == queue\_cards[48] || queue\_cards[card2] == queue\_cards[49] || queue\_cards[card2] == queue\_cards[50] || queue\_cards[card2] == queue\_cards[51]) && (queue\_cards[card3] == queue\_cards[48] || queue\_cards[card3] == queue\_cards[49] || queue\_cards[card3] == queue\_cards[50] || queue\_cards[card3] == queue\_cards[51]) && (queue\_cards[card4] == queue\_cards[48] || queue\_cards[card4] == queue\_cards[49] || queue\_cards[card4] == queue\_cards[50] || queue\_cards[card4] == queue\_cards[51]))

{

\*check\_kare=14;

}

}

//--------------CARDS--------------------

void CApp::combinations\_straight\_flash(int card1, int card2, int card3, int card4, int card5, int \*check\_flush\_royal, int \*check\_straight\_flush)

{

if((queue\_cards[card1] == queue\_cards[0]||queue\_cards[card1] == queue\_cards[4]||queue\_cards[card1] == queue\_cards[8]||queue\_cards[card1] == queue\_cards[12]||queue\_cards[card1] == queue\_cards[16]) && (queue\_cards[card2] == queue\_cards[0]||queue\_cards[card2] == queue\_cards[4]||queue\_cards[card2] == queue\_cards[8]||queue\_cards[card2] == queue\_cards[12]||queue\_cards[card2] == queue\_cards[16]) && (queue\_cards[card3] == queue\_cards[0]||queue\_cards[card3] == queue\_cards[4]||queue\_cards[card3] == queue\_cards[8]||queue\_cards[card3] == queue\_cards[12]||queue\_cards[card3] == queue\_cards[16]) && (queue\_cards[card4] == queue\_cards[0]||queue\_cards[card4] == queue\_cards[4]||queue\_cards[card4] == queue\_cards[8]||queue\_cards[card4] == queue\_cards[12]||queue\_cards[card4] == queue\_cards[16]) && (queue\_cards[card5] == queue\_cards[0]||queue\_cards[card5] == queue\_cards[4]||queue\_cards[card5] == queue\_cards[8]||queue\_cards[card5] == queue\_cards[12]||queue\_cards[card5] == queue\_cards[16]))

{

\*check\_straight\_flush=1;

}

if((queue\_cards[card1] == queue\_cards[1]||queue\_cards[card1] == queue\_cards[5]||queue\_cards[card1] == queue\_cards[9]||queue\_cards[card1] == queue\_cards[13]||queue\_cards[card1] == queue\_cards[17]) && (queue\_cards[card2] == queue\_cards[1]||queue\_cards[card2] == queue\_cards[5]||queue\_cards[card2] == queue\_cards[9]||queue\_cards[card2] == queue\_cards[13]||queue\_cards[card2] == queue\_cards[17]) && (queue\_cards[card3] == queue\_cards[1]||queue\_cards[card3] == queue\_cards[5]||queue\_cards[card3] == queue\_cards[9]||queue\_cards[card3] == queue\_cards[13]||queue\_cards[card3] == queue\_cards[17]) && (queue\_cards[card4] == queue\_cards[1]||queue\_cards[card4] == queue\_cards[5]||queue\_cards[card4] == queue\_cards[9]||queue\_cards[card4] == queue\_cards[13]||queue\_cards[card4] == queue\_cards[17]) && (queue\_cards[card5] == queue\_cards[1]||queue\_cards[card5] == queue\_cards[5]||queue\_cards[card5] == queue\_cards[9]||queue\_cards[card5] == queue\_cards[13]||queue\_cards[card5] == queue\_cards[17]))

{

\*check\_straight\_flush=1;

}

if((queue\_cards[card1] == queue\_cards[2]||queue\_cards[card1] == queue\_cards[6]||queue\_cards[card1] == queue\_cards[10]||queue\_cards[card1] == queue\_cards[14]||queue\_cards[card1] == queue\_cards[18]) && (queue\_cards[card2] == queue\_cards[2]||queue\_cards[card2] == queue\_cards[6]||queue\_cards[card2] == queue\_cards[10]||queue\_cards[card2] == queue\_cards[14]||queue\_cards[card2] == queue\_cards[18]) && (queue\_cards[card3] == queue\_cards[2]||queue\_cards[card3] == queue\_cards[6]||queue\_cards[card3] == queue\_cards[10]||queue\_cards[card3] == queue\_cards[14]||queue\_cards[card3] == queue\_cards[18]) && (queue\_cards[card4] == queue\_cards[2]||queue\_cards[card4] == queue\_cards[6]||queue\_cards[card4] == queue\_cards[10]||queue\_cards[card4] == queue\_cards[14]||queue\_cards[card4] == queue\_cards[18]) && (queue\_cards[card5] == queue\_cards[2]||queue\_cards[card5] == queue\_cards[6]||queue\_cards[card5] == queue\_cards[10]||queue\_cards[card5] == queue\_cards[14]||queue\_cards[card5] == queue\_cards[18]))

{

\*check\_straight\_flush=1;

}

if((queue\_cards[card1] == queue\_cards[3]||queue\_cards[card1] == queue\_cards[7]||queue\_cards[card1] == queue\_cards[11]||queue\_cards[card1] == queue\_cards[15]||queue\_cards[card1] == queue\_cards[19]) && (queue\_cards[card2] == queue\_cards[3]||queue\_cards[card2] == queue\_cards[7]||queue\_cards[card2] == queue\_cards[11]||queue\_cards[card2] == queue\_cards[15]||queue\_cards[card2] == queue\_cards[19]) && (queue\_cards[card3] == queue\_cards[3]||queue\_cards[card3] == queue\_cards[7]||queue\_cards[card3] == queue\_cards[11]||queue\_cards[card3] == queue\_cards[15]||queue\_cards[card3] == queue\_cards[19]) && (queue\_cards[card4] == queue\_cards[3]||queue\_cards[card4] == queue\_cards[7]||queue\_cards[card4] == queue\_cards[11]||queue\_cards[card4] == queue\_cards[15]||queue\_cards[card4] == queue\_cards[19]) && (queue\_cards[card5] == queue\_cards[3]||queue\_cards[card5] == queue\_cards[7]||queue\_cards[card5] == queue\_cards[11]||queue\_cards[card5] == queue\_cards[15]||queue\_cards[card5] == queue\_cards[19]))

{

\*check\_straight\_flush=1;

}

if((queue\_cards[card1] == queue\_cards[4]||queue\_cards[card1] == queue\_cards[8]||queue\_cards[card1] == queue\_cards[12]||queue\_cards[card1] == queue\_cards[16]||queue\_cards[card1] == queue\_cards[20]) && (queue\_cards[card2] == queue\_cards[4]||queue\_cards[card2] == queue\_cards[8]||queue\_cards[card2] == queue\_cards[12]||queue\_cards[card2] == queue\_cards[16]||queue\_cards[card2] == queue\_cards[20]) && (queue\_cards[card3] == queue\_cards[4]||queue\_cards[card3] == queue\_cards[8]||queue\_cards[card3] == queue\_cards[12]||queue\_cards[card3] == queue\_cards[16]||queue\_cards[card3] == queue\_cards[20]) && (queue\_cards[card4] == queue\_cards[4]||queue\_cards[card4] == queue\_cards[8]||queue\_cards[card4] == queue\_cards[12]||queue\_cards[card4] == queue\_cards[16]||queue\_cards[card4] == queue\_cards[20]) && (queue\_cards[card5] == queue\_cards[4]||queue\_cards[card5] == queue\_cards[8]||queue\_cards[card5] == queue\_cards[12]||queue\_cards[card5] == queue\_cards[16]||queue\_cards[card5] == queue\_cards[20]))

{

\*check\_straight\_flush=2;

}

if((queue\_cards[card1] == queue\_cards[5]||queue\_cards[card1] == queue\_cards[9]||queue\_cards[card1] == queue\_cards[13]||queue\_cards[card1] == queue\_cards[17]||queue\_cards[card1] == queue\_cards[21]) && (queue\_cards[card2] == queue\_cards[5]||queue\_cards[card2] == queue\_cards[9]||queue\_cards[card2] == queue\_cards[13]||queue\_cards[card2] == queue\_cards[17]||queue\_cards[card2] == queue\_cards[21]) && (queue\_cards[card3] == queue\_cards[5]||queue\_cards[card3] == queue\_cards[9]||queue\_cards[card3] == queue\_cards[13]||queue\_cards[card3] == queue\_cards[17]||queue\_cards[card3] == queue\_cards[21]) && (queue\_cards[card4] == queue\_cards[5]||queue\_cards[card4] == queue\_cards[9]||queue\_cards[card4] == queue\_cards[13]||queue\_cards[card4] == queue\_cards[17]||queue\_cards[card4] == queue\_cards[21]) && (queue\_cards[card5] == queue\_cards[5]||queue\_cards[card5] == queue\_cards[9]||queue\_cards[card5] == queue\_cards[13]||queue\_cards[card5] == queue\_cards[17]||queue\_cards[card5] == queue\_cards[21]))

{

\*check\_straight\_flush=2;

}

if((queue\_cards[card1] == queue\_cards[6]||queue\_cards[card1] == queue\_cards[10]||queue\_cards[card1] == queue\_cards[14]||queue\_cards[card1] == queue\_cards[18]||queue\_cards[card1] == queue\_cards[22]) && (queue\_cards[card2] == queue\_cards[6]||queue\_cards[card2] == queue\_cards[10]||queue\_cards[card2] == queue\_cards[14]||queue\_cards[card2] == queue\_cards[18]||queue\_cards[card2] == queue\_cards[22]) && (queue\_cards[card3] == queue\_cards[6]||queue\_cards[card3] == queue\_cards[10]||queue\_cards[card3] == queue\_cards[14]||queue\_cards[card3] == queue\_cards[18]||queue\_cards[card3] == queue\_cards[22]) && (queue\_cards[card4] == queue\_cards[6]||queue\_cards[card4] == queue\_cards[10]||queue\_cards[card4] == queue\_cards[14]||queue\_cards[card4] == queue\_cards[18]||queue\_cards[card4] == queue\_cards[22]) && (queue\_cards[card5] == queue\_cards[6]||queue\_cards[card5] == queue\_cards[10]||queue\_cards[card5] == queue\_cards[14]||queue\_cards[card5] == queue\_cards[18]||queue\_cards[card5] == queue\_cards[22]))

{

\*check\_straight\_flush=2;

}

if((queue\_cards[card1] == queue\_cards[7]||queue\_cards[card1] == queue\_cards[11]||queue\_cards[card1] == queue\_cards[15]||queue\_cards[card1] == queue\_cards[19]||queue\_cards[card1] == queue\_cards[23]) && (queue\_cards[card2] == queue\_cards[7]||queue\_cards[card2] == queue\_cards[11]||queue\_cards[card2] == queue\_cards[15]||queue\_cards[card2] == queue\_cards[19]||queue\_cards[card2] == queue\_cards[23]) && (queue\_cards[card3] == queue\_cards[7]||queue\_cards[card3] == queue\_cards[11]||queue\_cards[card3] == queue\_cards[15]||queue\_cards[card3] == queue\_cards[19]||queue\_cards[card3] == queue\_cards[23]) && (queue\_cards[card4] == queue\_cards[7]||queue\_cards[card4] == queue\_cards[11]||queue\_cards[card4] == queue\_cards[15]||queue\_cards[card4] == queue\_cards[19]||queue\_cards[card4] == queue\_cards[23]) && (queue\_cards[card5] == queue\_cards[7]||queue\_cards[card5] == queue\_cards[11]||queue\_cards[card5] == queue\_cards[15]||queue\_cards[card5] == queue\_cards[19]||queue\_cards[card5] == queue\_cards[23]))

{

\*check\_straight\_flush=2;

}

if((queue\_cards[card1] == queue\_cards[8]||queue\_cards[card1] == queue\_cards[12]||queue\_cards[card1] == queue\_cards[16]||queue\_cards[card1] == queue\_cards[20]||queue\_cards[card1] == queue\_cards[24]) && (queue\_cards[card2] == queue\_cards[8]||queue\_cards[card2] == queue\_cards[12]||queue\_cards[card2] == queue\_cards[16]||queue\_cards[card2] == queue\_cards[20]||queue\_cards[card2] == queue\_cards[24]) && (queue\_cards[card3] == queue\_cards[8]||queue\_cards[card3] == queue\_cards[12]||queue\_cards[card3] == queue\_cards[16]||queue\_cards[card3] == queue\_cards[20]||queue\_cards[card3] == queue\_cards[24]) && (queue\_cards[card4] == queue\_cards[8]||queue\_cards[card4] == queue\_cards[12]||queue\_cards[card4] == queue\_cards[16]||queue\_cards[card4] == queue\_cards[20]||queue\_cards[card4] == queue\_cards[24]) && (queue\_cards[card5] == queue\_cards[8]||queue\_cards[card5] == queue\_cards[12]||queue\_cards[card5] == queue\_cards[16]||queue\_cards[card5] == queue\_cards[20]||queue\_cards[card5] == queue\_cards[24]))

{

\*check\_straight\_flush=3;

}

if((queue\_cards[card1] == queue\_cards[9]||queue\_cards[card1] == queue\_cards[13]||queue\_cards[card1] == queue\_cards[17]||queue\_cards[card1] == queue\_cards[21]||queue\_cards[card1] == queue\_cards[25]) && (queue\_cards[card2] == queue\_cards[9]||queue\_cards[card2] == queue\_cards[13]||queue\_cards[card2] == queue\_cards[17]||queue\_cards[card2] == queue\_cards[21]||queue\_cards[card2] == queue\_cards[25]) && (queue\_cards[card3] == queue\_cards[9]||queue\_cards[card3] == queue\_cards[13]||queue\_cards[card3] == queue\_cards[17]||queue\_cards[card3] == queue\_cards[21]||queue\_cards[card3] == queue\_cards[25]) && (queue\_cards[card4] == queue\_cards[9]||queue\_cards[card4] == queue\_cards[13]||queue\_cards[card4] == queue\_cards[17]||queue\_cards[card4] == queue\_cards[21]||queue\_cards[card4] == queue\_cards[25]) && (queue\_cards[card5] == queue\_cards[9]||queue\_cards[card5] == queue\_cards[13]||queue\_cards[card5] == queue\_cards[17]||queue\_cards[card5] == queue\_cards[21]||queue\_cards[card5] == queue\_cards[25]))

{

\*check\_straight\_flush=3;

}

if((queue\_cards[card1] == queue\_cards[10]||queue\_cards[card1] == queue\_cards[14]||queue\_cards[card1] == queue\_cards[18]||queue\_cards[card1] == queue\_cards[22]||queue\_cards[card1] == queue\_cards[26]) && (queue\_cards[card2] == queue\_cards[10]||queue\_cards[card2] == queue\_cards[14]||queue\_cards[card2] == queue\_cards[18]||queue\_cards[card2] == queue\_cards[22]||queue\_cards[card2] == queue\_cards[26]) && (queue\_cards[card3] == queue\_cards[10]||queue\_cards[card3] == queue\_cards[14]||queue\_cards[card3] == queue\_cards[18]||queue\_cards[card3] == queue\_cards[22]||queue\_cards[card3] == queue\_cards[26]) && (queue\_cards[card4] == queue\_cards[10]||queue\_cards[card4] == queue\_cards[14]||queue\_cards[card4] == queue\_cards[18]||queue\_cards[card4] == queue\_cards[22]||queue\_cards[card4] == queue\_cards[26]) && (queue\_cards[card5] == queue\_cards[10]||queue\_cards[card5] == queue\_cards[14]||queue\_cards[card5] == queue\_cards[18]||queue\_cards[card5] == queue\_cards[22]||queue\_cards[card5] == queue\_cards[26]))

{

\*check\_straight\_flush=3;

}

if((queue\_cards[card1] == queue\_cards[11]||queue\_cards[card1] == queue\_cards[15]||queue\_cards[card1] == queue\_cards[19]||queue\_cards[card1] == queue\_cards[23]||queue\_cards[card1] == queue\_cards[27]) && (queue\_cards[card2] == queue\_cards[11]||queue\_cards[card2] == queue\_cards[15]||queue\_cards[card2] == queue\_cards[19]||queue\_cards[card2] == queue\_cards[23]||queue\_cards[card2] == queue\_cards[27]) && (queue\_cards[card3] == queue\_cards[11]||queue\_cards[card3] == queue\_cards[15]||queue\_cards[card3] == queue\_cards[19]||queue\_cards[card3] == queue\_cards[23]||queue\_cards[card3] == queue\_cards[27]) && (queue\_cards[card4] == queue\_cards[11]||queue\_cards[card4] == queue\_cards[15]||queue\_cards[card4] == queue\_cards[19]||queue\_cards[card4] == queue\_cards[23]||queue\_cards[card4] == queue\_cards[27]) && (queue\_cards[card5] == queue\_cards[11]||queue\_cards[card5] == queue\_cards[15]||queue\_cards[card5] == queue\_cards[19]||queue\_cards[card5] == queue\_cards[23]||queue\_cards[card5] == queue\_cards[27]))

{

\*check\_straight\_flush=3;

}

if((queue\_cards[card1] == queue\_cards[12]||queue\_cards[card1] == queue\_cards[16]||queue\_cards[card1] == queue\_cards[20]||queue\_cards[card1] == queue\_cards[24]||queue\_cards[card1] == queue\_cards[28]) && (queue\_cards[card2] == queue\_cards[12]||queue\_cards[card2] == queue\_cards[16]||queue\_cards[card2] == queue\_cards[20]||queue\_cards[card2] == queue\_cards[24]||queue\_cards[card2] == queue\_cards[28]) && (queue\_cards[card3] == queue\_cards[12]||queue\_cards[card3] == queue\_cards[16]||queue\_cards[card3] == queue\_cards[20]||queue\_cards[card3] == queue\_cards[24]||queue\_cards[card3] == queue\_cards[28]) && (queue\_cards[card4] == queue\_cards[12]||queue\_cards[card4] == queue\_cards[16]||queue\_cards[card4] == queue\_cards[20]||queue\_cards[card4] == queue\_cards[24]||queue\_cards[card4] == queue\_cards[28]) && (queue\_cards[card5] == queue\_cards[12]||queue\_cards[card5] == queue\_cards[16]||queue\_cards[card5] == queue\_cards[20]||queue\_cards[card5] == queue\_cards[24]||queue\_cards[card5] == queue\_cards[28]))

{

\*check\_straight\_flush=4;

}

if((queue\_cards[card1] == queue\_cards[13]||queue\_cards[card1] == queue\_cards[17]||queue\_cards[card1] == queue\_cards[21]||queue\_cards[card1] == queue\_cards[25]||queue\_cards[card1] == queue\_cards[29]) && (queue\_cards[card2] == queue\_cards[13]||queue\_cards[card2] == queue\_cards[17]||queue\_cards[card2] == queue\_cards[21]||queue\_cards[card2] == queue\_cards[25]||queue\_cards[card2] == queue\_cards[29]) && (queue\_cards[card3] == queue\_cards[13]||queue\_cards[card3] == queue\_cards[17]||queue\_cards[card3] == queue\_cards[21]||queue\_cards[card3] == queue\_cards[25]||queue\_cards[card3] == queue\_cards[29]) && (queue\_cards[card4] == queue\_cards[13]||queue\_cards[card4] == queue\_cards[17]||queue\_cards[card4] == queue\_cards[21]||queue\_cards[card4] == queue\_cards[25]||queue\_cards[card4] == queue\_cards[29]) && (queue\_cards[card5] == queue\_cards[13]||queue\_cards[card5] == queue\_cards[17]||queue\_cards[card5] == queue\_cards[21]||queue\_cards[card5] == queue\_cards[25]||queue\_cards[card5] == queue\_cards[29]))

{

\*check\_straight\_flush=4;

}

if((queue\_cards[card1] == queue\_cards[14]||queue\_cards[card1] == queue\_cards[18]||queue\_cards[card1] == queue\_cards[22]||queue\_cards[card1] == queue\_cards[26]||queue\_cards[card1] == queue\_cards[30]) && (queue\_cards[card2] == queue\_cards[14]||queue\_cards[card2] == queue\_cards[18]||queue\_cards[card2] == queue\_cards[22]||queue\_cards[card2] == queue\_cards[26]||queue\_cards[card2] == queue\_cards[30]) && (queue\_cards[card3] == queue\_cards[14]||queue\_cards[card3] == queue\_cards[18]||queue\_cards[card3] == queue\_cards[22]||queue\_cards[card3] == queue\_cards[26]||queue\_cards[card3] == queue\_cards[30]) && (queue\_cards[card4] == queue\_cards[14]||queue\_cards[card4] == queue\_cards[18]||queue\_cards[card4] == queue\_cards[22]||queue\_cards[card4] == queue\_cards[26]||queue\_cards[card4] == queue\_cards[30]) && (queue\_cards[card5] == queue\_cards[14]||queue\_cards[card5] == queue\_cards[18]||queue\_cards[card5] == queue\_cards[22]||queue\_cards[card5] == queue\_cards[26]||queue\_cards[card5] == queue\_cards[30]))

{

\*check\_straight\_flush=4;

}

if((queue\_cards[card1] == queue\_cards[15]||queue\_cards[card1] == queue\_cards[19]||queue\_cards[card1] == queue\_cards[23]||queue\_cards[card1] == queue\_cards[27]||queue\_cards[card1] == queue\_cards[31]) && (queue\_cards[card2] == queue\_cards[15]||queue\_cards[card2] == queue\_cards[19]||queue\_cards[card2] == queue\_cards[23]||queue\_cards[card2] == queue\_cards[27]||queue\_cards[card2] == queue\_cards[31]) && (queue\_cards[card3] == queue\_cards[15]||queue\_cards[card3] == queue\_cards[19]||queue\_cards[card3] == queue\_cards[23]||queue\_cards[card3] == queue\_cards[27]||queue\_cards[card3] == queue\_cards[31]) && (queue\_cards[card4] == queue\_cards[15]||queue\_cards[card4] == queue\_cards[19]||queue\_cards[card4] == queue\_cards[23]||queue\_cards[card4] == queue\_cards[27]||queue\_cards[card4] == queue\_cards[31]) && (queue\_cards[card5] == queue\_cards[15]||queue\_cards[card5] == queue\_cards[19]||queue\_cards[card5] == queue\_cards[23]||queue\_cards[card5] == queue\_cards[27]||queue\_cards[card5] == queue\_cards[31]))

{

\*check\_straight\_flush=4;

}

if((queue\_cards[card1] == queue\_cards[16]||queue\_cards[card1] == queue\_cards[20]||queue\_cards[card1] == queue\_cards[24]||queue\_cards[card1] == queue\_cards[28]||queue\_cards[card1] == queue\_cards[32]) && (queue\_cards[card2] == queue\_cards[16]||queue\_cards[card2] == queue\_cards[20]||queue\_cards[card2] == queue\_cards[24]||queue\_cards[card2] == queue\_cards[28]||queue\_cards[card2] == queue\_cards[32]) && (queue\_cards[card3] == queue\_cards[16]||queue\_cards[card3] == queue\_cards[20]||queue\_cards[card3] == queue\_cards[24]||queue\_cards[card3] == queue\_cards[28]||queue\_cards[card3] == queue\_cards[32]) && (queue\_cards[card4] == queue\_cards[16]||queue\_cards[card4] == queue\_cards[20]||queue\_cards[card4] == queue\_cards[24]||queue\_cards[card4] == queue\_cards[28]||queue\_cards[card4] == queue\_cards[32]) && (queue\_cards[card5] == queue\_cards[16]||queue\_cards[card5] == queue\_cards[20]||queue\_cards[card5] == queue\_cards[24]||queue\_cards[card5] == queue\_cards[28]||queue\_cards[card5] == queue\_cards[32]))

{

\*check\_straight\_flush=5;

}

if((queue\_cards[card1] == queue\_cards[17]||queue\_cards[card1] == queue\_cards[21]||queue\_cards[card1] == queue\_cards[25]||queue\_cards[card1] == queue\_cards[29]||queue\_cards[card1] == queue\_cards[33]) && (queue\_cards[card2] == queue\_cards[17]||queue\_cards[card2] == queue\_cards[21]||queue\_cards[card2] == queue\_cards[25]||queue\_cards[card2] == queue\_cards[29]||queue\_cards[card2] == queue\_cards[33]) && (queue\_cards[card3] == queue\_cards[17]||queue\_cards[card3] == queue\_cards[21]||queue\_cards[card3] == queue\_cards[25]||queue\_cards[card3] == queue\_cards[29]||queue\_cards[card3] == queue\_cards[33]) && (queue\_cards[card4] == queue\_cards[17]||queue\_cards[card4] == queue\_cards[21]||queue\_cards[card4] == queue\_cards[25]||queue\_cards[card4] == queue\_cards[29]||queue\_cards[card4] == queue\_cards[33]) && (queue\_cards[card5] == queue\_cards[17]||queue\_cards[card5] == queue\_cards[21]||queue\_cards[card5] == queue\_cards[25]||queue\_cards[card5] == queue\_cards[29]||queue\_cards[card5] == queue\_cards[33]))

{

\*check\_straight\_flush=5;

}

if((queue\_cards[card1] == queue\_cards[18]||queue\_cards[card1] == queue\_cards[22]||queue\_cards[card1] == queue\_cards[26]||queue\_cards[card1] == queue\_cards[30]||queue\_cards[card1] == queue\_cards[34]) && (queue\_cards[card2] == queue\_cards[18]||queue\_cards[card2] == queue\_cards[22]||queue\_cards[card2] == queue\_cards[26]||queue\_cards[card2] == queue\_cards[30]||queue\_cards[card2] == queue\_cards[34]) && (queue\_cards[card3] == queue\_cards[18]||queue\_cards[card3] == queue\_cards[22]||queue\_cards[card3] == queue\_cards[26]||queue\_cards[card3] == queue\_cards[30]||queue\_cards[card3] == queue\_cards[34]) && (queue\_cards[card4] == queue\_cards[18]||queue\_cards[card4] == queue\_cards[22]||queue\_cards[card4] == queue\_cards[26]||queue\_cards[card4] == queue\_cards[30]||queue\_cards[card4] == queue\_cards[34]) && (queue\_cards[card5] == queue\_cards[18]||queue\_cards[card5] == queue\_cards[22]||queue\_cards[card5] == queue\_cards[26]||queue\_cards[card5] == queue\_cards[30]||queue\_cards[card5] == queue\_cards[34]))

{

\*check\_straight\_flush=5;

}

if((queue\_cards[card1] == queue\_cards[19]||queue\_cards[card1] == queue\_cards[23]||queue\_cards[card1] == queue\_cards[27]||queue\_cards[card1] == queue\_cards[31]||queue\_cards[card1] == queue\_cards[35]) && (queue\_cards[card2] == queue\_cards[19]||queue\_cards[card2] == queue\_cards[23]||queue\_cards[card2] == queue\_cards[27]||queue\_cards[card2] == queue\_cards[31]||queue\_cards[card2] == queue\_cards[35]) && (queue\_cards[card3] == queue\_cards[19]||queue\_cards[card3] == queue\_cards[23]||queue\_cards[card3] == queue\_cards[27]||queue\_cards[card3] == queue\_cards[31]||queue\_cards[card3] == queue\_cards[35]) && (queue\_cards[card4] == queue\_cards[19]||queue\_cards[card4] == queue\_cards[23]||queue\_cards[card4] == queue\_cards[27]||queue\_cards[card4] == queue\_cards[31]||queue\_cards[card4] == queue\_cards[35]) && (queue\_cards[card5] == queue\_cards[19]||queue\_cards[card5] == queue\_cards[23]||queue\_cards[card5] == queue\_cards[27]||queue\_cards[card5] == queue\_cards[31]||queue\_cards[card5] == queue\_cards[35]))

{

\*check\_straight\_flush=5;

}

if((queue\_cards[card1] == queue\_cards[20]||queue\_cards[card1] == queue\_cards[24]||queue\_cards[card1] == queue\_cards[28]||queue\_cards[card1] == queue\_cards[32]||queue\_cards[card1] == queue\_cards[36]) && (queue\_cards[card2] == queue\_cards[20]||queue\_cards[card2] == queue\_cards[24]||queue\_cards[card2] == queue\_cards[28]||queue\_cards[card2] == queue\_cards[32]||queue\_cards[card2] == queue\_cards[36]) && (queue\_cards[card3] == queue\_cards[20]||queue\_cards[card3] == queue\_cards[24]||queue\_cards[card3] == queue\_cards[28]||queue\_cards[card3] == queue\_cards[32]||queue\_cards[card3] == queue\_cards[36]) && (queue\_cards[card4] == queue\_cards[20]||queue\_cards[card4] == queue\_cards[24]||queue\_cards[card4] == queue\_cards[28]||queue\_cards[card4] == queue\_cards[32]||queue\_cards[card4] == queue\_cards[36]) && (queue\_cards[card5] == queue\_cards[20]||queue\_cards[card5] == queue\_cards[24]||queue\_cards[card5] == queue\_cards[28]||queue\_cards[card5] == queue\_cards[32]||queue\_cards[card5] == queue\_cards[36]))

{

\*check\_straight\_flush=6;

}

if((queue\_cards[card1] == queue\_cards[21]||queue\_cards[card1] == queue\_cards[25]||queue\_cards[card1] == queue\_cards[29]||queue\_cards[card1] == queue\_cards[33]||queue\_cards[card1] == queue\_cards[37]) && (queue\_cards[card2] == queue\_cards[21]||queue\_cards[card2] == queue\_cards[25]||queue\_cards[card2] == queue\_cards[29]||queue\_cards[card2] == queue\_cards[33]||queue\_cards[card2] == queue\_cards[37]) && (queue\_cards[card3] == queue\_cards[21]||queue\_cards[card3] == queue\_cards[25]||queue\_cards[card3] == queue\_cards[29]||queue\_cards[card3] == queue\_cards[33]||queue\_cards[card3] == queue\_cards[37]) && (queue\_cards[card4] == queue\_cards[21]||queue\_cards[card4] == queue\_cards[25]||queue\_cards[card4] == queue\_cards[29]||queue\_cards[card4] == queue\_cards[33]||queue\_cards[card4] == queue\_cards[37]) && (queue\_cards[card5] == queue\_cards[21]||queue\_cards[card5] == queue\_cards[25]||queue\_cards[card5] == queue\_cards[29]||queue\_cards[card5] == queue\_cards[33]||queue\_cards[card5] == queue\_cards[37]))

{

\*check\_straight\_flush=6;

}

if((queue\_cards[card1] == queue\_cards[22]||queue\_cards[card1] == queue\_cards[26]||queue\_cards[card1] == queue\_cards[30]||queue\_cards[card1] == queue\_cards[34]||queue\_cards[card1] == queue\_cards[38]) && (queue\_cards[card2] == queue\_cards[22]||queue\_cards[card2] == queue\_cards[26]||queue\_cards[card2] == queue\_cards[30]||queue\_cards[card2] == queue\_cards[34]||queue\_cards[card2] == queue\_cards[38]) && (queue\_cards[card3] == queue\_cards[22]||queue\_cards[card3] == queue\_cards[26]||queue\_cards[card3] == queue\_cards[30]||queue\_cards[card3] == queue\_cards[34]||queue\_cards[card3] == queue\_cards[38]) && (queue\_cards[card4] == queue\_cards[22]||queue\_cards[card4] == queue\_cards[26]||queue\_cards[card4] == queue\_cards[30]||queue\_cards[card4] == queue\_cards[34]||queue\_cards[card4] == queue\_cards[38]) && (queue\_cards[card5] == queue\_cards[22]||queue\_cards[card5] == queue\_cards[26]||queue\_cards[card5] == queue\_cards[30]||queue\_cards[card5] == queue\_cards[34]||queue\_cards[card5] == queue\_cards[38]))

{

\*check\_straight\_flush=6;

}

if((queue\_cards[card1] == queue\_cards[23]||queue\_cards[card1] == queue\_cards[27]||queue\_cards[card1] == queue\_cards[31]||queue\_cards[card1] == queue\_cards[35]||queue\_cards[card1] == queue\_cards[39]) && (queue\_cards[card2] == queue\_cards[23]||queue\_cards[card2] == queue\_cards[27]||queue\_cards[card2] == queue\_cards[31]||queue\_cards[card2] == queue\_cards[35]||queue\_cards[card2] == queue\_cards[39]) && (queue\_cards[card3] == queue\_cards[23]||queue\_cards[card3] == queue\_cards[27]||queue\_cards[card3] == queue\_cards[31]||queue\_cards[card3] == queue\_cards[35]||queue\_cards[card3] == queue\_cards[39]) && (queue\_cards[card4] == queue\_cards[23]||queue\_cards[card4] == queue\_cards[27]||queue\_cards[card4] == queue\_cards[31]||queue\_cards[card4] == queue\_cards[35]||queue\_cards[card4] == queue\_cards[39]) && (queue\_cards[card5] == queue\_cards[23]||queue\_cards[card5] == queue\_cards[27]||queue\_cards[card5] == queue\_cards[31]||queue\_cards[card5] == queue\_cards[35]||queue\_cards[card5] == queue\_cards[39]))

{

\*check\_straight\_flush=6;

}

if((queue\_cards[card1] == queue\_cards[24]||queue\_cards[card1] == queue\_cards[28]||queue\_cards[card1] == queue\_cards[32]||queue\_cards[card1] == queue\_cards[36]||queue\_cards[card1] == queue\_cards[40]) && (queue\_cards[card2] == queue\_cards[24]||queue\_cards[card2] == queue\_cards[28]||queue\_cards[card2] == queue\_cards[32]||queue\_cards[card2] == queue\_cards[36]||queue\_cards[card2] == queue\_cards[40]) && (queue\_cards[card3] == queue\_cards[24]||queue\_cards[card3] == queue\_cards[28]||queue\_cards[card3] == queue\_cards[32]||queue\_cards[card3] == queue\_cards[36]||queue\_cards[card3] == queue\_cards[40]) && (queue\_cards[card4] == queue\_cards[24]||queue\_cards[card4] == queue\_cards[28]||queue\_cards[card4] == queue\_cards[32]||queue\_cards[card4] == queue\_cards[36]||queue\_cards[card4] == queue\_cards[40]) && (queue\_cards[card5] == queue\_cards[24]||queue\_cards[card5] == queue\_cards[28]||queue\_cards[card5] == queue\_cards[32]||queue\_cards[card5] == queue\_cards[36]||queue\_cards[card5] == queue\_cards[40]))

{

\*check\_straight\_flush=7;

}

if((queue\_cards[card1] == queue\_cards[25]||queue\_cards[card1] == queue\_cards[29]||queue\_cards[card1] == queue\_cards[33]||queue\_cards[card1] == queue\_cards[37]||queue\_cards[card1] == queue\_cards[41]) && (queue\_cards[card2] == queue\_cards[25]||queue\_cards[card2] == queue\_cards[29]||queue\_cards[card2] == queue\_cards[33]||queue\_cards[card2] == queue\_cards[37]||queue\_cards[card2] == queue\_cards[41]) && (queue\_cards[card3] == queue\_cards[25]||queue\_cards[card3] == queue\_cards[29]||queue\_cards[card3] == queue\_cards[33]||queue\_cards[card3] == queue\_cards[37]||queue\_cards[card3] == queue\_cards[41]) && (queue\_cards[card4] == queue\_cards[25]||queue\_cards[card4] == queue\_cards[29]||queue\_cards[card4] == queue\_cards[33]||queue\_cards[card4] == queue\_cards[37]||queue\_cards[card4] == queue\_cards[41]) && (queue\_cards[card5] == queue\_cards[25]||queue\_cards[card5] == queue\_cards[29]||queue\_cards[card5] == queue\_cards[33]||queue\_cards[card5] == queue\_cards[37]||queue\_cards[card5] == queue\_cards[41]))

{

\*check\_straight\_flush=7;

}

if((queue\_cards[card1] == queue\_cards[26]||queue\_cards[card1] == queue\_cards[30]||queue\_cards[card1] == queue\_cards[34]||queue\_cards[card1] == queue\_cards[38]||queue\_cards[card1] == queue\_cards[42]) && (queue\_cards[card2] == queue\_cards[26]||queue\_cards[card2] == queue\_cards[30]||queue\_cards[card2] == queue\_cards[34]||queue\_cards[card2] == queue\_cards[38]||queue\_cards[card2] == queue\_cards[42]) && (queue\_cards[card3] == queue\_cards[26]||queue\_cards[card3] == queue\_cards[30]||queue\_cards[card3] == queue\_cards[34]||queue\_cards[card3] == queue\_cards[38]||queue\_cards[card3] == queue\_cards[42]) && (queue\_cards[card4] == queue\_cards[26]||queue\_cards[card4] == queue\_cards[30]||queue\_cards[card4] == queue\_cards[34]||queue\_cards[card4] == queue\_cards[38]||queue\_cards[card4] == queue\_cards[42]) && (queue\_cards[card5] == queue\_cards[26]||queue\_cards[card5] == queue\_cards[30]||queue\_cards[card5] == queue\_cards[34]||queue\_cards[card5] == queue\_cards[38]||queue\_cards[card5] == queue\_cards[42]))

{

\*check\_straight\_flush=7;

}

if((queue\_cards[card1] == queue\_cards[27]||queue\_cards[card1] == queue\_cards[31]||queue\_cards[card1] == queue\_cards[35]||queue\_cards[card1] == queue\_cards[39]||queue\_cards[card1] == queue\_cards[43]) && (queue\_cards[card2] == queue\_cards[27]||queue\_cards[card2] == queue\_cards[31]||queue\_cards[card2] == queue\_cards[35]||queue\_cards[card2] == queue\_cards[39]||queue\_cards[card2] == queue\_cards[43]) && (queue\_cards[card3] == queue\_cards[27]||queue\_cards[card3] == queue\_cards[31]||queue\_cards[card3] == queue\_cards[35]||queue\_cards[card3] == queue\_cards[39]||queue\_cards[card3] == queue\_cards[43]) && (queue\_cards[card4] == queue\_cards[27]||queue\_cards[card4] == queue\_cards[31]||queue\_cards[card4] == queue\_cards[35]||queue\_cards[card4] == queue\_cards[39]||queue\_cards[card4] == queue\_cards[43]) && (queue\_cards[card5] == queue\_cards[27]||queue\_cards[card5] == queue\_cards[31]||queue\_cards[card5] == queue\_cards[35]||queue\_cards[card5] == queue\_cards[39]||queue\_cards[card5] == queue\_cards[43]))

{

\*check\_straight\_flush=7;

}

if((queue\_cards[card1] == queue\_cards[28]||queue\_cards[card1] == queue\_cards[32]||queue\_cards[card1] == queue\_cards[36]||queue\_cards[card1] == queue\_cards[40]||queue\_cards[card1] == queue\_cards[44]) && (queue\_cards[card2] == queue\_cards[28]||queue\_cards[card2] == queue\_cards[32]||queue\_cards[card2] == queue\_cards[36]||queue\_cards[card2] == queue\_cards[40]||queue\_cards[card2] == queue\_cards[44]) && (queue\_cards[card3] == queue\_cards[28]||queue\_cards[card3] == queue\_cards[32]||queue\_cards[card3] == queue\_cards[36]||queue\_cards[card3] == queue\_cards[40]||queue\_cards[card3] == queue\_cards[44]) && (queue\_cards[card4] == queue\_cards[28]||queue\_cards[card4] == queue\_cards[32]||queue\_cards[card4] == queue\_cards[36]||queue\_cards[card4] == queue\_cards[40]||queue\_cards[card4] == queue\_cards[44]) && (queue\_cards[card5] == queue\_cards[28]||queue\_cards[card5] == queue\_cards[32]||queue\_cards[card5] == queue\_cards[36]||queue\_cards[card5] == queue\_cards[40]||queue\_cards[card5] == queue\_cards[44]))

{

\*check\_straight\_flush=8;

}

if((queue\_cards[card1] == queue\_cards[29]||queue\_cards[card1] == queue\_cards[33]||queue\_cards[card1] == queue\_cards[37]||queue\_cards[card1] == queue\_cards[41]||queue\_cards[card1] == queue\_cards[45]) && (queue\_cards[card2] == queue\_cards[29]||queue\_cards[card2] == queue\_cards[33]||queue\_cards[card2] == queue\_cards[37]||queue\_cards[card2] == queue\_cards[41]||queue\_cards[card2] == queue\_cards[45]) && (queue\_cards[card3] == queue\_cards[29]||queue\_cards[card3] == queue\_cards[33]||queue\_cards[card3] == queue\_cards[37]||queue\_cards[card3] == queue\_cards[41]||queue\_cards[card3] == queue\_cards[45]) && (queue\_cards[card4] == queue\_cards[29]||queue\_cards[card4] == queue\_cards[33]||queue\_cards[card4] == queue\_cards[37]||queue\_cards[card4] == queue\_cards[41]||queue\_cards[card4] == queue\_cards[45]) && (queue\_cards[card5] == queue\_cards[29]||queue\_cards[card5] == queue\_cards[33]||queue\_cards[card5] == queue\_cards[37]||queue\_cards[card5] == queue\_cards[41]||queue\_cards[card5] == queue\_cards[45]))

{

\*check\_straight\_flush=8;

}

if((queue\_cards[card1] == queue\_cards[30]||queue\_cards[card1] == queue\_cards[34]||queue\_cards[card1] == queue\_cards[38]||queue\_cards[card1] == queue\_cards[42]||queue\_cards[card1] == queue\_cards[46]) && (queue\_cards[card2] == queue\_cards[30]||queue\_cards[card2] == queue\_cards[34]||queue\_cards[card2] == queue\_cards[38]||queue\_cards[card2] == queue\_cards[42]||queue\_cards[card2] == queue\_cards[46]) && (queue\_cards[card3] == queue\_cards[30]||queue\_cards[card3] == queue\_cards[34]||queue\_cards[card3] == queue\_cards[38]||queue\_cards[card3] == queue\_cards[42]||queue\_cards[card3] == queue\_cards[46]) && (queue\_cards[card4] == queue\_cards[30]||queue\_cards[card4] == queue\_cards[34]||queue\_cards[card4] == queue\_cards[38]||queue\_cards[card4] == queue\_cards[42]||queue\_cards[card4] == queue\_cards[46]) && (queue\_cards[card5] == queue\_cards[30]||queue\_cards[card5] == queue\_cards[34]||queue\_cards[card5] == queue\_cards[38]||queue\_cards[card5] == queue\_cards[42]||queue\_cards[card5] == queue\_cards[46]))

{

\*check\_straight\_flush=8;

}

if((queue\_cards[card1] == queue\_cards[31]||queue\_cards[card1] == queue\_cards[35]||queue\_cards[card1] == queue\_cards[39]||queue\_cards[card1] == queue\_cards[43]||queue\_cards[card1] == queue\_cards[47]) && (queue\_cards[card2] == queue\_cards[31]||queue\_cards[card2] == queue\_cards[35]||queue\_cards[card2] == queue\_cards[39]||queue\_cards[card2] == queue\_cards[43]||queue\_cards[card2] == queue\_cards[47]) && (queue\_cards[card3] == queue\_cards[31]||queue\_cards[card3] == queue\_cards[35]||queue\_cards[card3] == queue\_cards[39]||queue\_cards[card3] == queue\_cards[43]||queue\_cards[card3] == queue\_cards[47]) && (queue\_cards[card4] == queue\_cards[31]||queue\_cards[card4] == queue\_cards[35]||queue\_cards[card4] == queue\_cards[39]||queue\_cards[card4] == queue\_cards[43]||queue\_cards[card4] == queue\_cards[47]) && (queue\_cards[card5] == queue\_cards[31]||queue\_cards[card5] == queue\_cards[35]||queue\_cards[card5] == queue\_cards[39]||queue\_cards[card5] == queue\_cards[43]||queue\_cards[card5] == queue\_cards[47]))

{

\*check\_straight\_flush=8;

}

if((queue\_cards[card1] == queue\_cards[32]||queue\_cards[card1] == queue\_cards[36]||queue\_cards[card1] == queue\_cards[40]||queue\_cards[card1] == queue\_cards[44]||queue\_cards[card1] == queue\_cards[48]) && (queue\_cards[card2] == queue\_cards[32]||queue\_cards[card2] == queue\_cards[36]||queue\_cards[card2] == queue\_cards[40]||queue\_cards[card2] == queue\_cards[44]||queue\_cards[card2] == queue\_cards[48]) && (queue\_cards[card3] == queue\_cards[32]||queue\_cards[card3] == queue\_cards[36]||queue\_cards[card3] == queue\_cards[40]||queue\_cards[card3] == queue\_cards[44]||queue\_cards[card3] == queue\_cards[48]) && (queue\_cards[card4] == queue\_cards[32]||queue\_cards[card4] == queue\_cards[36]||queue\_cards[card4] == queue\_cards[40]||queue\_cards[card4] == queue\_cards[44]||queue\_cards[card4] == queue\_cards[48]) && (queue\_cards[card5] == queue\_cards[32]||queue\_cards[card5] == queue\_cards[36]||queue\_cards[card5] == queue\_cards[40]||queue\_cards[card5] == queue\_cards[44]||queue\_cards[card5] == queue\_cards[48]))

{

\*check\_flush\_royal=1;

}

if((queue\_cards[card1] == queue\_cards[33]||queue\_cards[card1] == queue\_cards[37]||queue\_cards[card1] == queue\_cards[41]||queue\_cards[card1] == queue\_cards[45]||queue\_cards[card1] == queue\_cards[49]) && (queue\_cards[card2] == queue\_cards[33]||queue\_cards[card2] == queue\_cards[37]||queue\_cards[card2] == queue\_cards[41]||queue\_cards[card2] == queue\_cards[45]||queue\_cards[card2] == queue\_cards[49]) && (queue\_cards[card3] == queue\_cards[33]||queue\_cards[card3] == queue\_cards[37]||queue\_cards[card3] == queue\_cards[41]||queue\_cards[card3] == queue\_cards[45]||queue\_cards[card3] == queue\_cards[49]) && (queue\_cards[card4] == queue\_cards[33]||queue\_cards[card4] == queue\_cards[37]||queue\_cards[card4] == queue\_cards[41]||queue\_cards[card4] == queue\_cards[45]||queue\_cards[card4] == queue\_cards[49]) && (queue\_cards[card5] == queue\_cards[33]||queue\_cards[card5] == queue\_cards[37]||queue\_cards[card5] == queue\_cards[41]||queue\_cards[card5] == queue\_cards[45]||queue\_cards[card5] == queue\_cards[49]))

{

\*check\_flush\_royal=1;

}

if((queue\_cards[card1] == queue\_cards[34]||queue\_cards[card1] == queue\_cards[38]||queue\_cards[card1] == queue\_cards[42]||queue\_cards[card1] == queue\_cards[46]||queue\_cards[card1] == queue\_cards[50]) && (queue\_cards[card2] == queue\_cards[34]||queue\_cards[card2] == queue\_cards[38]||queue\_cards[card2] == queue\_cards[42]||queue\_cards[card2] == queue\_cards[46]||queue\_cards[card2] == queue\_cards[50]) && (queue\_cards[card3] == queue\_cards[34]||queue\_cards[card3] == queue\_cards[38]||queue\_cards[card3] == queue\_cards[42]||queue\_cards[card3] == queue\_cards[46]||queue\_cards[card3] == queue\_cards[50]) && (queue\_cards[card4] == queue\_cards[34]||queue\_cards[card4] == queue\_cards[38]||queue\_cards[card4] == queue\_cards[42]||queue\_cards[card4] == queue\_cards[46]||queue\_cards[card4] == queue\_cards[50]) && (queue\_cards[card5] == queue\_cards[34]||queue\_cards[card5] == queue\_cards[38]||queue\_cards[card5] == queue\_cards[42]||queue\_cards[card5] == queue\_cards[46]||queue\_cards[card5] == queue\_cards[50]))

{

\*check\_flush\_royal=1;

}

if((queue\_cards[card1] == queue\_cards[35]||queue\_cards[card1] == queue\_cards[39]||queue\_cards[card1] == queue\_cards[43]||queue\_cards[card1] == queue\_cards[47]||queue\_cards[card1] == queue\_cards[51]) && (queue\_cards[card2] == queue\_cards[35]||queue\_cards[card2] == queue\_cards[39]||queue\_cards[card2] == queue\_cards[43]||queue\_cards[card2] == queue\_cards[47]||queue\_cards[card2] == queue\_cards[51]) && (queue\_cards[card3] == queue\_cards[35]||queue\_cards[card3] == queue\_cards[39]||queue\_cards[card3] == queue\_cards[43]||queue\_cards[card3] == queue\_cards[47]||queue\_cards[card3] == queue\_cards[51]) && (queue\_cards[card4] == queue\_cards[35]||queue\_cards[card4] == queue\_cards[39]||queue\_cards[card4] == queue\_cards[43]||queue\_cards[card4] == queue\_cards[47]||queue\_cards[card4] == queue\_cards[51]) && (queue\_cards[card5] == queue\_cards[35]||queue\_cards[card5] == queue\_cards[39]||queue\_cards[card5] == queue\_cards[43]||queue\_cards[card5] == queue\_cards[47]||queue\_cards[card5] == queue\_cards[51]))

{

\*check\_flush\_royal=1;

}

}

//--------------CARDS--------------------

void CApp::combinations\_straight(int card1, int card2, int card3, int card4, int card5, int NS, int NS1, int temp, int \*check\_straigh)

{

//----------------STRAIGHT-2-------------

if(queue\_cards[card1] == queue\_cards[NS+0]||queue\_cards[card1] == queue\_cards[NS+1]||queue\_cards[card1] == queue\_cards[NS+2]||queue\_cards[card1] == queue\_cards[NS+3])

{

if(queue\_cards[card2] == queue\_cards[NS+4]||queue\_cards[card2] == queue\_cards[NS+5]||queue\_cards[card2] == queue\_cards[NS+6]||queue\_cards[card2] == queue\_cards[NS+7])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS+8]||queue\_cards[card2] == queue\_cards[NS+9]||queue\_cards[card2] == queue\_cards[NS+10]||queue\_cards[card2] == queue\_cards[NS+11])

{

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS+12]||queue\_cards[card2] == queue\_cards[NS+13]||queue\_cards[card2] == queue\_cards[NS+14]||queue\_cards[card2] == queue\_cards[NS+15])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS1+NS+16]||queue\_cards[card2] == queue\_cards[NS1+NS+17]||queue\_cards[card2] == queue\_cards[NS1+NS+18]||queue\_cards[card2] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

}

if(queue\_cards[card1] == queue\_cards[NS+4]||queue\_cards[card1] == queue\_cards[NS+5]||queue\_cards[card1] == queue\_cards[NS+6]||queue\_cards[card1] == queue\_cards[NS+7])

{

if(queue\_cards[card2] == queue\_cards[NS+0]||queue\_cards[card2] == queue\_cards[NS+1]||queue\_cards[card2] == queue\_cards[NS+2]||queue\_cards[card2] == queue\_cards[NS+3])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS+8]||queue\_cards[card2] == queue\_cards[NS+9]||queue\_cards[card2] == queue\_cards[NS+10]||queue\_cards[card2] == queue\_cards[NS+11])

{

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS+12]||queue\_cards[card2] == queue\_cards[NS+13]||queue\_cards[card2] == queue\_cards[NS+14]||queue\_cards[card2] == queue\_cards[NS+15])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS1+NS+16]||queue\_cards[card2] == queue\_cards[NS1+NS+17]||queue\_cards[card2] == queue\_cards[NS1+NS+18]||queue\_cards[card2] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

}

if(queue\_cards[card1] == queue\_cards[NS+8]||queue\_cards[card1] == queue\_cards[NS+9]||queue\_cards[card1] == queue\_cards[NS+10]||queue\_cards[card1] == queue\_cards[NS+11])

{

if(queue\_cards[card2] == queue\_cards[NS+4]||queue\_cards[card2] == queue\_cards[NS+5]||queue\_cards[card2] == queue\_cards[NS+6]||queue\_cards[card2] == queue\_cards[NS+7])

{

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS+0]||queue\_cards[card2] == queue\_cards[NS+1]||queue\_cards[card2] == queue\_cards[NS+2]||queue\_cards[card2] == queue\_cards[NS+3])

{

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS+12]||queue\_cards[card2] == queue\_cards[NS+13]||queue\_cards[card2] == queue\_cards[NS+14]||queue\_cards[card2] == queue\_cards[NS+15])

{

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS1+NS+16]||queue\_cards[card2] == queue\_cards[NS1+NS+17]||queue\_cards[card2] == queue\_cards[NS1+NS+18]||queue\_cards[card2] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

}

if(queue\_cards[card1] == queue\_cards[NS+12]||queue\_cards[card1] == queue\_cards[NS+13]||queue\_cards[card1] == queue\_cards[NS+14]||queue\_cards[card1] == queue\_cards[NS+15])

{

if(queue\_cards[card2] == queue\_cards[NS+4]||queue\_cards[card2] == queue\_cards[NS+5]||queue\_cards[card2] == queue\_cards[NS+6]||queue\_cards[card2] == queue\_cards[NS+7])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS+8]||queue\_cards[card2] == queue\_cards[NS+9]||queue\_cards[card2] == queue\_cards[NS+10]||queue\_cards[card2] == queue\_cards[NS+11])

{

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS+0]||queue\_cards[card2] == queue\_cards[NS+1]||queue\_cards[card2] == queue\_cards[NS+2]||queue\_cards[card2] == queue\_cards[NS+3])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS1+NS+16]||queue\_cards[card5] == queue\_cards[NS1+NS+17]||queue\_cards[card5] == queue\_cards[NS1+NS+18]||queue\_cards[card5] == queue\_cards[NS1+NS+19])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS1+NS+16]||queue\_cards[card4] == queue\_cards[NS1+NS+17]||queue\_cards[card4] == queue\_cards[NS1+NS+18]||queue\_cards[card4] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS1+NS+16]||queue\_cards[card3] == queue\_cards[NS1+NS+17]||queue\_cards[card3] == queue\_cards[NS1+NS+18]||queue\_cards[card3] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS1+NS+16]||queue\_cards[card2] == queue\_cards[NS1+NS+17]||queue\_cards[card2] == queue\_cards[NS1+NS+18]||queue\_cards[card2] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

}

if(queue\_cards[card1] == queue\_cards[NS1+NS+16]||queue\_cards[card1] == queue\_cards[NS1+NS+17]||queue\_cards[card1] == queue\_cards[NS1+NS+18]||queue\_cards[card1] == queue\_cards[NS1+NS+19])

{

if(queue\_cards[card2] == queue\_cards[NS+4]||queue\_cards[card2] == queue\_cards[NS+5]||queue\_cards[card2] == queue\_cards[NS+6]||queue\_cards[card2] == queue\_cards[NS+7])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS+8]||queue\_cards[card2] == queue\_cards[NS+9]||queue\_cards[card2] == queue\_cards[NS+10]||queue\_cards[card2] == queue\_cards[NS+11])

{

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS+12]||queue\_cards[card2] == queue\_cards[NS+13]||queue\_cards[card2] == queue\_cards[NS+14]||queue\_cards[card2] == queue\_cards[NS+15])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+0]||queue\_cards[card5] == queue\_cards[NS+1]||queue\_cards[card5] == queue\_cards[NS+2]||queue\_cards[card5] == queue\_cards[NS+3])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+0]||queue\_cards[card4] == queue\_cards[NS+1]||queue\_cards[card4] == queue\_cards[NS+2]||queue\_cards[card4] == queue\_cards[NS+3])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+0]||queue\_cards[card3] == queue\_cards[NS+1]||queue\_cards[card3] == queue\_cards[NS+2]||queue\_cards[card3] == queue\_cards[NS+3])

{

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

if(queue\_cards[card2] == queue\_cards[NS+0]||queue\_cards[card2] == queue\_cards[NS+1]||queue\_cards[card2] == queue\_cards[NS+2]||queue\_cards[card2] == queue\_cards[NS+3])

{

if(queue\_cards[card3] == queue\_cards[NS+8]||queue\_cards[card3] == queue\_cards[NS+9]||queue\_cards[card3] == queue\_cards[NS+10]||queue\_cards[card3] == queue\_cards[NS+11])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+12]||queue\_cards[card3] == queue\_cards[NS+13]||queue\_cards[card3] == queue\_cards[NS+14]||queue\_cards[card3] == queue\_cards[NS+15])

{

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+4]||queue\_cards[card5] == queue\_cards[NS+5]||queue\_cards[card5] == queue\_cards[NS+6]||queue\_cards[card5] == queue\_cards[NS+7])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+4]||queue\_cards[card4] == queue\_cards[NS+5]||queue\_cards[card4] == queue\_cards[NS+6]||queue\_cards[card4] == queue\_cards[NS+7])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

if(queue\_cards[card3] == queue\_cards[NS+4]||queue\_cards[card3] == queue\_cards[NS+5]||queue\_cards[card3] == queue\_cards[NS+6]||queue\_cards[card3] == queue\_cards[NS+7])

{

if(queue\_cards[card4] == queue\_cards[NS+12]||queue\_cards[card4] == queue\_cards[NS+13]||queue\_cards[card4] == queue\_cards[NS+14]||queue\_cards[card4] == queue\_cards[NS+15])

{

if(queue\_cards[card5] == queue\_cards[NS+8]||queue\_cards[card5] == queue\_cards[NS+9]||queue\_cards[card5] == queue\_cards[NS+10]||queue\_cards[card5] == queue\_cards[NS+11])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

if(queue\_cards[card4] == queue\_cards[NS+8]||queue\_cards[card4] == queue\_cards[NS+9]||queue\_cards[card4] == queue\_cards[NS+10]||queue\_cards[card4] == queue\_cards[NS+11])

{

if(queue\_cards[card5] == queue\_cards[NS+12]||queue\_cards[card5] == queue\_cards[NS+13]||queue\_cards[card5] == queue\_cards[NS+14]||queue\_cards[card5] == queue\_cards[NS+15])

{

\*check\_straigh=1+temp;

check\_temp\_card\_straight=1;

return;

}

}

}

}

}

//----------------STRAIGHT-2(END)--------

}

//--------------CARDS--------------------

void CApp::combinations\_flash(int card1, int card2, int card3, int card4, int card5, int \*check\_flush)

{

//--------------SPADES-------------------

if(queue\_cards[card1] == queue\_cards[0]||queue\_cards[card1] == queue\_cards[4]||queue\_cards[card1] == queue\_cards[8]||queue\_cards[card1] == queue\_cards[12]||queue\_cards[card1] == queue\_cards[16]||queue\_cards[card1] == queue\_cards[20]||queue\_cards[card1] == queue\_cards[24]||queue\_cards[card1] == queue\_cards[28]||queue\_cards[card1] == queue\_cards[32]||queue\_cards[card1] == queue\_cards[36]||queue\_cards[card1] == queue\_cards[40]||queue\_cards[card1] == queue\_cards[44]||queue\_cards[card1] == queue\_cards[48])

{

if(queue\_cards[card2] == queue\_cards[0]||queue\_cards[card2] == queue\_cards[4]||queue\_cards[card2] == queue\_cards[8]||queue\_cards[card2] == queue\_cards[12]||queue\_cards[card2] == queue\_cards[16]||queue\_cards[card2] == queue\_cards[20]||queue\_cards[card2] == queue\_cards[24]||queue\_cards[card2] == queue\_cards[28]||queue\_cards[card2] == queue\_cards[32]||queue\_cards[card2] == queue\_cards[36]||queue\_cards[card2] == queue\_cards[40]||queue\_cards[card2] == queue\_cards[44]||queue\_cards[card2] == queue\_cards[48])

{

if(queue\_cards[card3] == queue\_cards[0]||queue\_cards[card3] == queue\_cards[4]||queue\_cards[card3] == queue\_cards[8]||queue\_cards[card3] == queue\_cards[12]||queue\_cards[card3] == queue\_cards[16]||queue\_cards[card3] == queue\_cards[20]||queue\_cards[card3] == queue\_cards[24]||queue\_cards[card3] == queue\_cards[28]||queue\_cards[card3] == queue\_cards[32]||queue\_cards[card3] == queue\_cards[36]||queue\_cards[card3] == queue\_cards[40]||queue\_cards[card3] == queue\_cards[44]||queue\_cards[card3] == queue\_cards[48])

{

if(queue\_cards[card4] == queue\_cards[0]||queue\_cards[card4] == queue\_cards[4]||queue\_cards[card4] == queue\_cards[8]||queue\_cards[card4] == queue\_cards[12]||queue\_cards[card4] == queue\_cards[16]||queue\_cards[card4] == queue\_cards[20]||queue\_cards[card4] == queue\_cards[24]||queue\_cards[card4] == queue\_cards[28]||queue\_cards[card4] == queue\_cards[32]||queue\_cards[card4] == queue\_cards[36]||queue\_cards[card4] == queue\_cards[40]||queue\_cards[card4] == queue\_cards[44]||queue\_cards[card4] == queue\_cards[48])

{

if(queue\_cards[card5] == queue\_cards[0]||queue\_cards[card5] == queue\_cards[4]||queue\_cards[card5] == queue\_cards[8]||queue\_cards[card5] == queue\_cards[12]||queue\_cards[card5] == queue\_cards[16]||queue\_cards[card5] == queue\_cards[20]||queue\_cards[card5] == queue\_cards[24]||queue\_cards[card5] == queue\_cards[28]||queue\_cards[card5] == queue\_cards[32]||queue\_cards[card5] == queue\_cards[36]||queue\_cards[card5] == queue\_cards[40]||queue\_cards[card5] == queue\_cards[44]||queue\_cards[card5] == queue\_cards[48])

{

\*check\_flush=1;

}

}

}

}

}

//--------------SPADES(END)--------------

//--------------CLUBS--------------------

if(queue\_cards[card1] == queue\_cards[1]||queue\_cards[card1] == queue\_cards[5]||queue\_cards[card1] == queue\_cards[9]||queue\_cards[card1] == queue\_cards[13]||queue\_cards[card1] == queue\_cards[17]||queue\_cards[card1] == queue\_cards[21]||queue\_cards[card1] == queue\_cards[25]||queue\_cards[card1] == queue\_cards[29]||queue\_cards[card1] == queue\_cards[33]||queue\_cards[card1] == queue\_cards[37]||queue\_cards[card1] == queue\_cards[41]||queue\_cards[card1] == queue\_cards[45]||queue\_cards[card1] == queue\_cards[49])

{

if(queue\_cards[card2] == queue\_cards[1]||queue\_cards[card2] == queue\_cards[5]||queue\_cards[card2] == queue\_cards[9]||queue\_cards[card2] == queue\_cards[13]||queue\_cards[card2] == queue\_cards[17]||queue\_cards[card2] == queue\_cards[21]||queue\_cards[card2] == queue\_cards[25]||queue\_cards[card2] == queue\_cards[29]||queue\_cards[card2] == queue\_cards[33]||queue\_cards[card2] == queue\_cards[37]||queue\_cards[card2] == queue\_cards[41]||queue\_cards[card2] == queue\_cards[45]||queue\_cards[card2] == queue\_cards[49])

{

if(queue\_cards[card3] == queue\_cards[1]||queue\_cards[card3] == queue\_cards[5]||queue\_cards[card3] == queue\_cards[9]||queue\_cards[card3] == queue\_cards[13]||queue\_cards[card3] == queue\_cards[17]||queue\_cards[card3] == queue\_cards[21]||queue\_cards[card3] == queue\_cards[25]||queue\_cards[card3] == queue\_cards[29]||queue\_cards[card3] == queue\_cards[33]||queue\_cards[card3] == queue\_cards[37]||queue\_cards[card3] == queue\_cards[41]||queue\_cards[card3] == queue\_cards[45]||queue\_cards[card3] == queue\_cards[49])

{

if(queue\_cards[card4] == queue\_cards[1]||queue\_cards[card4] == queue\_cards[5]||queue\_cards[card4] == queue\_cards[9]||queue\_cards[card4] == queue\_cards[13]||queue\_cards[card4] == queue\_cards[17]||queue\_cards[card4] == queue\_cards[21]||queue\_cards[card4] == queue\_cards[25]||queue\_cards[card4] == queue\_cards[29]||queue\_cards[card4] == queue\_cards[33]||queue\_cards[card4] == queue\_cards[37]||queue\_cards[card4] == queue\_cards[41]||queue\_cards[card4] == queue\_cards[45]||queue\_cards[card4] == queue\_cards[49])

{

if(queue\_cards[card5] == queue\_cards[1]||queue\_cards[card5] == queue\_cards[5]||queue\_cards[card5] == queue\_cards[9]||queue\_cards[card5] == queue\_cards[13]||queue\_cards[card5] == queue\_cards[17]||queue\_cards[card5] == queue\_cards[21]||queue\_cards[card5] == queue\_cards[25]||queue\_cards[card5] == queue\_cards[29]||queue\_cards[card5] == queue\_cards[33]||queue\_cards[card5] == queue\_cards[37]||queue\_cards[card5] == queue\_cards[41]||queue\_cards[card5] == queue\_cards[45]||queue\_cards[card5] == queue\_cards[49])

{

\*check\_flush=1;

}

}

}

}

}

//--------------CLUBS(END)---------------

//--------------HEARTS-------------------

if(queue\_cards[card1] == queue\_cards[2]||queue\_cards[card1] == queue\_cards[6]||queue\_cards[card1] == queue\_cards[10]||queue\_cards[card1] == queue\_cards[14]||queue\_cards[card1] == queue\_cards[18]||queue\_cards[card1] == queue\_cards[22]||queue\_cards[card1] == queue\_cards[26]||queue\_cards[card1] == queue\_cards[30]||queue\_cards[card1] == queue\_cards[34]||queue\_cards[card1] == queue\_cards[38]||queue\_cards[card1] == queue\_cards[42]||queue\_cards[card1] == queue\_cards[46]||queue\_cards[card1] == queue\_cards[50])

{

if(queue\_cards[card2] == queue\_cards[2]||queue\_cards[card2] == queue\_cards[6]||queue\_cards[card2] == queue\_cards[10]||queue\_cards[card2] == queue\_cards[14]||queue\_cards[card2] == queue\_cards[18]||queue\_cards[card2] == queue\_cards[22]||queue\_cards[card2] == queue\_cards[26]||queue\_cards[card2] == queue\_cards[30]||queue\_cards[card2] == queue\_cards[34]||queue\_cards[card2] == queue\_cards[38]||queue\_cards[card2] == queue\_cards[42]||queue\_cards[card2] == queue\_cards[46]||queue\_cards[card2] == queue\_cards[50])

{

if(queue\_cards[card3] == queue\_cards[2]||queue\_cards[card3] == queue\_cards[6]||queue\_cards[card3] == queue\_cards[10]||queue\_cards[card3] == queue\_cards[14]||queue\_cards[card3] == queue\_cards[18]||queue\_cards[card3] == queue\_cards[22]||queue\_cards[card3] == queue\_cards[26]||queue\_cards[card3] == queue\_cards[30]||queue\_cards[card3] == queue\_cards[34]||queue\_cards[card3] == queue\_cards[38]||queue\_cards[card3] == queue\_cards[42]||queue\_cards[card3] == queue\_cards[46]||queue\_cards[card3] == queue\_cards[50])

{

if(queue\_cards[card4] == queue\_cards[2]||queue\_cards[card4] == queue\_cards[6]||queue\_cards[card4] == queue\_cards[10]||queue\_cards[card4] == queue\_cards[14]||queue\_cards[card4] == queue\_cards[18]||queue\_cards[card4] == queue\_cards[22]||queue\_cards[card4] == queue\_cards[26]||queue\_cards[card4] == queue\_cards[30]||queue\_cards[card4] == queue\_cards[34]||queue\_cards[card4] == queue\_cards[38]||queue\_cards[card4] == queue\_cards[42]||queue\_cards[card4] == queue\_cards[46]||queue\_cards[card4] == queue\_cards[50])

{

if(queue\_cards[card5] == queue\_cards[2]||queue\_cards[card5] == queue\_cards[6]||queue\_cards[card5] == queue\_cards[10]||queue\_cards[card5] == queue\_cards[14]||queue\_cards[card5] == queue\_cards[18]||queue\_cards[card5] == queue\_cards[22]||queue\_cards[card5] == queue\_cards[26]||queue\_cards[card5] == queue\_cards[30]||queue\_cards[card5] == queue\_cards[34]||queue\_cards[card5] == queue\_cards[38]||queue\_cards[card5] == queue\_cards[42]||queue\_cards[card5] == queue\_cards[46]||queue\_cards[card5] == queue\_cards[50])

{

\*check\_flush=1;

}

}

}

}

}

//--------------HEARTS(END)--------------

//--------------DIAMONDS-----------------

if(queue\_cards[card1] == queue\_cards[3]||queue\_cards[card1] == queue\_cards[7]||queue\_cards[card1] == queue\_cards[11]||queue\_cards[card1] == queue\_cards[15]||queue\_cards[card1] == queue\_cards[19]||queue\_cards[card1] == queue\_cards[23]||queue\_cards[card1] == queue\_cards[27]||queue\_cards[card1] == queue\_cards[31]||queue\_cards[card1] == queue\_cards[35]||queue\_cards[card1] == queue\_cards[39]||queue\_cards[card1] == queue\_cards[43]||queue\_cards[card1] == queue\_cards[47]||queue\_cards[card1] == queue\_cards[51])

{

if(queue\_cards[card2] == queue\_cards[3]||queue\_cards[card2] == queue\_cards[7]||queue\_cards[card2] == queue\_cards[11]||queue\_cards[card2] == queue\_cards[15]||queue\_cards[card2] == queue\_cards[19]||queue\_cards[card2] == queue\_cards[23]||queue\_cards[card2] == queue\_cards[27]||queue\_cards[card2] == queue\_cards[31]||queue\_cards[card2] == queue\_cards[35]||queue\_cards[card2] == queue\_cards[39]||queue\_cards[card2] == queue\_cards[43]||queue\_cards[card2] == queue\_cards[47]||queue\_cards[card2] == queue\_cards[51])

{

if(queue\_cards[card3] == queue\_cards[3]||queue\_cards[card3] == queue\_cards[7]||queue\_cards[card3] == queue\_cards[11]||queue\_cards[card3] == queue\_cards[15]||queue\_cards[card3] == queue\_cards[19]||queue\_cards[card3] == queue\_cards[23]||queue\_cards[card3] == queue\_cards[27]||queue\_cards[card3] == queue\_cards[31]||queue\_cards[card3] == queue\_cards[35]||queue\_cards[card3] == queue\_cards[39]||queue\_cards[card3] == queue\_cards[43]||queue\_cards[card3] == queue\_cards[47]||queue\_cards[card3] == queue\_cards[51])

{

if(queue\_cards[card4] == queue\_cards[3]||queue\_cards[card4] == queue\_cards[7]||queue\_cards[card4] == queue\_cards[11]||queue\_cards[card4] == queue\_cards[15]||queue\_cards[card4] == queue\_cards[19]||queue\_cards[card4] == queue\_cards[23]||queue\_cards[card4] == queue\_cards[27]||queue\_cards[card4] == queue\_cards[31]||queue\_cards[card4] == queue\_cards[35]||queue\_cards[card4] == queue\_cards[39]||queue\_cards[card4] == queue\_cards[43]||queue\_cards[card4] == queue\_cards[47]||queue\_cards[card4] == queue\_cards[51])

{

if(queue\_cards[card5] == queue\_cards[3]||queue\_cards[card5] == queue\_cards[7]||queue\_cards[card5] == queue\_cards[11]||queue\_cards[card5] == queue\_cards[15]||queue\_cards[card5] == queue\_cards[19]||queue\_cards[card5] == queue\_cards[23]||queue\_cards[card5] == queue\_cards[27]||queue\_cards[card5] == queue\_cards[31]||queue\_cards[card5] == queue\_cards[35]||queue\_cards[card5] == queue\_cards[39]||queue\_cards[card5] == queue\_cards[43]||queue\_cards[card5] == queue\_cards[47]||queue\_cards[card5] == queue\_cards[51])

{

\*check\_flush=1;

}

}

}

}

}

//--------------DIAMONDS(END)------------

}

//--------------CARDS--------------------

int CApp::high\_card(int card1)

{

int check\_temp\_high\_card=1;

if(card1==48||card1==49||card1==50||card1==51)

{

return check\_temp\_high\_card=14;

}

if(card1==44||card1==45||card1==46||card1==47)

{

return check\_temp\_high\_card=13;

}

if(card1==40||card1==41||card1==42||card1==43)

{

return check\_temp\_high\_card=12;

}

if(card1==36||card1==37||card1==38||card1==39)

{

return check\_temp\_high\_card=11;

}

if(card1==32||card1==33||card1==34||card1==35)

{

return check\_temp\_high\_card=10;

}

if(card1==28||card1==29||card1==30||card1==31)

{

return check\_temp\_high\_card=9;

}

if(card1==24||card1==25||card1==26||card1==27)

{

return check\_temp\_high\_card=8;

}

if(card1==20||card1==21||card1==22||card1==23)

{

return check\_temp\_high\_card=7;

}

if(card1==16||card1==17||card1==18||card1==19)

{

return check\_temp\_high\_card=6;

}

if(card1==12||card1==13||card1==14||card1==15)

{

return check\_temp\_high\_card=5;

}

if(card1==8||card1==9||card1==10||card1==11)

{

return check\_temp\_high\_card=4;

}

if(card1==4||card1==5||card1==6||card1==7)

{

return check\_temp\_high\_card=3;

}

if(card1==0||card1==1||card1==2||card1==3)

{

return check\_temp\_high\_card=2;

}

}

//--------------CARDS--------------------

void CApp::combinations\_full\_house(int \*check\_full\_house, int \*check\_double, int \*check\_double2, int \*check\_double3, int \*check\_triple, int \*check\_triple2)

{

if(\*check\_double!=\*check\_triple&&\*check\_double>0&&\*check\_triple>0)

{

\*check\_full\_house=1;

}

if(\*check\_double2!=\*check\_triple&&\*check\_double2>0&&\*check\_triple>0)

{

\*check\_full\_house=1;

}

if(\*check\_double3!=\*check\_triple&&\*check\_double3>0&&\*check\_triple>0)

{

\*check\_full\_house=1;

}

}

//--------------CARDS--------------------

void CApp::high\_card\_all(int card1,int card2, int \*check\_high)

{

//---------------------------HIGH-CARD---------------------------

if(check\_temp\_high\_card!=1)

{

int temp\_hc1;

int temp\_hc2;

int temp\_hc3;

int temp\_hc4;

int temp\_hc5;

int temp\_hc6;

int temp\_hc7;

temp\_hc1=high\_card(card1);

temp\_hc2=high\_card(card2);

temp\_hc3=high\_card(card\_1\_all);

temp\_hc4=high\_card(card\_2\_all);

temp\_hc5=high\_card(card\_3\_all);

temp\_hc6=high\_card(card\_4\_all);

temp\_hc7=high\_card(card\_5\_all);

compare\_temp( temp\_hc1, temp\_hc2, temp\_hc3, temp\_hc4, temp\_hc5, temp\_hc6, temp\_hc7, check\_high);

compare\_temp( temp\_hc2, temp\_hc1, temp\_hc3, temp\_hc4, temp\_hc5, temp\_hc6, temp\_hc7, check\_high);

compare\_temp( temp\_hc3, temp\_hc1, temp\_hc2, temp\_hc4, temp\_hc5, temp\_hc6, temp\_hc7, check\_high);

compare\_temp( temp\_hc4, temp\_hc1, temp\_hc2, temp\_hc3, temp\_hc5, temp\_hc6, temp\_hc7, check\_high);

compare\_temp( temp\_hc5, temp\_hc1, temp\_hc2, temp\_hc3, temp\_hc4, temp\_hc6, temp\_hc7, check\_high);

compare\_temp( temp\_hc6, temp\_hc1, temp\_hc2, temp\_hc3, temp\_hc4, temp\_hc5, temp\_hc7, check\_high);

compare\_temp( temp\_hc7, temp\_hc1, temp\_hc2, temp\_hc3, temp\_hc4, temp\_hc5, temp\_hc6, check\_high);

}

//---------------------------HIGH-CARD(END)----------------------

}

//--------------CARDS--------------------

void CApp::compare\_temp(int temp1,int temp2,int temp3,int temp4,int temp5,int temp6,int temp7, int \*check\_high)

{

if(temp1>=temp2&&temp1>=temp3&&temp1>=temp4&&temp1>=temp5&&temp1>=temp6&&temp1>=temp7)

{

\*check\_high=temp1;

check\_temp\_high\_card=1;

}

}

//--------------CARDS--------------------

void CApp::combinations\_check(int card1, int card2, int \*check\_flush\_royal, int \*check\_straight\_flush, int \*check\_kare, int \*check\_triple, int \*check\_triple2, int \*check\_double, int \*check\_double2, int \*check\_double3, int \*check\_straigh, int \*check\_flush, int \*check\_full\_house, int \*check\_high, int \*straight\_temp)

{

if(temp\_card==9)

{

//---------------CHECK-COMBINATIONS-TRIPLE--------------------

//----------------check 2 hands card with 1 card table--------

combinations\_triple(card1,card2,card\_1\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card2,card\_2\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card2,card\_3\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card2,card\_4\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card2,card\_5\_all, check\_triple, check\_triple2);

//----------------check 2 hands card with 1 card table(END)---

//----------------check 1 hands card with 2 card table--------

//---------------------------1 player-------------------------

combinations\_triple(card1,card\_1\_all,card\_2\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card\_1\_all,card\_3\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card\_1\_all,card\_4\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card\_1\_all,card\_5\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card\_2\_all,card\_3\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card\_2\_all,card\_4\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card\_2\_all,card\_5\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card\_3\_all,card\_4\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card\_3\_all,card\_5\_all, check\_triple, check\_triple2);

combinations\_triple(card1,card\_4\_all,card\_5\_all, check\_triple, check\_triple2);

//---------------------------2 player-------------------------

combinations\_triple(card2,card\_1\_all,card\_2\_all, check\_triple, check\_triple2);

combinations\_triple(card2,card\_1\_all,card\_3\_all, check\_triple, check\_triple2);

combinations\_triple(card2,card\_1\_all,card\_4\_all, check\_triple, check\_triple2);

combinations\_triple(card2,card\_1\_all,card\_5\_all, check\_triple, check\_triple2);

combinations\_triple(card2,card\_2\_all,card\_3\_all, check\_triple, check\_triple2);

combinations\_triple(card2,card\_2\_all,card\_4\_all, check\_triple, check\_triple2);

combinations\_triple(card2,card\_2\_all,card\_5\_all, check\_triple, check\_triple2);

combinations\_triple(card2,card\_3\_all,card\_4\_all, check\_triple, check\_triple2);

combinations\_triple(card2,card\_3\_all,card\_5\_all, check\_triple, check\_triple2);

combinations\_triple(card2,card\_4\_all,card\_5\_all, check\_triple, check\_triple2);

//----------------check 1 hands card with 2 card table(END)---

//--------------------------table-----------------------------

combinations\_triple(card\_1\_all,card\_2\_all,card\_3\_all, check\_triple, check\_triple2);

combinations\_triple(card\_1\_all,card\_2\_all,card\_4\_all, check\_triple, check\_triple2);

combinations\_triple(card\_1\_all,card\_2\_all,card\_5\_all, check\_triple, check\_triple2);

combinations\_triple(card\_1\_all,card\_3\_all,card\_4\_all, check\_triple, check\_triple2);

combinations\_triple(card\_1\_all,card\_3\_all,card\_5\_all, check\_triple, check\_triple2);

combinations\_triple(card\_1\_all,card\_4\_all,card\_5\_all, check\_triple, check\_triple2);

combinations\_triple(card\_2\_all,card\_3\_all,card\_4\_all, check\_triple, check\_triple2);

combinations\_triple(card\_2\_all,card\_3\_all,card\_5\_all, check\_triple, check\_triple2);

combinations\_triple(card\_2\_all,card\_4\_all,card\_5\_all, check\_triple, check\_triple2);

combinations\_triple(card\_3\_all,card\_4\_all,card\_5\_all, check\_triple, check\_triple2);

//--------------------------table(END)------------------------

//---------------CHECK-COMBINATIONS-TRIPLE(END)---------------

//---------------CHECK-COMBINATIONS-DOUBLE--------------------

//----------------check hands---------------------------------

combinations\_double(card1,card2,check\_double,check\_double2,check\_double3);

//------------------------------------------------------------

//----------------check table with hands----------------------

combinations\_double(card1,card\_1\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card2,card\_1\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card1,card\_2\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card2,card\_2\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card1,card\_3\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card2,card\_3\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card1,card\_4\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card2,card\_4\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card1,card\_5\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card2,card\_5\_all,check\_double,check\_double2,check\_double3);

//------------------------------------------------------------

//-----------------------check table--------------------------

combinations\_double(card\_1\_all,card\_2\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card\_1\_all,card\_3\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card\_1\_all,card\_4\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card\_1\_all,card\_5\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card\_2\_all,card\_3\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card\_2\_all,card\_4\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card\_2\_all,card\_5\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card\_3\_all,card\_4\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card\_3\_all,card\_5\_all,check\_double,check\_double2,check\_double3);

combinations\_double(card\_4\_all,card\_5\_all,check\_double,check\_double2,check\_double3);

//------------------------------------------------------------

//---------------CHECK-COMBINATIONS-DOUBLE(END)---------------

//-----------------CHECK-COMBINATIONS-KARE--------------------

//----------------check 2 hands card with 2 card table--------

combinations\_kare(card1, card2, card\_1\_all, card\_2\_all, check\_kare);

combinations\_kare(card1, card2, card\_1\_all, card\_3\_all, check\_kare);

combinations\_kare(card1, card2, card\_1\_all, card\_4\_all, check\_kare);

combinations\_kare(card1, card2, card\_1\_all, card\_5\_all, check\_kare);

combinations\_kare(card1, card2, card\_2\_all, card\_3\_all, check\_kare);

combinations\_kare(card1, card2, card\_2\_all, card\_4\_all, check\_kare);

combinations\_kare(card1, card2, card\_2\_all, card\_5\_all, check\_kare);

combinations\_kare(card1, card2, card\_3\_all, card\_4\_all, check\_kare);

combinations\_kare(card1, card2, card\_3\_all, card\_5\_all, check\_kare);

combinations\_kare(card1, card2, card\_4\_all, card\_5\_all, check\_kare);

//----------------check 2 hands card with 2 card table(END)---

//----------------check 1 hands card with 3 card table--------

//---------------------------1 player-------------------------

combinations\_kare(card1, card\_1\_all, card\_2\_all, card\_3\_all, check\_kare);

combinations\_kare(card1, card\_1\_all, card\_2\_all, card\_4\_all, check\_kare);

combinations\_kare(card1, card\_1\_all, card\_2\_all, card\_5\_all, check\_kare);

combinations\_kare(card1, card\_1\_all, card\_3\_all, card\_4\_all, check\_kare);

combinations\_kare(card1, card\_1\_all, card\_3\_all, card\_5\_all, check\_kare);

combinations\_kare(card1, card\_1\_all, card\_4\_all, card\_5\_all, check\_kare);

combinations\_kare(card1, card\_2\_all, card\_3\_all, card\_4\_all, check\_kare);

combinations\_kare(card1, card\_2\_all, card\_4\_all, card\_5\_all, check\_kare);

combinations\_kare(card1, card\_3\_all, card\_4\_all, card\_5\_all, check\_kare);

//---------------------------2 player-------------------------

combinations\_kare(card2, card\_1\_all, card\_2\_all, card\_3\_all, check\_kare);

combinations\_kare(card2, card\_1\_all, card\_2\_all, card\_4\_all, check\_kare);

combinations\_kare(card2, card\_1\_all, card\_2\_all, card\_5\_all, check\_kare);

combinations\_kare(card2, card\_1\_all, card\_3\_all, card\_4\_all, check\_kare);

combinations\_kare(card2, card\_1\_all, card\_3\_all, card\_5\_all, check\_kare);

combinations\_kare(card2, card\_1\_all, card\_4\_all, card\_5\_all, check\_kare);

combinations\_kare(card2, card\_2\_all, card\_3\_all, card\_4\_all, check\_kare);

combinations\_kare(card2, card\_2\_all, card\_4\_all, card\_5\_all, check\_kare);

combinations\_kare(card2, card\_3\_all, card\_4\_all, card\_5\_all, check\_kare);

//----------------check 1 hands card with 3 card table(END)---

//-----------------------------table--------------------------

combinations\_kare(card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, check\_kare);

combinations\_kare(card\_1\_all, card\_2\_all, card\_3\_all, card\_5\_all, check\_kare);

combinations\_kare(card\_1\_all, card\_2\_all, card\_4\_all, card\_5\_all, check\_kare);

combinations\_kare(card\_1\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_kare);

combinations\_kare(card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_kare);

//--------------------------table(END)------------------------

//---------------CHECK-COMBINATIONS-KARE(END)-----------------

//-------------CHECK-COMBINATIONS-STRAIGHT-FLASH--------------

//----------------check 2 hands card with 3 card table--------

combinations\_straight\_flash(card1, card2, card\_1\_all, card\_2\_all, card\_3\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card2, card\_1\_all, card\_3\_all, card\_2\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card2, card\_3\_all, card\_2\_all, card\_1\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card2, card\_1\_all, card\_2\_all, card\_4\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card2, card\_1\_all, card\_2\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card2, card\_1\_all, card\_3\_all, card\_4\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card2, card\_1\_all, card\_3\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card2, card\_1\_all, card\_4\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card2, card\_2\_all, card\_3\_all, card\_4\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card2, card\_2\_all, card\_3\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card2, card\_2\_all, card\_4\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card2, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

//----------------check 2 hands card with 3 card table(END)---

//----------------check 1 hands card with 4 card table--------

//---------------------------1 player-------------------------

combinations\_straight\_flash(card1, card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card\_1\_all, card\_2\_all, card\_3\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card\_1\_all, card\_2\_all, card\_4\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card\_1\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card1, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

//---------------------------2 player-------------------------

combinations\_straight\_flash(card2, card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card2, card\_1\_all, card\_2\_all, card\_3\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card2, card\_1\_all, card\_2\_all, card\_4\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card2, card\_1\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

combinations\_straight\_flash(card2, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

//----------------check 1 hands card with 3 card table(END)---

//-----------------------------table--------------------------

combinations\_straight\_flash(card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush\_royal, check\_straight\_flush);

//--------------------------table(END)------------------------

//----------CHECK-COMBINATIONS-STRAIGHT-FLASH(END)------------

//-----------------CHECK-COMBINATIONS-FLASH-------------------

//----------------check 2 hands card with 3 card table--------

combinations\_flash(card1, card2, card\_1\_all, card\_2\_all, card\_3\_all, check\_flush);

combinations\_flash(card1, card2, card\_1\_all, card\_3\_all, card\_2\_all, check\_flush);

combinations\_flash(card1, card2, card\_3\_all, card\_2\_all, card\_1\_all, check\_flush);

combinations\_flash(card1, card2, card\_1\_all, card\_2\_all, card\_4\_all, check\_flush);

combinations\_flash(card1, card2, card\_1\_all, card\_2\_all, card\_5\_all, check\_flush);

combinations\_flash(card1, card2, card\_1\_all, card\_3\_all, card\_4\_all, check\_flush);

combinations\_flash(card1, card2, card\_1\_all, card\_3\_all, card\_5\_all, check\_flush);

combinations\_flash(card1, card2, card\_1\_all, card\_4\_all, card\_5\_all, check\_flush);

combinations\_flash(card1, card2, card\_2\_all, card\_3\_all, card\_4\_all, check\_flush);

combinations\_flash(card1, card2, card\_2\_all, card\_3\_all, card\_5\_all, check\_flush);

combinations\_flash(card1, card2, card\_2\_all, card\_4\_all, card\_5\_all, check\_flush);

combinations\_flash(card1, card2, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush);

//----------------check 2 hands card with 3 card table(END)---

//----------------check 1 hands card with 4 card table--------

//---------------------------1 player-------------------------

combinations\_flash(card1, card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, check\_flush);

combinations\_flash(card1, card\_1\_all, card\_2\_all, card\_3\_all, card\_5\_all, check\_flush);

combinations\_flash(card1, card\_1\_all, card\_2\_all, card\_4\_all, card\_5\_all, check\_flush);

combinations\_flash(card1, card\_1\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush);

combinations\_flash(card1, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush);

//---------------------------2 player-------------------------

combinations\_flash(card2, card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, check\_flush);

combinations\_flash(card2, card\_1\_all, card\_2\_all, card\_3\_all, card\_5\_all, check\_flush);

combinations\_flash(card2, card\_1\_all, card\_2\_all, card\_4\_all, card\_5\_all, check\_flush);

combinations\_flash(card2, card\_1\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush);

combinations\_flash(card2, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush);

//----------------check 1 hands card with 3 card table(END)---

//-----------------------------table--------------------------

combinations\_flash(card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, check\_flush);

//--------------------------table(END)------------------------

//-----------------CHECK-COMBINATIONS-FLASH(END)--------------

//-----------------CHECK-COMBINATIONS-STRAIGHT----------------

for(int i = 32;i >= 0; i-=4 )

{

if(check\_temp\_card\_straight!=1)

{

//----------------check 2 hands card with 3 card table--------

combinations\_straight(card1, card2, card\_1\_all, card\_2\_all, card\_3\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_3\_all, card\_2\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_3\_all, card\_2\_all, card\_1\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_2\_all, card\_4\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_2\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_3\_all, card\_4\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_3\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_4\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_2\_all, card\_3\_all, card\_4\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_2\_all, card\_3\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_2\_all, card\_4\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_3\_all, card\_4\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

//----------------check 2 hands card with 3 card table(END)---

//----------------check 1 hands card with 4 card table--------

//---------------------------1 player-------------------------

combinations\_straight(card1, card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card\_1\_all, card\_2\_all, card\_3\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card\_1\_all, card\_2\_all, card\_4\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card\_1\_all, card\_3\_all, card\_4\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

//---------------------------2 player-------------------------

combinations\_straight(card2, card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card2, card\_1\_all, card\_2\_all, card\_3\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card2, card\_1\_all, card\_2\_all, card\_4\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card2, card\_1\_all, card\_3\_all, card\_4\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

combinations\_straight(card2, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

//----------------check 1 hands card with 3 card table(END)---

//-----------------------------table--------------------------

combinations\_straight(card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, i, 0, \*straight\_temp, check\_straigh);

//--------------------------table(END)------------------------

(\*straight\_temp)--;

}

}

//-----------------CHECK-COMBINATIONS-STRAIGHT(END)-----------

if(check\_temp\_card\_straight!=1)

{

\*straight\_temp=1;

//----------------check 2 hands card with 3 card table--------

combinations\_straight(card1, card2, card\_1\_all, card\_2\_all, card\_3\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_3\_all, card\_2\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_3\_all, card\_2\_all, card\_1\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_2\_all, card\_4\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_2\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_3\_all, card\_4\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_3\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_1\_all, card\_4\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_2\_all, card\_3\_all, card\_4\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_2\_all, card\_3\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_2\_all, card\_4\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card2, card\_3\_all, card\_4\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

//----------------check 2 hands card with 3 card table(END)---

//----------------check 1 hands card with 4 card table--------

//---------------------------1 player-------------------------

combinations\_straight(card1, card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card\_1\_all, card\_2\_all, card\_3\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card\_1\_all, card\_2\_all, card\_4\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card\_1\_all, card\_3\_all, card\_4\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card1, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

//---------------------------2 player-------------------------

combinations\_straight(card2, card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card2, card\_1\_all, card\_2\_all, card\_3\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card2, card\_1\_all, card\_2\_all, card\_4\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card2, card\_1\_all, card\_3\_all, card\_4\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

combinations\_straight(card2, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

//----------------check 1 hands card with 3 card table(END)---

//-----------------------------table--------------------------

combinations\_straight(card\_1\_all, card\_2\_all, card\_3\_all, card\_4\_all, card\_5\_all, 0, 32, \*straight\_temp, check\_straigh);

//--------------------------table(END)------------------------

\*straight\_temp=1;

}

//-----------------CHECK-COMBINATIONS-STRAIGHT(END)-----------

combinations\_full\_house(check\_full\_house, check\_double, check\_double2, check\_double3, check\_triple, check\_triple2);

high\_card\_all(card1,card2, check\_high);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_2\_player()

{

card\_2\_player=(rand() % 52);

while(card\_2\_player==card\_1\_player)

{

card\_2\_player=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_1\_computer1()

{

card\_1\_computer1=(rand() % 52);

while(card\_1\_computer1==card\_1\_player||card\_1\_computer1==card\_2\_player)

{

card\_1\_computer1=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_2\_computer1()

{

card\_2\_computer1=(rand() % 52);

while(card\_2\_computer1==card\_1\_player||card\_2\_computer1==card\_2\_player||card\_2\_computer1==card\_1\_computer1)

{

card\_2\_computer1=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_1\_computer2()

{

card\_1\_computer2=(rand() % 52);

while(card\_1\_computer2==card\_1\_player||card\_1\_computer2==card\_2\_player||card\_1\_computer2==card\_1\_computer1||card\_1\_computer2==card\_2\_computer1)

{

card\_1\_computer2=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_2\_computer2()

{

card\_2\_computer2=(rand() % 52);

while(card\_2\_computer2==card\_1\_player||card\_2\_computer2==card\_2\_player||card\_2\_computer2==card\_1\_computer1||card\_2\_computer2==card\_2\_computer1||card\_2\_computer2==card\_1\_computer2)

{

card\_2\_computer2=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_1\_computer3()

{

card\_1\_computer3=(rand() % 52);

while(card\_1\_computer3==card\_1\_player||card\_1\_computer3==card\_2\_player||card\_1\_computer3==card\_1\_computer1||card\_1\_computer3==card\_2\_computer1||card\_1\_computer3==card\_1\_computer2||card\_1\_computer3==card\_2\_computer2)

{

card\_1\_computer3=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_2\_computer3()

{

card\_2\_computer3=(rand() % 52);

while(card\_2\_computer3==card\_1\_player||card\_2\_computer3==card\_2\_player||card\_2\_computer3==card\_1\_computer1||card\_2\_computer3==card\_2\_computer1||card\_2\_computer3==card\_1\_computer2||card\_2\_computer3==card\_2\_computer2||card\_2\_computer3==card\_1\_computer3)

{

card\_2\_computer3=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_1\_computer4()

{

card\_1\_computer4=(rand() % 52);

while(card\_1\_computer4==card\_1\_player||card\_1\_computer4==card\_2\_player||card\_1\_computer4==card\_1\_computer1||card\_1\_computer4==card\_2\_computer1||card\_1\_computer4==card\_1\_computer2||card\_1\_computer4==card\_2\_computer2||card\_1\_computer4==card\_1\_computer3||card\_1\_computer4==card\_2\_computer3)

{

card\_1\_computer4=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_2\_computer4()

{

card\_2\_computer4=(rand() % 52);

while(card\_2\_computer4==card\_1\_player||card\_2\_computer4==card\_2\_player||card\_2\_computer4==card\_1\_computer1||card\_2\_computer4==card\_2\_computer1||card\_2\_computer4==card\_1\_computer2||card\_2\_computer4==card\_2\_computer2||card\_2\_computer4==card\_1\_computer3||card\_2\_computer4==card\_2\_computer3||card\_2\_computer4==card\_1\_computer4)

{

card\_2\_computer4=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_1\_computer5()

{

card\_1\_computer5=(rand() % 52);

while(card\_1\_computer5==card\_1\_player||card\_1\_computer5==card\_2\_player||card\_1\_computer5==card\_1\_computer1||card\_1\_computer5==card\_2\_computer1||card\_1\_computer5==card\_1\_computer2||card\_1\_computer5==card\_2\_computer2||card\_1\_computer5==card\_1\_computer3||card\_1\_computer5==card\_2\_computer3||card\_1\_computer5==card\_1\_computer4||card\_1\_computer5==card\_2\_computer4)

{

card\_1\_computer5=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_2\_computer5()

{

card\_2\_computer5=(rand() % 52);

while(card\_2\_computer5==card\_1\_player||card\_2\_computer5==card\_2\_player||card\_2\_computer5==card\_1\_computer1||card\_2\_computer5==card\_2\_computer1||card\_2\_computer5==card\_1\_computer2||card\_2\_computer5==card\_2\_computer2||card\_2\_computer5==card\_1\_computer3||card\_2\_computer5==card\_2\_computer3||card\_2\_computer5==card\_1\_computer4||card\_2\_computer5==card\_2\_computer4||card\_2\_computer5==card\_1\_computer5)

{

card\_2\_computer5=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_1\_all()

{

card\_1\_all=(rand() % 52);

while(card\_1\_all==card\_1\_player||card\_1\_all==card\_2\_player||card\_1\_all==card\_1\_computer1||card\_1\_all==card\_2\_computer1||card\_1\_all==card\_1\_computer2||card\_1\_all==card\_2\_computer2||card\_1\_all==card\_1\_computer3||card\_1\_all==card\_2\_computer3||card\_1\_all==card\_1\_computer4||card\_1\_all==card\_2\_computer4||card\_1\_all==card\_1\_computer5||card\_1\_all==card\_2\_computer5)

{

card\_1\_all=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_2\_all()

{

card\_2\_all=(rand() % 52);

while(card\_2\_all==card\_1\_player||card\_2\_all==card\_2\_player||card\_2\_all==card\_1\_all||card\_2\_all==card\_1\_computer1||card\_2\_all==card\_2\_computer1||card\_2\_all==card\_1\_computer2||card\_2\_all==card\_2\_computer2||card\_2\_all==card\_1\_computer3||card\_2\_all==card\_2\_computer3||card\_2\_all==card\_1\_computer4||card\_2\_all==card\_2\_computer4||card\_2\_all==card\_1\_computer5||card\_2\_all==card\_2\_computer5)

{

card\_2\_all=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_3\_all()

{

card\_3\_all=(rand() % 52);

while(card\_3\_all==card\_1\_player||card\_3\_all==card\_2\_player||card\_3\_all==card\_1\_all||card\_3\_all==card\_2\_all||card\_3\_all==card\_1\_computer1||card\_3\_all==card\_2\_computer1||card\_3\_all==card\_1\_computer2||card\_3\_all==card\_2\_computer2||card\_3\_all==card\_1\_computer3||card\_3\_all==card\_2\_computer3||card\_3\_all==card\_1\_computer4||card\_3\_all==card\_2\_computer4||card\_3\_all==card\_1\_computer5||card\_3\_all==card\_2\_computer5)

{

card\_3\_all=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_4\_all()

{

card\_4\_all=(rand() % 52);

while(card\_4\_all==card\_1\_player||card\_4\_all==card\_2\_player||card\_4\_all==card\_1\_all||card\_4\_all==card\_2\_all||card\_4\_all==card\_3\_all||card\_4\_all==card\_1\_computer1||card\_4\_all==card\_2\_computer1||card\_4\_all==card\_1\_computer2||card\_4\_all==card\_2\_computer2||card\_4\_all==card\_1\_computer3||card\_4\_all==card\_2\_computer3||card\_4\_all==card\_1\_computer4||card\_4\_all==card\_2\_computer4||card\_4\_all==card\_1\_computer5||card\_4\_all==card\_2\_computer5)

{

card\_4\_all=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::random\_card\_5\_all()

{

card\_5\_all=(rand() % 52);

while(card\_5\_all==card\_1\_player||card\_5\_all==card\_2\_player||card\_5\_all==card\_1\_all||card\_5\_all==card\_2\_all||card\_5\_all==card\_3\_all||card\_5\_all==card\_4\_all||card\_5\_all==card\_1\_computer1||card\_5\_all==card\_2\_computer1||card\_5\_all==card\_1\_computer2||card\_5\_all==card\_2\_computer2||card\_5\_all==card\_1\_computer3||card\_5\_all==card\_2\_computer3||card\_5\_all==card\_1\_computer4||card\_5\_all==card\_2\_computer4||card\_5\_all==card\_1\_computer5||card\_5\_all==card\_2\_computer5)

{

card\_5\_all=(rand() % 52);

}

}

//--------------CARDS--------------------

void CApp::check\_card\_win()

{

int check\_rfl=1;

int check\_fl=1;

int check\_tr=1;

int check\_ful=1;

int checl\_db=1;

int chek\_hg=1;

int check\_str=1;

int temp\_flhs=0;

int str\_tp=0;

int db\_temp=0;

int db1\_temp=0;

int db2\_temp=0;

int db\_max=0;

int db2\_max=0;

int hg\_temp=0;

int tp\_fl=0;

if(proverkaP==0)

{

cout<<"\t\t\t "<<round\_card<<" игра"<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<" Игрок"<<"\t\t\t "; cout<<"1 Компьютер"<<"\t\t "; cout<<"2 Компьютер"<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<" Флеш рояль: "<<check\_flush\_royalP<<"\t \t "; cout<<"Флеш рояль: "<<check\_flush\_royalC1<<"\t \t "; cout<<"Флеш рояль: "<<check\_flush\_royalC2<<endl;

cout<<" Стрит флеш: "<<check\_straight\_flushP<<"\t \t "; cout<<"Стрит флеш: "<<check\_straight\_flushC1<<"\t \t "; cout<<"Стрит флеш: "<<check\_straight\_flushC2<<endl;

cout<<" Каре: "<<check\_kareP<<"\t \t "; cout<<"Каре: "<<check\_kareC1<<"\t \t "; cout<<"Каре: "<<check\_kareC2<<endl;

cout<<" Фул хаус: "<<check\_full\_houseP<<"\t \t "; cout<<"Фул хаус: "<<check\_full\_houseC1<<"\t \t "; cout<<"Фул хаус: "<<check\_full\_houseC2<<endl;

cout<<" Флеш: "<<check\_flushP<<"\t \t "; cout<<"Флеш: "<<check\_flushC1<<"\t \t "; cout<<"Флеш: "<<check\_flushC2<<endl;

cout<<" Стрит: "<<check\_straighP<<"\t \t "; cout<<"Стрит: "<<check\_straighC1<<"\t \t "; cout<<"Стрит: "<<check\_straighC2<<endl;

cout<<" 1 тройка: "<<check\_tripleP<<"\t \t "; cout<<"1 тройка: "<<check\_tripleC1<<"\t \t "; cout<<"1 тройка: "<<check\_tripleC2<<endl;

cout<<" 2 тройка: "<<check\_triple2P<<"\t \t "; cout<<"2 тройка: "<<check\_triple2C1<<"\t \t "; cout<<"2 тройка: "<<check\_triple2C2<<endl;

cout<<" 1 пара: "<<check\_doubleP<<"\t \t "; cout<<"1 пара: "<<check\_doubleC1<<"\t \t "; cout<<"1 пара: "<<check\_doubleC2<<endl;

cout<<" 2 пара: "<<check\_double2P<<"\t \t "; cout<<"2 пара: "<<check\_double2C1<<"\t \t "; cout<<"2 пара: "<<check\_double2C2<<endl;

cout<<" 3 пара: "<<check\_double3P<<"\t \t "; cout<<"3 пара: "<<check\_double3C1<<"\t \t "; cout<<"3 пара: "<<check\_double3C2<<endl;

cout<<" Высшая карта: "<<check\_highP<<"\t "; cout<<"Высшая карта: "<<check\_highC1<<"\t "; cout<<"Высшая карта: "<<check\_highC2<<endl;

cout<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<" 3 Компьютер"<<"\t\t "; cout<<"4 Компьютер"<<"\t\t "; cout<<"5 Компьютер"<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<endl<<" Флеш рояль: "<<check\_flush\_royalC3<<"\t \t "; cout<<"Флеш рояль: "<<check\_flush\_royalC4<<"\t \t "; cout<<"Флеш рояль: "<<check\_flush\_royalC5<<endl;

cout<<" Стрит флеш: "<<check\_straight\_flushC3<<"\t \t "; cout<<"Стрит флеш: "<<check\_straight\_flushC4<<"\t \t "; cout<<"Стрит флеш: "<<check\_straight\_flushC5<<endl;

cout<<" Каре: "<<check\_kareC3<<"\t \t "; cout<<"Каре: "<<check\_kareC4<<"\t \t "; cout<<"Каре: "<<check\_kareC5<<endl;

cout<<" Фул хаус: "<<check\_full\_houseC3<<"\t \t "; cout<<"Фул хаус: "<<check\_full\_houseC4<<"\t \t "; cout<<"Фул хаус: "<<check\_full\_houseC5<<endl;

cout<<" Флеш: "<<check\_flushC3<<"\t \t "; cout<<"Флеш: "<<check\_flushC4<<"\t \t "; cout<<"Флеш: "<<check\_flushC5<<endl;

cout<<" Стрит: "<<check\_straighC3<<"\t \t "; cout<<"Стрит: "<<check\_straighC4<<"\t \t "; cout<<"Стрит: "<<check\_straighC5<<endl;

cout<<" 1 тройка: "<<check\_tripleC3<<"\t \t "; cout<<"1 тройка: "<<check\_tripleC4<<"\t \t "; cout<<"1 тройка: "<<check\_tripleC5<<endl;

cout<<" 2 тройка: "<<check\_triple2C3<<"\t \t "; cout<<"2 тройка: "<<check\_triple2C4<<"\t \t "; cout<<"2 тройка: "<<check\_triple2C5<<endl;

cout<<" 1 пара: "<<check\_doubleC3<<"\t \t "; cout<<"1 пара: "<<check\_doubleC4<<"\t \t "; cout<<"1 пара: "<<check\_doubleC5<<endl;

cout<<" 2 пара: "<<check\_double2C3<<"\t \t "; cout<<"2 пара: "<<check\_double2C4<<"\t \t "; cout<<"2 пара: "<<check\_double2C5<<endl;

cout<<" 3 пара: "<<check\_double3C3<<"\t \t "; cout<<"3 пара: "<<check\_double3C4<<"\t \t "; cout<<"3 пара: "<<check\_double3C5<<endl;

cout<<" Высшая карта: "<<check\_highC3<<"\t "; cout<<"Высшая карта: "<<check\_highC4<<"\t "; cout<<"Высшая карта: "<<check\_highC5<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<endl;

}

//--------------------FLUSH-ROYAL---------------------

if(check\_flush\_royalP==1||check\_flush\_royalC1==1||check\_flush\_royalC2==1||check\_flush\_royalC3==1||check\_flush\_royalC4==1||check\_flush\_royalC5==1)

{

if(check\_flush\_royalP>=check\_flush\_royalC1&&check\_flush\_royalP>=check\_flush\_royalC2&&check\_flush\_royalP>=check\_flush\_royalC3&&check\_flush\_royalP>=check\_flush\_royalC4&&check\_flush\_royalP>=check\_flush\_royalC5)

{

if(check\_flush\_royalC1==1)

{

check\_rfl++;

win\_cardC1=1;

}

if(check\_flush\_royalC2==1)

{

check\_rfl++;

win\_cardC2=1;

}

if(check\_flush\_royalC3==1)

{

check\_rfl++;

win\_cardC3=1;

}

if(check\_flush\_royalC4==1)

{

check\_rfl++;

win\_cardC4=1;

}

if(check\_flush\_royalC5==1)

{

check\_rfl++;

win\_cardC5=1;

}

money+=moneybox/check\_rfl;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

}

if(check\_flush\_royalP<check\_flush\_royalC1||check\_flush\_royalP<check\_flush\_royalC2||check\_flush\_royalP<check\_flush\_royalC3||check\_flush\_royalP<check\_flush\_royalC4||check\_flush\_royalP<check\_flush\_royalC5)

{

temp\_flhs=1;

temp\_card=10;

}

}

//--------------------STRAIGHT-FLUSH---------------------

else

{

if(check\_straight\_flushP>0||check\_straight\_flushC1>0||check\_straight\_flushC2>0||check\_straight\_flushC3>0||check\_straight\_flushC4>0||check\_straight\_flushC5>0)

{

if(check\_straight\_flushP>=check\_straight\_flushC1&&check\_straight\_flushP>=check\_straight\_flushC2&&check\_straight\_flushP>=check\_straight\_flushC3&&check\_straight\_flushP>=check\_straight\_flushC4&&check\_straight\_flushP>=check\_straight\_flushC5)

{

if(check\_straight\_flushP==check\_straight\_flushC1&&check\_straight\_flushP==check\_straight\_flushC2&&check\_straight\_flushP==check\_straight\_flushC3&&check\_straight\_flushP==check\_straight\_flushC4&&check\_straight\_flushP==check\_straight\_flushC5)

{

temp\_flhs=1;

money+=moneybox/6;

temp\_card=10;

win\_cardP=1;

win\_cardC1=1;

win\_cardC2=1;

win\_cardC3=1;

win\_cardC4=1;

win\_cardC5=1;

}

if(check\_straight\_flushP>check\_straight\_flushC1&&check\_straight\_flushP>check\_straight\_flushC2&&check\_straight\_flushP>check\_straight\_flushC3&&check\_straight\_flushP>check\_straight\_flushC4&&check\_straight\_flushP>check\_straight\_flushC5)

{

temp\_flhs=1;

money+=moneybox;

temp\_card=10;

win\_cardP=1;

}

}

if(check\_straight\_flushP<check\_straight\_flushC1||check\_straight\_flushP<check\_straight\_flushC2||check\_straight\_flushP<check\_straight\_flushC3||check\_straight\_flushP<check\_straight\_flushC4||check\_straight\_flushP<check\_straight\_flushC5)

{

temp\_flhs=1;

temp\_card=10;

}

}

//--------------------KARE-----------------------------

else

{

if(check\_kareP>=1||check\_kareC1>=1||check\_kareC2>=1||check\_kareC3>=1||check\_kareC4>=1||check\_kareC5>=1)

{

if(check\_kareP>=check\_kareC1&&check\_kareP>=check\_kareC2&&check\_kareP>=check\_kareC3&&check\_kareP>=check\_kareC4&&check\_kareP>=check\_kareC5)

{

if(check\_kareP==check\_kareC1&&check\_kareP==check\_kareC2&&check\_kareP==check\_kareC3&&check\_kareP==check\_kareC4&&check\_kareP==check\_kareC5)

{

if(check\_highP>check\_highC1&&check\_highP>check\_highC2&&check\_highP>check\_highC3&&check\_highP>check\_highC4&&check\_highP>check\_highC5)

{

temp\_flhs=1;

money+=moneybox;

temp\_card=10;

}

if(check\_highP==check\_highC1&&check\_highP==check\_highC2&&check\_highP==check\_highC3&&check\_highP==check\_highC4&&check\_highP==check\_highC5)

{

temp\_flhs=1;

money+=moneybox/6;

temp\_card=10;

win\_cardP=1;

win\_cardC1=1;

win\_cardC2=1;

win\_cardC3=1;

win\_cardC4=1;

win\_cardC5=1;

}

if(check\_highP<check\_highC1||check\_highP<check\_highC2||check\_highP<check\_highC3||check\_highP<check\_highC4||check\_highP<check\_highC5)

{

temp\_flhs=1;

temp\_card=10;

}

}

if(check\_kareP>check\_kareC1&&check\_kareP>check\_kareC2&&check\_kareP>check\_kareC3&&check\_kareP>check\_kareC4&&check\_kareP>check\_kareC5)

{

temp\_flhs=1;

money+=moneybox;

temp\_card=10;

win\_cardP=1;

}

}

if(check\_kareP<check\_kareC1||check\_kareP<check\_kareC2||check\_kareP<check\_kareC3||check\_kareP<check\_kareC4||check\_kareP<check\_kareC5)

{

temp\_flhs=1;

temp\_card=10;

}

}

//--------------------FULL-HOUSE----------------------

else

{

if(check\_full\_houseP>=1||check\_full\_houseC1>=1||check\_full\_houseC2>=1||check\_full\_houseC3>=1||check\_full\_houseC4>=1||check\_full\_houseC5>=1)

{

//------------------------PLAYER--------------------

if(check\_tripleP>check\_triple2P)

{

check\_fh\_triple\_P=check\_tripleP;

}

if(check\_triple2P>check\_tripleP)

{

check\_fh\_triple\_P=check\_triple2P;

}

if(check\_doubleP==check\_fh\_triple\_P)

{

if(check\_double2P>check\_double3P)

{

check\_fh\_double\_P=check\_double2P;

}

if(check\_double3P>check\_double2P)

{

check\_fh\_double\_P=check\_double3P;

}

}

if(check\_double2P==check\_fh\_triple\_P)

{

if(check\_doubleP>check\_double3P)

{

check\_fh\_double\_P=check\_doubleP;

}

if(check\_double3P>check\_doubleP)

{

check\_fh\_double\_P=check\_double3P;

}

}

if(check\_double3P==check\_fh\_triple\_P)

{

if(check\_doubleP>check\_double2P)

{

check\_fh\_double\_P=check\_doubleP;

}

if(check\_double2P>check\_doubleP)

{

check\_fh\_double\_P=check\_double2P;

}

}

//------------------------COMPUTER1-----------------

if(check\_tripleC1>check\_triple2C1)

{

check\_fh\_triple\_C1=check\_tripleC1;

}

if(check\_triple2C1>check\_tripleC1)

{

check\_fh\_triple\_C1=check\_triple2C1;

}

if(check\_doubleC1==check\_fh\_triple\_C1)

{

if(check\_double2C1>check\_double3C1)

{

check\_fh\_double\_C1=check\_double2C1;

}

if(check\_double3C1>check\_double2C1)

{

check\_fh\_double\_C1=check\_double3C1;

}

}

if(check\_double2C1==check\_fh\_triple\_C1)

{

if(check\_doubleC1>check\_double3C1)

{

check\_fh\_double\_C1=check\_doubleC1;

}

if(check\_double3C1>check\_doubleC1)

{

check\_fh\_double\_C1=check\_double3C1;

}

}

if(check\_double3C1==check\_fh\_triple\_C1)

{

if(check\_doubleC1>check\_double2C1)

{

check\_fh\_double\_C1=check\_doubleC1;

}

if(check\_double2C1>check\_doubleC1)

{

check\_fh\_double\_C1=check\_double2C1;

}

}

//------------------------COMPUTER2-----------------

if(check\_tripleC2>check\_triple2C2)

{

check\_fh\_triple\_C2=check\_tripleC2;

}

if(check\_triple2C2>check\_tripleC2)

{

check\_fh\_triple\_C2=check\_triple2C2;

}

if(check\_doubleC2==check\_fh\_triple\_C2)

{

if(check\_double2C2>check\_double3C2)

{

check\_fh\_double\_C2=check\_double2C2;

}

if(check\_double3C2>check\_double2C2)

{

check\_fh\_double\_C2=check\_double3C2;

}

}

if(check\_double2C2==check\_fh\_triple\_C2)

{

if(check\_doubleC2>check\_double3C2)

{

check\_fh\_double\_C2=check\_doubleC2;

}

if(check\_double3C2>check\_doubleC2)

{

check\_fh\_double\_C2=check\_double3C2;

}

}

if(check\_double3C2==check\_fh\_triple\_C2)

{

if(check\_doubleC2>check\_double2C2)

{

check\_fh\_double\_C2=check\_doubleC2;

}

if(check\_double2C2>check\_doubleC2)

{

check\_fh\_double\_C2=check\_double2C2;

}

}

//------------------------COMPUTER3-----------------

if(check\_tripleC3>check\_triple2C3)

{

check\_fh\_triple\_C3=check\_tripleC3;

}

if(check\_triple2C3>check\_tripleC3)

{

check\_fh\_triple\_C3=check\_triple2C3;

}

if(check\_doubleC3==check\_fh\_triple\_C3)

{

if(check\_double2C3>check\_double3C3)

{

check\_fh\_double\_C3=check\_double2C3;

}

if(check\_double3C3>check\_double2C3)

{

check\_fh\_double\_C3=check\_double3C3;

}

}

if(check\_double2C3==check\_fh\_triple\_C3)

{

if(check\_doubleC3>check\_double3C3)

{

check\_fh\_double\_C3=check\_doubleC3;

}

if(check\_double3C3>check\_doubleC3)

{

check\_fh\_double\_C3=check\_double3C3;

}

}

if(check\_double3C3==check\_fh\_triple\_C3)

{

if(check\_doubleC3>check\_double2C3)

{

check\_fh\_double\_C3=check\_doubleC3;

}

if(check\_double2C3>check\_doubleC3)

{

check\_fh\_double\_C3=check\_double2C3;

}

}

//------------------------COMPUTER4-----------------

if(check\_tripleC4>check\_triple2C4)

{

check\_fh\_triple\_C4=check\_tripleC4;

}

if(check\_triple2C4>check\_tripleC4)

{

check\_fh\_triple\_C4=check\_triple2C4;

}

if(check\_doubleC4==check\_fh\_triple\_C4)

{

if(check\_double2C4>check\_double3C4)

{

check\_fh\_double\_C4=check\_double2C4;

}

if(check\_double3C4>check\_double2C4)

{

check\_fh\_double\_C4=check\_double3C4;

}

}

if(check\_double2C4==check\_fh\_triple\_C4)

{

if(check\_doubleC4>check\_double3C4)

{

check\_fh\_double\_C4=check\_doubleC4;

}

if(check\_double3C4>check\_doubleC4)

{

check\_fh\_double\_C4=check\_double3C4;

}

}

if(check\_double3C4==check\_fh\_triple\_C4)

{

if(check\_doubleC4>check\_double2C4)

{

check\_fh\_double\_C4=check\_doubleC4;

}

if(check\_double2C4>check\_doubleC4)

{

check\_fh\_double\_C4=check\_double2C4;

}

}

//------------------------COMPUTER5-----------------

if(check\_tripleC5>check\_triple2C5)

{

check\_fh\_triple\_C5=check\_tripleC5;

}

if(check\_triple2C5>check\_tripleC5)

{

check\_fh\_triple\_C5=check\_triple2C5;

}

if(check\_doubleC5==check\_fh\_triple\_C5)

{

if(check\_double2C5>check\_double3C5)

{

check\_fh\_double\_C5=check\_double2C5;

}

if(check\_double3C5>check\_double2C5)

{

check\_fh\_double\_C5=check\_double3C5;

}

}

if(check\_double2C5==check\_fh\_triple\_C5)

{

if(check\_doubleC5>check\_double3C5)

{

check\_fh\_double\_C5=check\_doubleC5;

}

if(check\_double3C5>check\_doubleC5)

{

check\_fh\_double\_C5=check\_double3C5;

}

}

if(check\_double3C5==check\_fh\_triple\_C5)

{

if(check\_doubleC5>check\_double2C5)

{

check\_fh\_double\_C5=check\_doubleC5;

}

if(check\_double2C5>check\_doubleC5)

{

check\_fh\_double\_C5=check\_double2C5;

}

}

//-------------------------------------------------

if(check\_full\_houseP>=check\_full\_houseC1&&check\_full\_houseP>=check\_full\_houseC2&&check\_full\_houseP>=check\_full\_houseC3&&check\_full\_houseP>=check\_full\_houseC4&&check\_full\_houseP>=check\_full\_houseC5)

{

if(check\_fh\_triple\_P>check\_fh\_triple\_C1&&check\_fh\_triple\_P>check\_fh\_triple\_C2&&check\_fh\_triple\_P>check\_fh\_triple\_C3&&check\_fh\_triple\_P>check\_fh\_triple\_C4&&check\_fh\_triple\_P>check\_fh\_triple\_C5)

{

money+=moneybox;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

tp\_fl=1;

}

if(check\_fh\_triple\_P==check\_fh\_triple\_C1||check\_fh\_triple\_P==check\_fh\_triple\_C2||check\_fh\_triple\_P==check\_fh\_triple\_C3||check\_fh\_triple\_P==check\_fh\_triple\_C4||check\_fh\_triple\_P==check\_fh\_triple\_C5)

{

if(win\_cardP==0)

{

if(check\_fh\_double\_P>=check\_fh\_double\_C1&&check\_fh\_double\_P>=check\_fh\_double\_C2&&check\_fh\_double\_P>=check\_fh\_double\_C3&&check\_fh\_double\_P>=check\_fh\_double\_C4&&check\_fh\_double\_P>=check\_fh\_double\_C5)

{

if(check\_fh\_double\_P>check\_fh\_double\_C1&&check\_fh\_double\_P>check\_fh\_double\_C2&&check\_fh\_double\_P>check\_fh\_double\_C3&&check\_fh\_double\_P>check\_fh\_double\_C4&&check\_fh\_double\_P>check\_fh\_double\_C5)

{

money+=moneybox;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

tp\_fl=1;

}

if(check\_fh\_double\_P>check\_fh\_double\_C1&&check\_fh\_double\_P>check\_fh\_double\_C2&&check\_fh\_double\_P>check\_fh\_double\_C3&&check\_fh\_double\_P>check\_fh\_double\_C4&&check\_fh\_double\_P>check\_fh\_double\_C5)

{

money+=moneybox;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

tp\_fl=1;

}

if(check\_fh\_double\_P<check\_fh\_double\_C1||check\_fh\_double\_P<check\_fh\_double\_C2||check\_fh\_double\_P<check\_fh\_double\_C3||check\_fh\_double\_P<check\_fh\_double\_C4||check\_fh\_double\_P<check\_fh\_double\_C5)

{

temp\_flhs=1;

temp\_card=10;

}

if(check\_fh\_double\_P==check\_fh\_double\_C1&&check\_fh\_triple\_P==check\_fh\_triple\_C1)

{

check\_ful++;

}

if(check\_fh\_double\_P==check\_fh\_double\_C2&&check\_fh\_triple\_P==check\_fh\_triple\_C2)

{

check\_ful++;

}

if(check\_fh\_double\_P==check\_fh\_double\_C2&&check\_fh\_triple\_P==check\_fh\_triple\_C2)

{

check\_ful++;

}

if(check\_fh\_double\_P==check\_fh\_double\_C2&&check\_fh\_triple\_P==check\_fh\_triple\_C2)

{

check\_ful++;

}

if(check\_fh\_double\_P==check\_fh\_double\_C2&&check\_fh\_triple\_P==check\_fh\_triple\_C2)

{

check\_ful++;

}

money+=moneybox/check\_ful;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

tp\_fl=1;

}

}

if(check\_fh\_triple\_P<check\_fh\_triple\_C1||check\_fh\_triple\_P<check\_fh\_triple\_C2||check\_fh\_triple\_P<check\_fh\_triple\_C3||check\_fh\_triple\_P<check\_fh\_triple\_C4||check\_fh\_triple\_P<check\_fh\_triple\_C5)

{

temp\_flhs=1;

temp\_card=10;

}

}

if(check\_full\_houseP>check\_full\_houseC1&&check\_full\_houseP>check\_full\_houseC2&&check\_full\_houseP>check\_full\_houseC3&&check\_full\_houseP>check\_full\_houseC4&&check\_full\_houseP>check\_full\_houseC5)

{

money+=moneybox;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

tp\_fl=1;

}

if(check\_full\_houseP<check\_full\_houseC1||check\_full\_houseP<check\_full\_houseC2||check\_full\_houseP<check\_full\_houseC3||check\_full\_houseP<check\_full\_houseC4||check\_full\_houseP<check\_full\_houseC5)

{

temp\_flhs=1;

temp\_card=10;

}

}

if(check\_full\_houseP<check\_full\_houseC1||check\_full\_houseP<check\_full\_houseC2||check\_full\_houseP<check\_full\_houseC3||check\_full\_houseP<check\_full\_houseC4||check\_full\_houseP<check\_full\_houseC5)

{

temp\_flhs=1;

temp\_card=10;

}

}

//--------------------FLUSH-------------------------

else

{

if(check\_flushP>=1||check\_flushC1>=1||check\_flushC2>=1||check\_flushC3>=1||check\_flushC4>=1||check\_flushC5>=1)

{

if(check\_flushP>=check\_flushC1&&check\_flushP>=check\_flushC2&&check\_flushP>=check\_flushC3&&check\_flushP>=check\_flushC4&&check\_flushP>=check\_flushC5)

{

if(check\_flushC1==1)

{

check\_fl++;

win\_cardC1=1;

}

if(check\_flushC2==1)

{

check\_fl++;

win\_cardC2=1;

}

if(check\_flushC3==1)

{

check\_fl++;

win\_cardC3=1;

}

if(check\_flushC4==1)

{

check\_fl++;

win\_cardC4=1;

}

if(check\_flushC5==1)

{

check\_fl++;

win\_cardC5=1;

}

temp\_flhs=1;

money+=moneybox/check\_fl;

temp\_card=10;

win\_cardP=1;

}

if(check\_flushP<check\_flushC1||check\_flushP<check\_flushC2||check\_flushP<check\_flushC3||check\_flushP<check\_flushC4||check\_flushP<check\_flushC5)

{

temp\_flhs=1;

temp\_card=10;

}

}

//--------------------STRAIGHT-----------------------

else

{

if(check\_straighP>=1||check\_straighC1>=1||check\_straighC2>=1||check\_straighC3>=1||check\_straighC4>=1||check\_straighC5>=1)

{

if(check\_straighP>=check\_straighC1&&check\_straighP>=check\_straighC2&&check\_straighP>=check\_straighC3&&check\_straighP>=check\_straighC4&&check\_straighP>=check\_straighC5)

{

if(check\_straighP==check\_straighC1||check\_straighP==check\_straighC2||check\_straighP==check\_straighC3||check\_straighP==check\_straighC4||check\_straighP||check\_straighC5)

{

if(check\_straighP==check\_straighC1)

{

check\_str++;

win\_cardC1=1;

}

if(check\_straighP==check\_straighC2)

{

check\_str++;

win\_cardC2=1;

}

if(check\_straighP==check\_straighC3)

{

check\_str++;

win\_cardC3=1;

}

if(check\_straighP==check\_straighC4)

{

check\_str++;

win\_cardC4=1;

}

if(check\_straighP==check\_straighC5)

{

check\_str++;

win\_cardC5=1;

}

temp\_flhs=1;

money+=moneybox/check\_str;

str\_tp=1;

temp\_card=10;

win\_cardP=1;

}

if(str\_tp==0)

{

if(check\_straighP>check\_straighC1&&check\_straighP>check\_straighC2&&check\_straighP>check\_straighC3&&check\_straighP>check\_straighC4&&check\_straighP>check\_straighC5)

{

temp\_flhs=1;

money+=moneybox;

temp\_card=10;

win\_cardP=1;

}

}

}

if(check\_straighP<check\_straighC1||check\_straighP<check\_straighC2||check\_straighP<check\_straighC3||check\_straighP<check\_straighC4||check\_straighP<check\_straighC5)

{

temp\_flhs=1;

temp\_card=10;

}

}

//--------------------TRIPLE-----------------------

else

{

if(check\_tripleP>0||check\_tripleC1>0||check\_tripleC2>0||check\_tripleC3>0||check\_tripleC4>0||check\_tripleC5>0)

{

if(temp\_flhs==0)

{

//-------------------PLAYER------------------------

if(check\_tripleP>check\_triple2P)

{

check\_tr\_triple\_P=check\_tripleP;

}

if(check\_triple2P>check\_tripleP)

{

check\_tr\_triple\_P=check\_triple2P;

}

//-------------------COMPUTER1---------------------

if(check\_tripleC1>check\_triple2C1)

{

check\_tr\_triple\_C1=check\_tripleC1;

}

if(check\_triple2C1>check\_tripleC1)

{

check\_tr\_triple\_C1=check\_triple2C1;

}

//-------------------COMPUTER2---------------------

if(check\_tripleC2>check\_triple2C2)

{

check\_tr\_triple\_C2=check\_tripleC2;

}

if(check\_triple2C2>check\_tripleC2)

{

check\_tr\_triple\_C2=check\_triple2C2;

}

//-------------------COMPUTER3---------------------

if(check\_tripleC3>check\_triple2C3)

{

check\_tr\_triple\_C3=check\_tripleC3;

}

if(check\_triple2C3>check\_tripleC3)

{

check\_tr\_triple\_C3=check\_triple2C3;

}

//-------------------COMPUTER4---------------------

if(check\_tripleC4>check\_triple2C4)

{

check\_tr\_triple\_C4=check\_tripleC4;

}

if(check\_triple2C4>check\_tripleC4)

{

check\_tr\_triple\_C4=check\_triple2C4;

}

//-------------------COMPUTER5---------------------

if(check\_tripleC5>check\_triple2C5)

{

check\_tr\_triple\_C5=check\_tripleC5;

}

if(check\_triple2C5>check\_tripleC5)

{

check\_tr\_triple\_C5=check\_triple2C5;

}

//-------------------------------------------------

if(check\_tr\_triple\_P>=check\_tr\_triple\_C1&&check\_tr\_triple\_P>=check\_tr\_triple\_C2&&check\_tr\_triple\_P>=check\_tr\_triple\_C3&&check\_tr\_triple\_P>=check\_tr\_triple\_C4&&check\_tr\_triple\_P>=check\_tr\_triple\_C5)

{

if(check\_tr\_triple\_P>check\_tr\_triple\_C1&&check\_tr\_triple\_P>check\_tr\_triple\_C2&&check\_tr\_triple\_P>check\_tr\_triple\_C3&&check\_tr\_triple\_P>check\_tr\_triple\_C4&&check\_tr\_triple\_P>check\_tr\_triple\_C5)

{

money+=moneybox;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

}

if(check\_tr\_triple\_P==check\_tr\_triple\_C1||check\_tr\_triple\_P==check\_tr\_triple\_C2||check\_tr\_triple\_P==check\_tr\_triple\_C3||check\_tr\_triple\_P==check\_tr\_triple\_C4||check\_tr\_triple\_P==check\_tr\_triple\_C5)

{

if((check\_highP>=check\_highC1&&check\_tr\_triple\_P==check\_tr\_triple\_C1)||(check\_highP>=check\_highC2&&check\_tr\_triple\_P==check\_tr\_triple\_C2)||(check\_highP>=check\_highC3&&check\_tr\_triple\_P==check\_tr\_triple\_C3)||(check\_highP>=check\_highC4&&check\_tr\_triple\_P==check\_tr\_triple\_C4)||(check\_highP>=check\_highC5&&check\_tr\_triple\_P==check\_tr\_triple\_C5))

{

if(check\_highP>check\_highC1&&check\_highP>check\_highC2&&check\_highP>check\_highC3&&check\_highP>check\_highC4&&check\_highP>check\_highC5)

{

money+=moneybox;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

}

if(check\_highP==check\_highC1&&check\_tr\_triple\_P==check\_tr\_triple\_C1)

{

check\_tr++;

win\_cardC1=1;

}

if(check\_highP==check\_highC2&&check\_tr\_triple\_P==check\_tr\_triple\_C2)

{

check\_tr++;

win\_cardC2=1;

}

if(check\_highP==check\_highC3&&check\_tr\_triple\_P==check\_tr\_triple\_C3)

{

check\_tr++;

win\_cardC3=1;

}

if(check\_highP==check\_highC4&&check\_tr\_triple\_P==check\_tr\_triple\_C4)

{

check\_tr++;

win\_cardC4=1;

}

if(check\_highP==check\_highC5&&check\_tr\_triple\_P==check\_tr\_triple\_C5)

{

check\_tr++;

win\_cardC5=1;

}

temp\_flhs=1;

money+=moneybox/check\_tr;

temp\_card=10;

win\_cardP=1;

}

}

}

if(check\_tr\_triple\_P<check\_tr\_triple\_C1||check\_tr\_triple\_P<check\_tr\_triple\_C2||check\_tr\_triple\_P<check\_tr\_triple\_C3||check\_tr\_triple\_P<check\_tr\_triple\_C4||check\_tr\_triple\_P<check\_tr\_triple\_C5)

{

temp\_flhs=1;

temp\_card=10;

}

}

}

//--------------------DOUBLE-----------------------

else

{

if(check\_doubleP>0||check\_doubleC1>0||check\_doubleC2>0||check\_doubleC3>0||check\_doubleC4>0||check\_doubleC5>0)

{

if(tp\_fl==0)

{

if(temp\_flhs==0)

{

//-------------------PLAYER------------------------

if(check\_doubleP>check\_double2P&&check\_doubleP>check\_double3P)

{

check\_db\_double\_P=check\_doubleP;

if(check\_double2P>check\_double3P)

{

check\_2db\_double\_P=check\_double2P;

}

if(check\_double3P>check\_double2P)

{

check\_2db\_double\_P=check\_double3P;

}

}

if(check\_double2P>check\_doubleP&&check\_double2P>check\_double3P)

{

check\_db\_double\_P=check\_double2P;

if(check\_doubleP>check\_double3P)

{

check\_2db\_double\_P=check\_doubleP;

}

if(check\_double3P>check\_doubleP)

{

check\_2db\_double\_P=check\_double3P;

}

}

if(check\_double3P>check\_doubleP&&check\_double3P>check\_double2P)

{

check\_db\_double\_P=check\_double3P;

if(check\_doubleP>check\_double2P)

{

check\_2db\_double\_P=check\_doubleP;

}

if(check\_double2P>check\_doubleP)

{

check\_2db\_double\_P=check\_double2P;

}

}

//-------------------COMPUTER1---------------------

if(check\_doubleC1>check\_double2C1&&check\_doubleC1>check\_double3C1)

{

check\_db\_double\_C1=check\_doubleC1;

if(check\_double2C1>check\_double3C1)

{

check\_2db\_double\_C1=check\_double2C1;

}

if(check\_double3C1>check\_double2C1)

{

check\_2db\_double\_C1=check\_double3C1;

}

}

if(check\_double2C1>check\_doubleC1&&check\_double2C1>check\_double3C1)

{

check\_db\_double\_C1=check\_double2C1;

if(check\_doubleC1>check\_double3C1)

{

check\_2db\_double\_C1=check\_doubleC1;

}

if(check\_double3C1>check\_doubleC1)

{

check\_2db\_double\_C1=check\_double3C1;

}

}

if(check\_double3C1>check\_doubleC1&&check\_double3C1>check\_double2C1)

{

check\_db\_double\_C1=check\_double3C1;

if(check\_doubleC1>check\_double2C1)

{

check\_2db\_double\_C1=check\_doubleC1;

}

if(check\_double2C1>check\_doubleC1)

{

check\_2db\_double\_C1=check\_double2C1;

}

}

//-------------------COMPUTER2---------------------

if(check\_doubleC2>check\_double2C2&&check\_doubleC2>check\_double3C2)

{

check\_db\_double\_C2=check\_doubleC2;

if(check\_double2C2>check\_double3C2)

{

check\_2db\_double\_C2=check\_double2C2;

}

if(check\_double3C2>check\_double2C2)

{

check\_2db\_double\_C2=check\_double3C2;

}

}

if(check\_double2C2>check\_doubleC2&&check\_double2C2>check\_double3C2)

{

check\_db\_double\_C2=check\_double2C2;

if(check\_doubleC2>check\_double3C2)

{

check\_2db\_double\_C2=check\_doubleC2;

}

if(check\_double3C2>check\_doubleC2)

{

check\_2db\_double\_C2=check\_double3C2;

}

}

if(check\_double3C2>check\_doubleC2&&check\_double3C2>check\_double2C2)

{

check\_db\_double\_C2=check\_double3C2;

if(check\_doubleC2>check\_double2C2)

{

check\_2db\_double\_C2=check\_doubleC2;

}

if(check\_double2C2>check\_doubleC2)

{

check\_2db\_double\_C2=check\_double2C2;

}

}

//-------------------COMPUTER3---------------------

if(check\_doubleC3>check\_double2C3&&check\_doubleC3>check\_double3C3)

{

check\_db\_double\_C3=check\_doubleC3;

if(check\_double2C3>check\_double3C3)

{

check\_2db\_double\_C3=check\_double2C3;

}

if(check\_double3C3>check\_double2C3)

{

check\_2db\_double\_C3=check\_double3C3;

}

}

if(check\_double2C3>check\_doubleC3&&check\_double2C3>check\_double3C3)

{

check\_db\_double\_C3=check\_double2C3;

if(check\_doubleC3>check\_double3C3)

{

check\_2db\_double\_C3=check\_doubleC3;

}

if(check\_double3C3>check\_doubleC3)

{

check\_2db\_double\_C3=check\_double3C3;

}

}

if(check\_double3C3>check\_doubleC3&&check\_double3C3>check\_double2C3)

{

check\_db\_double\_C3=check\_double3C3;

if(check\_doubleC3>check\_double2C3)

{

check\_2db\_double\_C3=check\_doubleC3;

}

if(check\_double2C3>check\_doubleC3)

{

check\_2db\_double\_C3=check\_double2C3;

}

}

//-------------------COMPUTER4---------------------

if(check\_doubleC4>check\_double2C4&&check\_doubleC4>check\_double3C4)

{

check\_db\_double\_C4=check\_doubleC4;

if(check\_double2C4>check\_double3C4)

{

check\_2db\_double\_C4=check\_double2C4;

}

if(check\_double3C4>check\_double2C4)

{

check\_2db\_double\_C4=check\_double3C4;

}

}

if(check\_double2C4>check\_doubleC4&&check\_double2C4>check\_double3C4)

{

check\_db\_double\_C4=check\_double2C4;

if(check\_doubleC4>check\_double3C4)

{

check\_2db\_double\_C4=check\_doubleC4;

}

if(check\_double3C4>check\_doubleC4)

{

check\_2db\_double\_C4=check\_double3C4;

}

}

if(check\_double3C4>check\_doubleC4&&check\_double3C4>check\_double2C4)

{

check\_db\_double\_C4=check\_double3C4;

if(check\_doubleC4>check\_double2C4)

{

check\_2db\_double\_C4=check\_doubleC4;

}

if(check\_double2C4>check\_doubleC4)

{

check\_2db\_double\_C4=check\_double2C4;

}

}

//-------------------COMPUTER5---------------------

if(check\_doubleC5>check\_double2C5&&check\_doubleC5>check\_double3C5)

{

check\_db\_double\_C5=check\_doubleC5;

if(check\_double2C5>check\_double3C5)

{

check\_2db\_double\_C5=check\_double2C5;

}

if(check\_double3C5>check\_double2C5)

{

check\_2db\_double\_C5=check\_double3C5;

}

}

if(check\_double2C5>check\_doubleC5&&check\_double2C5>check\_double3C5)

{

check\_db\_double\_C5=check\_double2C5;

if(check\_doubleC5>check\_double3C5)

{

check\_2db\_double\_C5=check\_doubleC5;

}

if(check\_double3C5>check\_doubleC5)

{

check\_2db\_double\_C5=check\_double3C5;

}

}

if(check\_double3C5>check\_doubleC5&&check\_double3C5>check\_double2C5)

{

check\_db\_double\_C5=check\_double3C5;

if(check\_doubleC5>check\_double2C5)

{

check\_2db\_double\_C5=check\_doubleC5;

}

if(check\_double2C5>check\_doubleC5)

{

check\_2db\_double\_C5=check\_double2C5;

}

}

//-------------------------------------------------

if(check\_db\_double\_P>check\_db\_double\_C1)

{

if(db\_max<check\_db\_double\_P)db\_max=check\_db\_double\_P;

}

else if(db\_max<check\_db\_double\_C1)db\_max=check\_db\_double\_C1;

if(check\_db\_double\_P>check\_db\_double\_C2)

{

if(db\_max<check\_db\_double\_P)db\_max=check\_db\_double\_P;

}

else if(db\_max<check\_db\_double\_C2)db\_max=check\_db\_double\_C2;

if(check\_db\_double\_P>check\_db\_double\_C3)

{

if(db\_max<check\_db\_double\_P)db\_max=check\_db\_double\_P;

}

else if(db\_max<check\_db\_double\_C3)db\_max=check\_db\_double\_C3;

if(check\_db\_double\_P>check\_db\_double\_C4)

{

if(db\_max<check\_db\_double\_P)db\_max=check\_db\_double\_P;

}

else if(db\_max<check\_db\_double\_C4)db\_max=check\_db\_double\_C4;

if(check\_db\_double\_P>check\_db\_double\_C5)

{

if(db\_max<check\_db\_double\_P)db\_max=check\_db\_double\_P;

}

else if(db\_max<check\_db\_double\_C5)db\_max=check\_db\_double\_C5;

//-------------------------------------------------

if(check\_2db\_double\_P>check\_2db\_double\_C1)

{

if(db2\_max<check\_2db\_double\_P)db2\_max=check\_2db\_double\_P;

}

else if(db2\_max<check\_2db\_double\_C1)db2\_max=check\_2db\_double\_C1;

if(check\_2db\_double\_P>check\_2db\_double\_C2)

{

if(db2\_max<check\_2db\_double\_P)db2\_max=check\_2db\_double\_P;

}

else if(db2\_max<check\_2db\_double\_C2)db2\_max=check\_2db\_double\_C2;

if(check\_2db\_double\_P>check\_2db\_double\_C3)

{

if(db2\_max<check\_2db\_double\_P)db2\_max=check\_2db\_double\_P;

}

else if(db2\_max<check\_2db\_double\_C3)db2\_max=check\_2db\_double\_C3;

if(check\_2db\_double\_P>check\_2db\_double\_C4)

{

if(db2\_max<check\_2db\_double\_P)db2\_max=check\_2db\_double\_P;

}

else if(db2\_max<check\_2db\_double\_C4)db2\_max=check\_2db\_double\_C4;

if(check\_2db\_double\_P>check\_2db\_double\_C5)

{

if(db2\_max<check\_2db\_double\_P)db2\_max=check\_2db\_double\_P;

}

else if(db2\_max<check\_2db\_double\_C5)db2\_max=check\_2db\_double\_C5;

//-------------------------------------------------

if(win\_cardP==0&&win\_cardC1==0&&win\_cardC2==0&&win\_cardC3==0&&win\_cardC4==0&&win\_cardC5==0)

{

if((check\_db\_double\_P>=check\_db\_double\_C1&&check\_db\_double\_P>=check\_db\_double\_C2&&check\_db\_double\_P>=check\_db\_double\_C3&&check\_db\_double\_P>=check\_db\_double\_C4&&check\_db\_double\_P>=check\_db\_double\_C5)||(check\_2db\_double\_P>=check\_2db\_double\_C1&&check\_2db\_double\_P>=check\_2db\_double\_C2&&check\_2db\_double\_P>=check\_2db\_double\_C3&&check\_2db\_double\_P>=check\_2db\_double\_C4&&check\_2db\_double\_P>=check\_2db\_double\_C5))

{

if(check\_2db\_double\_P>0&&check\_2db\_double\_C1==0&&check\_2db\_double\_C2==0&&check\_2db\_double\_C3==0&&check\_2db\_double\_C4==0&&check\_2db\_double\_C5==0)

{

money+=moneybox;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

}

if(win\_cardP==0&&win\_cardC1==0&&win\_cardC2==0&&win\_cardC3==0&&win\_cardC4==0&&win\_cardC5==0)

{

if(check\_db\_double\_P>check\_db\_double\_C1&&check\_2db\_double\_P>0&&check\_db\_double\_P>check\_db\_double\_C2&&check\_2db\_double\_P>0&&check\_db\_double\_P>check\_db\_double\_C3&&check\_2db\_double\_P>0&&check\_db\_double\_P>check\_db\_double\_C4&&check\_2db\_double\_P>0&&check\_db\_double\_P>check\_db\_double\_C5&&check\_2db\_double\_P>0)

{

money+=moneybox;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

}

}

if(check\_2db\_double\_P==0&&check\_2db\_double\_C1==0&&check\_2db\_double\_C2==0&&check\_2db\_double\_C3==0&&check\_2db\_double\_C4==0&&check\_2db\_double\_C5==0)

{

if(check\_db\_double\_P==0&&check\_2db\_double\_P==0)

{

db\_temp=1;

}

if(win\_cardP==0&&win\_cardC1==0&&win\_cardC2==0&&win\_cardC3==0&&win\_cardC4==0&&win\_cardC5==0)

{

if(check\_db\_double\_P>0)

{

if((check\_db\_double\_P>=db2\_max&&check\_2db\_double\_P>=db\_max)||(check\_2db\_double\_P>=db2\_max&&check\_db\_double\_P>=db\_max))

{

if(check\_db\_double\_P>check\_db\_double\_C1&&check\_db\_double\_P>check\_db\_double\_C2&&check\_db\_double\_P>check\_db\_double\_C3&&check\_db\_double\_P>check\_db\_double\_C4&&check\_db\_double\_P>check\_db\_double\_C5)

{

money+=moneybox;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

}

if((check\_highP<check\_highC1&&((check\_db\_double\_C1>=db2\_max&&check\_2db\_double\_C1>=db\_max)||(check\_2db\_double\_C1>=db2\_max&&check\_db\_double\_C1>=db\_max)))||(check\_highP<check\_highC2&&((check\_db\_double\_C2>=db2\_max&&check\_2db\_double\_C2>=db\_max)||(check\_2db\_double\_C2>=db2\_max&&check\_db\_double\_C2>=db\_max)))||(check\_highP<check\_highC3&&((check\_db\_double\_C3>=db2\_max&&check\_2db\_double\_C3>=db\_max)||(check\_2db\_double\_C3>=db2\_max&&check\_db\_double\_C3>=db\_max)))||(check\_highP<check\_highC4&&((check\_db\_double\_C4>=db2\_max&&check\_2db\_double\_C4>=db\_max)||(check\_2db\_double\_C4>=db2\_max&&check\_db\_double\_C4>=db\_max)))||(check\_highP<check\_highC5&&((check\_db\_double\_C5>=db2\_max&&check\_2db\_double\_C5>=db\_max)||(check\_2db\_double\_C5>=db2\_max&&check\_db\_double\_C5>=db\_max))))

{

db\_temp=1;

temp\_flhs=1;

temp\_card=10;

}

if(check\_highP>=check\_highC1||check\_highP>=check\_highC2||check\_highP>=check\_highC3||check\_highP>=check\_highC4||check\_highP>=check\_highC5)

{

if(db\_temp==0)

{

if(win\_cardP==0&&win\_cardC1==0&&win\_cardC2==0&&win\_cardC3==0&&win\_cardC4==0&&win\_cardC5==0)

{

if(check\_highP>check\_highC1&&check\_highP>check\_highC2&&check\_highP>check\_highC3&&check\_highP>check\_highC4&&check\_highP>check\_highC5)

{

money+=moneybox;

temp\_flhs=1;

temp\_card=10;

}

if(win\_cardP==0&&win\_cardC1==0&&win\_cardC2==0&&win\_cardC3==0&&win\_cardC4==0&&win\_cardC5==0)

{

if(check\_highP==check\_highC1&&check\_db\_double\_P==check\_db\_double\_C1)

{

checl\_db++;

win\_cardC1=1;

}

if(check\_highP==check\_highC2&&check\_db\_double\_P==check\_db\_double\_C2)

{

checl\_db++;

win\_cardC2=1;

}

if(check\_highP==check\_highC3&&check\_db\_double\_P==check\_db\_double\_C3)

{

checl\_db++;

win\_cardC3=1;

}

if(check\_highP==check\_highC4&&check\_db\_double\_P==check\_db\_double\_C4)

{

checl\_db++;

win\_cardC4=1;

}

if(check\_highP==check\_highC5&&check\_db\_double\_P==check\_db\_double\_C5)

{

checl\_db++;

win\_cardC5=1;

}

temp\_flhs=1;

money+=moneybox/checl\_db;

temp\_card=10;

win\_cardP=1;

}

}

}

}

}

}

}

}

if(check\_db\_double\_P>0&&check\_2db\_double\_P>=check\_2db\_double\_C1&&check\_2db\_double\_P>=check\_2db\_double\_C2&&check\_2db\_double\_P>=check\_2db\_double\_C3&&check\_2db\_double\_P>=check\_2db\_double\_C4&&check\_2db\_double\_P>=check\_2db\_double\_C5)

{

if(win\_cardP==0&&win\_cardC1==0&&win\_cardC2==0&&win\_cardC3==0&&win\_cardC4==0&&win\_cardC5==0)

{

if(check\_2db\_double\_P>check\_2db\_double\_C1&&check\_2db\_double\_P>check\_2db\_double\_C2&&check\_2db\_double\_P>check\_2db\_double\_C3&&check\_2db\_double\_P>check\_2db\_double\_C4&&check\_2db\_double\_P>check\_2db\_double\_C5)

{

money+=moneybox;

temp\_flhs=1;

db1\_temp=1;

temp\_card=10;

win\_cardP=1;

}

}

if(win\_cardP==0&&win\_cardC1==0&&win\_cardC2==0&&win\_cardC3==0&&win\_cardC4==0&&win\_cardC5==0)

{

if(check\_2db\_double\_P>0)

{

if((check\_db\_double\_P>=db2\_max&&check\_2db\_double\_P>=db\_max)||(check\_2db\_double\_P>=db2\_max&&check\_db\_double\_P>=db\_max))

{

if((check\_highP<check\_highC1&&((check\_db\_double\_C1>=db2\_max&&check\_2db\_double\_C1>=db\_max)||(check\_2db\_double\_C1>=db2\_max&&check\_db\_double\_C1>=db\_max)))||(check\_highP<check\_highC2&&((check\_db\_double\_C2>=db2\_max&&check\_2db\_double\_C2>=db\_max)||(check\_2db\_double\_C2>=db2\_max&&check\_db\_double\_C2>=db\_max)))||(check\_highP<check\_highC3&&((check\_db\_double\_C3>=db2\_max&&check\_2db\_double\_C3>=db\_max)||(check\_2db\_double\_C3>=db2\_max&&check\_db\_double\_C3>=db\_max)))||(check\_highP<check\_highC4&&((check\_db\_double\_C4>=db2\_max&&check\_2db\_double\_C4>=db\_max)||(check\_2db\_double\_C4>=db2\_max&&check\_db\_double\_C4>=db\_max)))||(check\_highP<check\_highC5&&((check\_db\_double\_C5>=db2\_max&&check\_2db\_double\_C5>=db\_max)||(check\_2db\_double\_C5>=db2\_max&&check\_db\_double\_C5>=db\_max))))

{

db2\_temp=1;

temp\_flhs=1;

temp\_card=10;

}

if(check\_highP>=check\_highC1||check\_highP>=check\_highC2||check\_highP>=check\_highC3||check\_highP>=check\_highC4||check\_highP>=check\_highC5)

{

if(db2\_temp==0)

{

if(check\_highP>check\_highC1&&check\_highP>check\_highC2&&check\_highP>check\_highC3&&check\_highP>check\_highC4&&check\_highP>check\_highC5)

{

money+=moneybox;

temp\_flhs=1;

temp\_card=10;

win\_cardP=1;

}

if(check\_highP==check\_highC1&&check\_2db\_double\_P==check\_2db\_double\_C1&&check\_db\_double\_P==check\_db\_double\_C1)

{

checl\_db++;

win\_cardC1=1;

}

if(check\_highP==check\_highC2&&check\_2db\_double\_P==check\_2db\_double\_C2&&check\_db\_double\_P==check\_db\_double\_C1)

{

checl\_db++;

win\_cardC2=1;

}

if(check\_highP==check\_highC3&&check\_2db\_double\_P==check\_2db\_double\_C3&&check\_db\_double\_P==check\_db\_double\_C1)

{

checl\_db++;

win\_cardC3=1;

}

if(check\_highP==check\_highC4&&check\_2db\_double\_P==check\_2db\_double\_C4&&check\_db\_double\_P==check\_db\_double\_C1)

{

checl\_db++;

win\_cardC4=1;

}

if(check\_highP==check\_highC5&&check\_2db\_double\_P==check\_2db\_double\_C5&&check\_db\_double\_P==check\_db\_double\_C1)

{

checl\_db++;

win\_cardC5=1;

}

temp\_flhs=1;

money+=moneybox/checl\_db;

temp\_card=10;

win\_cardP=1;

}

}

}

}

}

}

if(check\_2db\_double\_P<check\_2db\_double\_C1||check\_2db\_double\_P<check\_2db\_double\_C2||check\_2db\_double\_P<check\_2db\_double\_C3||check\_2db\_double\_P<check\_2db\_double\_C4||check\_2db\_double\_P<check\_2db\_double\_C5)

{

temp\_flhs=1;

temp\_card=10;

}

if(check\_db\_double\_P<check\_db\_double\_C1||check\_db\_double\_P<check\_db\_double\_C2||check\_db\_double\_P<check\_db\_double\_C3||check\_db\_double\_P<check\_db\_double\_C4||check\_db\_double\_P<check\_db\_double\_C5)

{

temp\_flhs=1;

temp\_card=10;

}

}

if(check\_db\_double\_P<check\_db\_double\_C1||check\_db\_double\_P<check\_db\_double\_C2||check\_db\_double\_P<check\_db\_double\_C3||check\_db\_double\_P<check\_db\_double\_C4||check\_db\_double\_P<check\_db\_double\_C5)

{

temp\_flhs=1;

temp\_card=10;

}

}

}

}

}

//-------------------HIGH-CARD---------------------

else

{

if(check\_db\_double\_P==0&&check\_db\_double\_C1==0&&check\_db\_double\_C2==0&&check\_db\_double\_C3==0&&check\_db\_double\_C4==0&&check\_db\_double\_C5==0&&check\_2db\_double\_P==0&&check\_2db\_double\_C1==0&&check\_2db\_double\_C2==0&&check\_2db\_double\_C3==0&&check\_2db\_double\_C4==0&&check\_2db\_double\_C5==0&&check\_tr\_triple\_P==0&&check\_tr\_triple\_C1==0&&check\_tr\_triple\_C2==0&&check\_tr\_triple\_C3==0&&check\_tr\_triple\_C4==0&&check\_tr\_triple\_C5==0&&check\_straighP==0&&check\_straighC1==0&&check\_straighC2==0&&check\_straighC3==0&&check\_straighC4==0&&check\_straighC5==0&&check\_flushP==0&&check\_flushC1==0&&check\_flushC2==0&&check\_flushC3==0&&check\_flushC4==0&&check\_flushC5==0&&check\_straight\_flushP==0&&check\_straight\_flushC1==0&&check\_straight\_flushC2==0&&check\_straight\_flushC3==0&&check\_straight\_flushC4==0&&check\_straight\_flushC5==0&&check\_flush\_royalP==0&&check\_flush\_royalC1==0&&check\_flush\_royalC2==0&&check\_flush\_royalC3==0&&check\_flush\_royalC4==0&&check\_flush\_royalC5==0)

{

if(check\_highP<check\_highC1||check\_highP<check\_highC2||check\_highP<check\_highC3||check\_highP<check\_highC4||check\_highP<check\_highC5)

{

temp\_flhs=1;

temp\_card=10;

hg\_temp=1;

}

if(check\_highP>=check\_highC1&&check\_highP>=check\_highC2&&check\_highP>=check\_highC3&&check\_highP>=check\_highC4&&check\_highP>=check\_highC5)

{

if(hg\_temp==0)

{

if(check\_highP>check\_highC1&&check\_highP>check\_highC2&&check\_highP>check\_highC3&&check\_highP>check\_highC4&&check\_highP>check\_highC5)

{

temp\_flhs=1;

money+=moneybox;

temp\_card=10;

win\_cardP=1;

}

if(check\_highP==check\_highC1)

{

chek\_hg++;

win\_cardC1=1;

}

if(check\_highP==check\_highC2)

{

chek\_hg++;

win\_cardC2=1;

}

if(check\_highP==check\_highC3)

{

chek\_hg++;

win\_cardC3=1;

}

if(check\_highP==check\_highC4)

{

chek\_hg++;

win\_cardC4=1;

}

if(check\_highP==check\_highC5)

{

chek\_hg++;

win\_cardC5=1;

}

temp\_flhs=1;

money+=moneybox/chek\_hg;

temp\_card=10;

win\_cardP=1;

}

}

}

}

}

}

}

}

}

}

}

temp\_card=10;

//-------------------------------------------------

check\_rfl=1;

check\_fl=1;

check\_tr=1;

checl\_db=1;

chek\_hg=1;

check\_ful=1;

check\_str=1;

temp\_flhs=0;

str\_tp=0;

db\_temp=0;

db1\_temp=0;

db2\_temp=0;

hg\_temp=0;

db\_max=0;

db2\_max=0;

check\_fh\_double\_P=0;

check\_fh\_triple\_P=0;

check\_tr\_triple\_P=0;

check\_db\_double\_P=0;

check\_2db\_double\_P=0;

check\_fh\_double\_C1=0;

check\_fh\_triple\_C1=0;

check\_tr\_triple\_C1=0;

check\_db\_double\_C1=0;

check\_2db\_double\_C1=0;

check\_fh\_double\_C2=0;

check\_fh\_triple\_C2=0;

check\_tr\_triple\_C2=0;

check\_db\_double\_C2=0;

check\_2db\_double\_C2=0;

check\_fh\_double\_C3=0;

check\_fh\_triple\_C3=0;

check\_tr\_triple\_C3=0;

check\_db\_double\_C3=0;

check\_2db\_double\_C3=0;

check\_fh\_double\_C4=0;

check\_fh\_triple\_C4=0;

check\_tr\_triple\_C4=0;

check\_db\_double\_C4=0;

check\_2db\_double\_C4=0;

check\_fh\_double\_C5=0;

check\_fh\_triple\_C5=0;

check\_tr\_triple\_C5=0;

check\_db\_double\_C5=0;

check\_2db\_double\_C5=0;

}

//--------------CARDS--------------------

void CApp::check\_card\_win(int \*check\_flush\_royalP,int \*check\_straight\_flushP,int \*check\_kareP,int \*check\_tripleP,int \*check\_triple2P,int \*check\_doubleP,int \*check\_double2P,

int \*check\_double3P,int \*check\_straighP,int \*check\_flushP,int \*check\_full\_houseP,int \*check\_highP,int \*straight\_tempP,int \*check\_fh\_double\_P,int \*check\_fh\_triple\_P,

int \*check\_tr\_triple\_P,int \*check\_db\_double\_P,int \*check\_2db\_double\_P,int \*win\_cardP,

int \*check\_flush\_royalC1,int \*check\_straight\_flushC1,int \*check\_kareC1,int \*check\_tripleC1,int \*check\_triple2C1,int \*check\_doubleC1,

int \*check\_double2C1,int \*check\_double3C1,int \*check\_straighC1,int \*check\_flushC1,int \*check\_full\_houseC1,int \*check\_highC1,int \*straight\_tempC1,int \*check\_fh\_double\_C1,

int \*check\_fh\_triple\_C1,int \*check\_tr\_triple\_C1,int \*check\_db\_double\_C1,int \*check\_2db\_double\_C1,int \*win\_cardC1,

int \*check\_flush\_royalC2,int \*check\_straight\_flushC2,int \*check\_kareC2,int \*check\_tripleC2,int \*check\_triple2C2,int \*check\_doubleC2,int \*check\_double2C2,int \*check\_double3C2,

int \*check\_straighC2,int \*check\_flushC2,int \*check\_full\_houseC2,int \*check\_highC2,int \*straight\_tempC2,int \*check\_fh\_double\_C2,int \*check\_fh\_triple\_C2,int \*check\_tr\_triple\_C2,

int \*check\_db\_double\_C2,int \*check\_2db\_double\_C2,int \*win\_cardC2,

int \*check\_flush\_royalC3,int \*check\_straight\_flushC3,int \*check\_kareC3,int \*check\_tripleC3,int \*check\_triple2C3,int \*check\_doubleC3,int \*check\_double2C3,

int \*check\_double3C3,int \*check\_straighC3,int \*check\_flushC3,int \*check\_full\_houseC3,int \*check\_highC3,int \*straight\_tempC3,int \*check\_fh\_double\_C3,int \*check\_fh\_triple\_C3,int \*check\_tr\_triple\_C3,

int \*check\_db\_double\_C3,int \*check\_2db\_double\_C3,int \*win\_cardC3,

int \*check\_flush\_royalC4,int \*check\_straight\_flushC4,int \*check\_kareC4,int \*check\_tripleC4,int \*check\_triple2C4,int \*check\_doubleC4,int \*check\_double2C4,int \*check\_double3C4,

int \*check\_straighC4,int \*check\_flushC4,int \*check\_full\_houseC4,int \*check\_highC4,int \*straight\_tempC4,int \*check\_fh\_double\_C4,int \*check\_fh\_triple\_C4,int \*check\_tr\_triple\_C4,

int \*check\_db\_double\_C4,int \*check\_2db\_double\_C4,int \*win\_cardC4,

int \*check\_flush\_royalC5,int \*check\_straight\_flushC5,int \*check\_kareC5,int \*check\_tripleC5,int \*check\_triple2C5,int \*check\_doubleC5,int \*check\_double2C5,int \*check\_double3C5,int \*check\_straighC5,

int \*check\_flushC5,int \*check\_full\_houseC5,int \*check\_highC5,int \*straight\_tempC5,int \*check\_fh\_double\_C5,int \*check\_fh\_triple\_C5,int \*check\_tr\_triple\_C5,

int \*check\_db\_double\_C5,int \*check\_2db\_double\_C5,int \*win\_cardC5)

{

int check\_rfl=1;

int check\_fl=1;

int check\_tr=1;

int checl\_db=1;

int chek\_hg=1;

int check\_str=1;

int temp\_flhs=0;

int str\_tp=0;

int db\_temp=0;

int db1\_temp=0;

int db2\_temp=0;

int hg\_temp=0;

int tp\_fl=0;

int db\_max=0;

int db2\_max=0;

if(proverkaT==0)

{

//--------------------FLUSH-ROYAL---------------------

if(\*check\_flush\_royalP==1||\*check\_flush\_royalC1==1||\*check\_flush\_royalC2==1||\*check\_flush\_royalC3==1||\*check\_flush\_royalC4==1||\*check\_flush\_royalC5==1)

{

if(\*check\_flush\_royalP>=\*check\_flush\_royalC1&&\*check\_flush\_royalP>=\*check\_flush\_royalC2&&\*check\_flush\_royalP>=\*check\_flush\_royalC3&&\*check\_flush\_royalP>=\*check\_flush\_royalC4&&\*check\_flush\_royalP>=\*check\_flush\_royalC5)

{

if(\*check\_flush\_royalC1==1)

{

check\_rfl++;

\*win\_cardC1=1;

}

if(\*check\_flush\_royalC2==1)

{

check\_rfl++;

\*win\_cardC2=1;

}

if(\*check\_flush\_royalC3==1)

{

check\_rfl++;

\*win\_cardC3=1;

}

if(\*check\_flush\_royalC4==1)

{

check\_rfl++;

\*win\_cardC4=1;

}

if(\*check\_flush\_royalC5==1)

{

check\_rfl++;

\*win\_cardC5=1;

}

temp\_flhs=1;

\*win\_cardP=1;

}

if(\*check\_flush\_royalP<\*check\_flush\_royalC1||\*check\_flush\_royalP<\*check\_flush\_royalC2||\*check\_flush\_royalP<\*check\_flush\_royalC3||\*check\_flush\_royalP<\*check\_flush\_royalC4||\*check\_flush\_royalP<\*check\_flush\_royalC5)

{

temp\_flhs=1;

}

}

//--------------------STRAIGHT-FLUSH---------------------

else

{

if(\*check\_straight\_flushP>0||\*check\_straight\_flushC1>0||\*check\_straight\_flushC2>0||\*check\_straight\_flushC3>0||\*check\_straight\_flushC4>0||\*check\_straight\_flushC5>0)

{

if(\*check\_straight\_flushP>=\*check\_straight\_flushC1&&\*check\_straight\_flushP>=\*check\_straight\_flushC2&&\*check\_straight\_flushP>=\*check\_straight\_flushC3&&\*check\_straight\_flushP>=\*check\_straight\_flushC4&&\*check\_straight\_flushP>=\*check\_straight\_flushC5)

{

if(\*check\_straight\_flushP==\*check\_straight\_flushC1&&\*check\_straight\_flushP==\*check\_straight\_flushC2&&\*check\_straight\_flushP==\*check\_straight\_flushC3&&\*check\_straight\_flushP==\*check\_straight\_flushC4&&\*check\_straight\_flushP==\*check\_straight\_flushC5)

{

temp\_flhs=1;

\*win\_cardP=1;

\*win\_cardC1=1;

\*win\_cardC2=1;

\*win\_cardC3=1;

\*win\_cardC4=1;

\*win\_cardC5=1;

}

if(\*check\_straight\_flushP>\*check\_straight\_flushC1&&\*check\_straight\_flushP>\*check\_straight\_flushC2&&\*check\_straight\_flushP>\*check\_straight\_flushC3&&\*check\_straight\_flushP>\*check\_straight\_flushC4&&\*check\_straight\_flushP>\*check\_straight\_flushC5)

{

temp\_flhs=1;

\*win\_cardP=1;

}

}

if(\*check\_straight\_flushP<\*check\_straight\_flushC1||\*check\_straight\_flushP<\*check\_straight\_flushC2||\*check\_straight\_flushP<\*check\_straight\_flushC3||\*check\_straight\_flushP<\*check\_straight\_flushC4||\*check\_straight\_flushP<\*check\_straight\_flushC5)

{

temp\_flhs=1;

}

}

//--------------------KARE-----------------------------

else

{

if(\*check\_kareP>=1||\*check\_kareC1>=1||\*check\_kareC2>=1||\*check\_kareC3>=1||\*check\_kareC4>=1||\*check\_kareC5>=1)

{

if(\*check\_kareP>=\*check\_kareC1&&\*check\_kareP>=\*check\_kareC2&&\*check\_kareP>=\*check\_kareC3&&\*check\_kareP>=\*check\_kareC4&&\*check\_kareP>=\*check\_kareC5)

{

if(\*check\_kareP==\*check\_kareC1&&\*check\_kareP==\*check\_kareC2&&\*check\_kareP==\*check\_kareC3&&\*check\_kareP==\*check\_kareC4&&\*check\_kareP==\*check\_kareC5)

{

if(\*check\_highP>\*check\_highC1&&\*check\_highP>\*check\_highC2&&\*check\_highP>\*check\_highC3&&\*check\_highP>\*check\_highC4&&\*check\_highP>\*check\_highC5)

{

temp\_flhs=1;

}

if(\*check\_highP==\*check\_highC1&&\*check\_highP==\*check\_highC2&&\*check\_highP==\*check\_highC3&&\*check\_highP==\*check\_highC4&&\*check\_highP==\*check\_highC5)

{

temp\_flhs=1;

\*win\_cardP=1;

\*win\_cardC1=1;

\*win\_cardC2=1;

\*win\_cardC3=1;

\*win\_cardC4=1;

\*win\_cardC5=1;

}

if(\*check\_highP<\*check\_highC1||\*check\_highP<\*check\_highC2||\*check\_highP<\*check\_highC3||\*check\_highP<\*check\_highC4||\*check\_highP<\*check\_highC5)

{

temp\_flhs=1;

}

}

if(\*check\_kareP>\*check\_kareC1&&\*check\_kareP>\*check\_kareC2&&\*check\_kareP>\*check\_kareC3&&\*check\_kareP>\*check\_kareC4&&\*check\_kareP>\*check\_kareC5)

{

temp\_flhs=1;

temp\_card=10;

\*win\_cardP=1;

}

}

if(\*check\_kareP<\*check\_kareC1||\*check\_kareP<\*check\_kareC2||\*check\_kareP<\*check\_kareC3||\*check\_kareP<\*check\_kareC4||\*check\_kareP<\*check\_kareC5)

{

temp\_flhs=1;

}

}

//--------------------FULL-HOUSE----------------------

else

{

if(\*check\_full\_houseP>=1||\*check\_full\_houseC1>=1||\*check\_full\_houseC2>=1||\*check\_full\_houseC3>=1||\*check\_full\_houseC4>=1||\*check\_full\_houseC5>=1)

{

//------------------------PLAYER--------------------

if(\*check\_tripleP>\*check\_triple2P)

{

\*check\_fh\_triple\_P=\*check\_tripleP;

}

if(\*check\_triple2P>\*check\_tripleP)

{

\*check\_fh\_triple\_P=\*check\_triple2P;

}

if(\*check\_doubleP==\*check\_fh\_triple\_P)

{

if(\*check\_double2P>\*check\_double3P)

{

\*check\_fh\_double\_P=\*check\_double2P;

}

if(\*check\_double3P>\*check\_double2P)

{

\*check\_fh\_double\_P=\*check\_double3P;

}

}

if(\*check\_double2P==\*check\_fh\_triple\_P)

{

if(\*check\_doubleP>\*check\_double3P)

{

\*check\_fh\_double\_P=\*check\_doubleP;

}

if(\*check\_double3P>\*check\_doubleP)

{

\*check\_fh\_double\_P=\*check\_double3P;

}

}

if(\*check\_double3P==\*check\_fh\_triple\_P)

{

if(\*check\_doubleP>\*check\_double2P)

{

\*check\_fh\_double\_P=\*check\_doubleP;

}

if(\*check\_double2P>\*check\_doubleP)

{

\*check\_fh\_double\_P=\*check\_double2P;

}

}

//------------------------COMPUTER1-----------------

if(\*check\_tripleC1>\*check\_triple2C1)

{

\*check\_fh\_triple\_C1=\*check\_tripleC1;

}

if(\*check\_triple2C1>\*check\_tripleC1)

{

\*check\_fh\_triple\_C1=\*check\_triple2C1;

}

if(\*check\_doubleC1==\*check\_fh\_triple\_C1)

{

if(\*check\_double2C1>\*check\_double3C1)

{

\*check\_fh\_double\_C1=\*check\_double2C1;

}

if(\*check\_double3C1>\*check\_double2C1)

{

\*check\_fh\_double\_C1=\*check\_double3C1;

}

}

if(\*check\_double2C1==\*check\_fh\_triple\_C1)

{

if(\*check\_doubleC1>\*check\_double3C1)

{

\*check\_fh\_double\_C1=\*check\_doubleC1;

}

if(\*check\_double3C1>\*check\_doubleC1)

{

\*check\_fh\_double\_C1=\*check\_double3C1;

}

}

if(\*check\_double3C1==\*check\_fh\_triple\_C1)

{

if(\*check\_doubleC1>\*check\_double2C1)

{

\*check\_fh\_double\_C1=\*check\_doubleC1;

}

if(\*check\_double2C1>\*check\_doubleC1)

{

\*check\_fh\_double\_C1=\*check\_double2C1;

}

}

//------------------------COMPUTER2-----------------

if(\*check\_tripleC2>\*check\_triple2C2)

{

\*check\_fh\_triple\_C2=\*check\_tripleC2;

}

if(\*check\_triple2C2>\*check\_tripleC2)

{

\*check\_fh\_triple\_C2=\*check\_triple2C2;

}

if(\*check\_doubleC2==\*check\_fh\_triple\_C2)

{

if(\*check\_double2C2>\*check\_double3C2)

{

\*check\_fh\_double\_C2=\*check\_double2C2;

}

if(\*check\_double3C2>\*check\_double2C2)

{

\*check\_fh\_double\_C2=\*check\_double3C2;

}

}

if(\*check\_double2C2==\*check\_fh\_triple\_C2)

{

if(\*check\_doubleC2>\*check\_double3C2)

{

\*check\_fh\_double\_C2=\*check\_doubleC2;

}

if(\*check\_double3C2>\*check\_doubleC2)

{

\*check\_fh\_double\_C2=\*check\_double3C2;

}

}

if(\*check\_double3C2==\*check\_fh\_triple\_C2)

{

if(\*check\_doubleC2>\*check\_double2C2)

{

\*check\_fh\_double\_C2=\*check\_doubleC2;

}

if(\*check\_double2C2>\*check\_doubleC2)

{

\*check\_fh\_double\_C2=\*check\_double2C2;

}

}

//------------------------COMPUTER3-----------------

if(\*check\_tripleC3>\*check\_triple2C3)

{

\*check\_fh\_triple\_C3=\*check\_tripleC3;

}

if(\*check\_triple2C3>\*check\_tripleC3)

{

\*check\_fh\_triple\_C3=\*check\_triple2C3;

}

if(\*check\_doubleC3==\*check\_fh\_triple\_C3)

{

if(\*check\_double2C3>\*check\_double3C3)

{

\*check\_fh\_double\_C3=\*check\_double2C3;

}

if(\*check\_double3C3>\*check\_double2C3)

{

\*check\_fh\_double\_C3=\*check\_double3C3;

}

}

if(\*check\_double2C3==\*check\_fh\_triple\_C3)

{

if(\*check\_doubleC3>\*check\_double3C3)

{

\*check\_fh\_double\_C3=\*check\_doubleC3;

}

if(\*check\_double3C3>\*check\_doubleC3)

{

\*check\_fh\_double\_C3=\*check\_double3C3;

}

}

if(\*check\_double3C3==\*check\_fh\_triple\_C3)

{

if(\*check\_doubleC3>\*check\_double2C3)

{

\*check\_fh\_double\_C3=\*check\_doubleC3;

}

if(\*check\_double2C3>\*check\_doubleC3)

{

\*check\_fh\_double\_C3=\*check\_double2C3;

}

}

//------------------------COMPUTER4-----------------

if(\*check\_tripleC4>\*check\_triple2C4)

{

\*check\_fh\_triple\_C4=\*check\_tripleC4;

}

if(\*check\_triple2C4>\*check\_tripleC4)

{

\*check\_fh\_triple\_C4=\*check\_triple2C4;

}

if(\*check\_doubleC4==\*check\_fh\_triple\_C4)

{

if(\*check\_double2C4>\*check\_double3C4)

{

\*check\_fh\_double\_C4=\*check\_double2C4;

}

if(\*check\_double3C4>\*check\_double2C4)

{

\*check\_fh\_double\_C4=\*check\_double3C4;

}

}

if(\*check\_double2C4==\*check\_fh\_triple\_C4)

{

if(\*check\_doubleC4>\*check\_double3C4)

{

\*check\_fh\_double\_C4=\*check\_doubleC4;

}

if(\*check\_double3C4>\*check\_doubleC4)

{

\*check\_fh\_double\_C4=\*check\_double3C4;

}

}

if(\*check\_double3C4==\*check\_fh\_triple\_C4)

{

if(\*check\_doubleC4>\*check\_double2C4)

{

\*check\_fh\_double\_C4=\*check\_doubleC4;

}

if(\*check\_double2C4>\*check\_doubleC4)

{

\*check\_fh\_double\_C4=\*check\_double2C4;

}

}

//------------------------COMPUTER5-----------------

if(\*check\_tripleC5>\*check\_triple2C5)

{

\*check\_fh\_triple\_C5=\*check\_tripleC5;

}

if(\*check\_triple2C5>\*check\_tripleC5)

{

\*check\_fh\_triple\_C5=\*check\_triple2C5;

}

if(\*check\_doubleC5==\*check\_fh\_triple\_C5)

{

if(\*check\_double2C5>\*check\_double3C5)

{

\*check\_fh\_double\_C5=\*check\_double2C5;

}

if(\*check\_double3C5>\*check\_double2C5)

{

\*check\_fh\_double\_C5=\*check\_double3C5;

}

}

if(\*check\_double2C5==\*check\_fh\_triple\_C5)

{

if(\*check\_doubleC5>\*check\_double3C5)

{

\*check\_fh\_double\_C5=\*check\_doubleC5;

}

if(\*check\_double3C5>\*check\_doubleC5)

{

\*check\_fh\_double\_C5=\*check\_double3C5;

}

}

if(\*check\_double3C5==\*check\_fh\_triple\_C5)

{

if(\*check\_doubleC5>\*check\_double2C5)

{

\*check\_fh\_double\_C5=\*check\_doubleC5;

}

if(\*check\_double2C5>\*check\_doubleC5)

{

\*check\_fh\_double\_C5=\*check\_double2C5;

}

}

//-------------------------------------------------

if(\*check\_full\_houseP>=\*check\_full\_houseC1&&\*check\_full\_houseP>=\*check\_full\_houseC2&&\*check\_full\_houseP>=\*check\_full\_houseC3&&\*check\_full\_houseP>=\*check\_full\_houseC4&&\*check\_full\_houseP>=\*check\_full\_houseC5)

{

if(\*check\_fh\_triple\_P>\*check\_fh\_triple\_C1&&\*check\_fh\_triple\_P>\*check\_fh\_triple\_C2&&\*check\_fh\_triple\_P>\*check\_fh\_triple\_C3&&\*check\_fh\_triple\_P>\*check\_fh\_triple\_C4&&\*check\_fh\_triple\_P>\*check\_fh\_triple\_C5)

{

temp\_flhs=1;

\*win\_cardP=1;

tp\_fl=1;

}

if(\*check\_fh\_triple\_P==\*check\_fh\_triple\_C1||\*check\_fh\_triple\_P==\*check\_fh\_triple\_C2||\*check\_fh\_triple\_P==\*check\_fh\_triple\_C3||\*check\_fh\_triple\_P||\*check\_fh\_triple\_C4||\*check\_fh\_triple\_P||\*check\_fh\_triple\_C5)

{

if(\*check\_fh\_double\_P>=\*check\_fh\_double\_C1&&\*check\_fh\_double\_P>=\*check\_fh\_double\_C2&&\*check\_fh\_double\_P>=\*check\_fh\_double\_C3&&\*check\_fh\_double\_P>=\*check\_fh\_double\_C4&&\*check\_fh\_double\_P>=\*check\_fh\_double\_C5)

{

if(\*check\_fh\_double\_P>\*check\_fh\_double\_C1&&\*check\_fh\_double\_P>\*check\_fh\_double\_C2&&\*check\_fh\_double\_P>\*check\_fh\_double\_C3&&\*check\_fh\_double\_P>\*check\_fh\_double\_C4&&\*check\_fh\_double\_P>\*check\_fh\_double\_C5)

{

temp\_flhs=1;

\*win\_cardP=1;

tp\_fl=1;

}

if(\*win\_cardP==0)

{

if(\*check\_fh\_double\_P<\*check\_fh\_double\_C1||\*check\_fh\_double\_P<\*check\_fh\_double\_C2||\*check\_fh\_double\_P<\*check\_fh\_double\_C3||\*check\_fh\_double\_P<\*check\_fh\_double\_C4||\*check\_fh\_double\_P<\*check\_fh\_double\_C5)

{

temp\_flhs=1;

}

if(\*check\_fh\_double\_P==\*check\_fh\_double\_C1&&\*check\_fh\_triple\_P==\*check\_fh\_triple\_C1)

{

\*win\_cardC1=1;

}

if(\*check\_fh\_double\_P==\*check\_fh\_double\_C2&&\*check\_fh\_triple\_P==\*check\_fh\_triple\_C2)

{

\*win\_cardC2=1;

}

if(\*check\_fh\_double\_P==\*check\_fh\_double\_C3&&\*check\_fh\_triple\_P==\*check\_fh\_triple\_C3)

{

\*win\_cardC3=1;

}

if(\*check\_fh\_double\_P==\*check\_fh\_double\_C4&&\*check\_fh\_triple\_P==\*check\_fh\_triple\_C4)

{

\*win\_cardC4=1;

}

if(\*check\_fh\_double\_P==\*check\_fh\_double\_C5&&\*check\_fh\_triple\_P==\*check\_fh\_triple\_C5)

{

\*win\_cardC5=1;

}

temp\_flhs=1;

\*win\_cardP=1;

tp\_fl=1;

}

}

if(\*check\_fh\_triple\_P<\*check\_fh\_triple\_C1||\*check\_fh\_triple\_P<\*check\_fh\_triple\_C2||\*check\_fh\_triple\_P<\*check\_fh\_triple\_C3||\*check\_fh\_triple\_P<\*check\_fh\_triple\_C4||\*check\_fh\_triple\_P<\*check\_fh\_triple\_C5)

{

temp\_flhs=1;

}

}

if(\*check\_full\_houseP>\*check\_full\_houseC1&&\*check\_full\_houseP>\*check\_full\_houseC2&&\*check\_full\_houseP>\*check\_full\_houseC3&&\*check\_full\_houseP>\*check\_full\_houseC4&&\*check\_full\_houseP>\*check\_full\_houseC5)

{

temp\_flhs=1;

\*win\_cardP=1;

tp\_fl=1;

}

if(\*check\_full\_houseP<\*check\_full\_houseC1||\*check\_full\_houseP<\*check\_full\_houseC2||\*check\_full\_houseP<\*check\_full\_houseC3||\*check\_full\_houseP<\*check\_full\_houseC4||\*check\_full\_houseP<\*check\_full\_houseC5)

{

temp\_flhs=1;

}

}

if(\*check\_full\_houseP<\*check\_full\_houseC1||\*check\_full\_houseP<\*check\_full\_houseC2||\*check\_full\_houseP<\*check\_full\_houseC3||\*check\_full\_houseP<\*check\_full\_houseC4||\*check\_full\_houseP<\*check\_full\_houseC5)

{

temp\_flhs=1;

}

}

//--------------------FLUSH-------------------------

else

{

if(\*check\_flushP>=1||\*check\_flushC1>=1||\*check\_flushC2>=1||\*check\_flushC3>=1||\*check\_flushC4>=1||\*check\_flushC5>=1)

{

if(\*check\_flushP>=\*check\_flushC1&&\*check\_flushP>=\*check\_flushC2&&\*check\_flushP>=\*check\_flushC3&&\*check\_flushP>=\*check\_flushC4&&\*check\_flushP>=\*check\_flushC5)

{

if(\*check\_flushC1==1)

{

check\_fl++;

\*win\_cardC1=1;

}

if(\*check\_flushC2==1)

{

check\_fl++;

\*win\_cardC2=1;

}

if(\*check\_flushC3==1)

{

check\_fl++;

\*win\_cardC3=1;

}

if(\*check\_flushC4==1)

{

check\_fl++;

\*win\_cardC4=1;

}

if(\*check\_flushC5==1)

{

check\_fl++;

\*win\_cardC5=1;

}

temp\_flhs=1;

\*win\_cardP=1;

}

if(\*check\_flushP<\*check\_flushC1||\*check\_flushP<\*check\_flushC2||\*check\_flushP<\*check\_flushC3||\*check\_flushP<\*check\_flushC4||\*check\_flushP<\*check\_flushC5)

{

temp\_flhs=1;

}

}

//--------------------STRAIGHT-----------------------

else

{

if(\*check\_straighP>=1||\*check\_straighC1>=1||\*check\_straighC2>=1||\*check\_straighC3>=1||\*check\_straighC4>=1||\*check\_straighC5>=1)

{

if(\*check\_straighP>=\*check\_straighC1&&\*check\_straighP>=\*check\_straighC2&&\*check\_straighP>=\*check\_straighC3&&\*check\_straighP>=\*check\_straighC4&&\*check\_straighP>=\*check\_straighC5)

{

if(\*check\_straighP==\*check\_straighC1||\*check\_straighP==\*check\_straighC2||\*check\_straighP==\*check\_straighC3||\*check\_straighP==\*check\_straighC4||\*check\_straighP||\*check\_straighC5)

{

if(\*check\_straighP==\*check\_straighC1)

{

check\_str++;

\*win\_cardC1=1;

}

if(\*check\_straighP==\*check\_straighC2)

{

check\_str++;

\*win\_cardC2=1;

}

if(\*check\_straighP==\*check\_straighC3)

{

check\_str++;

\*win\_cardC3=1;

}

if(\*check\_straighP==\*check\_straighC4)

{

check\_str++;

\*win\_cardC4=1;

}

if(\*check\_straighP==\*check\_straighC5)

{

check\_str++;

\*win\_cardC5=1;

}

temp\_flhs=1;

str\_tp=1;

\*win\_cardP=1;

}

if(str\_tp==0)

{

if(\*check\_straighP>\*check\_straighC1&&\*check\_straighP>\*check\_straighC2&&\*check\_straighP>\*check\_straighC3&&\*check\_straighP>\*check\_straighC4&&\*check\_straighP>\*check\_straighC5)

{

temp\_flhs=1;

\*win\_cardP=1;

}

}

}

if(\*check\_straighP<\*check\_straighC1||\*check\_straighP<\*check\_straighC2||\*check\_straighP<\*check\_straighC3||\*check\_straighP<\*check\_straighC4||\*check\_straighP<\*check\_straighC5)

{

temp\_flhs=1;

}

}

//--------------------TRIPLE-----------------------

else

{

if(\*check\_tripleP>0||\*check\_tripleC1>0||\*check\_tripleC2>0||\*check\_tripleC3>0||\*check\_tripleC4>0||\*check\_tripleC5>0)

{

if(temp\_flhs==0)

{

//-------------------PLAYER------------------------

if(\*check\_tripleP>\*check\_triple2P)

{

\*check\_tr\_triple\_P=\*check\_tripleP;

}

if(\*check\_triple2P>\*check\_tripleP)

{

\*check\_tr\_triple\_P=\*check\_triple2P;

}

//-------------------COMPUTER1---------------------

if(\*check\_tripleC1>\*check\_triple2C1)

{

\*check\_tr\_triple\_C1=\*check\_tripleC1;

}

if(\*check\_triple2C1>\*check\_tripleC1)

{

\*check\_tr\_triple\_C1=\*check\_triple2C1;

}

//-------------------COMPUTER2---------------------

if(\*check\_tripleC2>\*check\_triple2C2)

{

\*check\_tr\_triple\_C2=\*check\_tripleC2;

}

if(\*check\_triple2C2>\*check\_tripleC2)

{

\*check\_tr\_triple\_C2=\*check\_triple2C2;

}

//-------------------COMPUTER3---------------------

if(\*check\_tripleC3>\*check\_triple2C3)

{

\*check\_tr\_triple\_C3=\*check\_tripleC3;

}

if(\*check\_triple2C3>\*check\_tripleC3)

{

\*check\_tr\_triple\_C3=\*check\_triple2C3;

}

//-------------------COMPUTER4---------------------

if(\*check\_tripleC4>\*check\_triple2C4)

{

\*check\_tr\_triple\_C4=\*check\_tripleC4;

}

if(\*check\_triple2C4>\*check\_tripleC4)

{

\*check\_tr\_triple\_C4=\*check\_triple2C4;

}

//-------------------COMPUTER5---------------------

if(\*check\_tripleC5>\*check\_triple2C5)

{

\*check\_tr\_triple\_C5=\*check\_tripleC5;

}

if(\*check\_triple2C5>\*check\_tripleC5)

{

\*check\_tr\_triple\_C5=\*check\_triple2C5;

}

//-------------------------------------------------

if(\*check\_tr\_triple\_P>=\*check\_tr\_triple\_C1&&\*check\_tr\_triple\_P>=\*check\_tr\_triple\_C2&&\*check\_tr\_triple\_P>=\*check\_tr\_triple\_C3&&\*check\_tr\_triple\_P>=\*check\_tr\_triple\_C4&&\*check\_tr\_triple\_P>=\*check\_tr\_triple\_C5)

{

if(\*check\_tr\_triple\_P>\*check\_tr\_triple\_C1&&\*check\_tr\_triple\_P>\*check\_tr\_triple\_C2&&\*check\_tr\_triple\_P>\*check\_tr\_triple\_C3&&\*check\_tr\_triple\_P>\*check\_tr\_triple\_C4&&\*check\_tr\_triple\_P>\*check\_tr\_triple\_C5)

{

temp\_flhs=1;

\*win\_cardP=1;

}

if(\*check\_tr\_triple\_P==\*check\_tr\_triple\_C1||\*check\_tr\_triple\_P==\*check\_tr\_triple\_C2||\*check\_tr\_triple\_P==\*check\_tr\_triple\_C3||\*check\_tr\_triple\_P==\*check\_tr\_triple\_C4||\*check\_tr\_triple\_P==\*check\_tr\_triple\_C5)

{

if((\*check\_highP>=\*check\_highC1&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C1)||(\*check\_highP>=\*check\_highC2&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C2)||(\*check\_highP>=\*check\_highC3&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C3)||(\*check\_highP>=\*check\_highC4&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C4)||(\*check\_highP>=\*check\_highC5&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C5))

{

if((\*check\_highP>\*check\_highC1&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C1)&&(\*check\_highP>\*check\_highC2&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C2)&&(\*check\_highP>\*check\_highC3&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C3)&&(\*check\_highP>\*check\_highC4&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C4)&&(\*check\_highP>\*check\_highC5&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C5))

{

temp\_flhs=1;

\*win\_cardP=1;

}

if(\*check\_highP==\*check\_highC1&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C1)

{

check\_tr++;

\*win\_cardC1=1;

}

if(\*check\_highP==\*check\_highC2&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C2)

{

check\_tr++;

\*win\_cardC2=1;

}

if(\*check\_highP==\*check\_highC3&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C3)

{

check\_tr++;

\*win\_cardC3=1;

}

if(\*check\_highP==\*check\_highC4&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C4)

{

check\_tr++;

\*win\_cardC4=1;

}

if(\*check\_highP==\*check\_highC5&&\*check\_tr\_triple\_P==\*check\_tr\_triple\_C5)

{

check\_tr++;

\*win\_cardC5=1;

}

temp\_flhs=1;

\*win\_cardP=1;

}

}

}

if(\*check\_tr\_triple\_P<\*check\_tr\_triple\_C1||\*check\_tr\_triple\_P<\*check\_tr\_triple\_C2||\*check\_tr\_triple\_P<\*check\_tr\_triple\_C3||\*check\_tr\_triple\_P<\*check\_tr\_triple\_C4||\*check\_tr\_triple\_P<\*check\_tr\_triple\_C5)

{

temp\_flhs=1;

}

}

}

//--------------------DOUBLE-----------------------

else

{

if(\*check\_doubleP>0||\*check\_doubleC1>0||\*check\_doubleC2>0||\*check\_doubleC3>0||\*check\_doubleC4>0||\*check\_doubleC5>0)

{

if(tp\_fl==0)

{

if(temp\_flhs==0)

{

//-------------------PLAYER------------------------

if(\*check\_doubleP>\*check\_double2P&&\*check\_doubleP>\*check\_double3P)

{

\*check\_db\_double\_P=\*check\_doubleP;

if(\*check\_double2P>\*check\_double3P)

{

\*check\_2db\_double\_P=\*check\_double2P;

}

if(\*check\_double3P>\*check\_double2P)

{

\*check\_2db\_double\_P=\*check\_double3P;

}

}

if(\*check\_double2P>\*check\_doubleP&&\*check\_double2P>\*check\_double3P)

{

\*check\_db\_double\_P=\*check\_double2P;

if(\*check\_doubleP>\*check\_double3P)

{

\*check\_2db\_double\_P=\*check\_doubleP;

}

if(\*check\_double3P>\*check\_doubleP)

{

\*check\_2db\_double\_P=\*check\_double3P;

}

}

if(\*check\_double3P>\*check\_doubleP&&\*check\_double3P>\*check\_double2P)

{

\*check\_db\_double\_P=\*check\_double3P;

if(\*check\_doubleP>\*check\_double2P)

{

\*check\_2db\_double\_P=\*check\_doubleP;

}

if(\*check\_double2P>\*check\_doubleP)

{

\*check\_2db\_double\_P=\*check\_double2P;

}

}

//-------------------COMPUTER1---------------------

if(\*check\_doubleC1>\*check\_double2C1&&\*check\_doubleC1>\*check\_double3C1)

{

\*check\_db\_double\_C1=\*check\_doubleC1;

if(\*check\_double2C1>\*check\_double3C1)

{

\*check\_2db\_double\_C1=\*check\_double2C1;

}

if(\*check\_double3C1>\*check\_double2C1)

{

\*check\_2db\_double\_C1=\*check\_double3C1;

}

}

if(\*check\_double2C1>\*check\_doubleC1&&\*check\_double2C1>\*check\_double3C1)

{

\*check\_db\_double\_C1=\*check\_double2C1;

if(\*check\_doubleC1>\*check\_double3C1)

{

\*check\_2db\_double\_C1=\*check\_doubleC1;

}

if(\*check\_double3C1>\*check\_doubleC1)

{

\*check\_2db\_double\_C1=\*check\_double3C1;

}

}

if(\*check\_double3C1>\*check\_doubleC1&&\*check\_double3C1>\*check\_double2C1)

{

\*check\_db\_double\_C1=\*check\_double3C1;

if(\*check\_doubleC1>\*check\_double2C1)

{

\*check\_2db\_double\_C1=\*check\_doubleC1;

}

if(\*check\_double2C1>\*check\_doubleC1)

{

\*check\_2db\_double\_C1=\*check\_double2C1;

}

}

//-------------------COMPUTER2---------------------

if(\*check\_doubleC2>\*check\_double2C2&&\*check\_doubleC2>\*check\_double3C2)

{

\*check\_db\_double\_C2=\*check\_doubleC2;

if(\*check\_double2C2>\*check\_double3C2)

{

\*check\_2db\_double\_C2=\*check\_double2C2;

}

if(\*check\_double3C2>\*check\_double2C2)

{

\*check\_2db\_double\_C2=\*check\_double3C2;

}

}

if(\*check\_double2C2>\*check\_doubleC2&&\*check\_double2C2>\*check\_double3C2)

{

\*check\_db\_double\_C2=\*check\_double2C2;

if(\*check\_doubleC2>\*check\_double3C2)

{

\*check\_2db\_double\_C2=\*check\_doubleC2;

}

if(\*check\_double3C2>\*check\_doubleC2)

{

\*check\_2db\_double\_C2=\*check\_double3C2;

}

}

if(\*check\_double3C2>\*check\_doubleC2&&\*check\_double3C2>\*check\_double2C2)

{

\*check\_db\_double\_C2=\*check\_double3C2;

if(\*check\_doubleC2>\*check\_double2C2)

{

\*check\_2db\_double\_C2=\*check\_doubleC2;

}

if(\*check\_double2C2>\*check\_doubleC2)

{

\*check\_2db\_double\_C2=\*check\_double2C2;

}

}

//-------------------COMPUTER3---------------------

if(\*check\_doubleC3>\*check\_double2C3&&\*check\_doubleC3>\*check\_double3C3)

{

\*check\_db\_double\_C3=\*check\_doubleC3;

if(\*check\_double2C3>\*check\_double3C3)

{

\*check\_2db\_double\_C3=\*check\_double2C3;

}

if(\*check\_double3C3>\*check\_double2C3)

{

\*check\_2db\_double\_C3=\*check\_double3C3;

}

}

if(\*check\_double2C3>\*check\_doubleC3&&\*check\_double2C3>\*check\_double3C3)

{

\*check\_db\_double\_C3=\*check\_double2C3;

if(\*check\_doubleC3>\*check\_double3C3)

{

\*check\_2db\_double\_C3=\*check\_doubleC3;

}

if(\*check\_double3C3>\*check\_doubleC3)

{

\*check\_2db\_double\_C3=\*check\_double3C3;

}

}

if(\*check\_double3C3>\*check\_doubleC3&&\*check\_double3C3>\*check\_double2C3)

{

\*check\_db\_double\_C3=\*check\_double3C3;

if(\*check\_doubleC3>\*check\_double2C3)

{

\*check\_2db\_double\_C3=\*check\_doubleC3;

}

if(\*check\_double2C3>\*check\_doubleC3)

{

\*check\_2db\_double\_C3=\*check\_double2C3;

}

}

//-------------------COMPUTER4---------------------

if(\*check\_doubleC4>\*check\_double2C4&&\*check\_doubleC4>\*check\_double3C4)

{

\*check\_db\_double\_C4=\*check\_doubleC4;

if(\*check\_double2C4>\*check\_double3C4)

{

\*check\_2db\_double\_C4=\*check\_double2C4;

}

if(\*check\_double3C4>\*check\_double2C4)

{

\*check\_2db\_double\_C4=\*check\_double3C4;

}

}

if(\*check\_double2C4>\*check\_doubleC4&&\*check\_double2C4>\*check\_double3C4)

{

\*check\_db\_double\_C4=\*check\_double2C4;

if(\*check\_doubleC4>\*check\_double3C4)

{

\*check\_2db\_double\_C4=\*check\_doubleC4;

}

if(\*check\_double3C4>\*check\_doubleC4)

{

\*check\_2db\_double\_C4=\*check\_double3C4;

}

}

if(\*check\_double3C4>\*check\_doubleC4&&\*check\_double3C4>\*check\_double2C4)

{

\*check\_db\_double\_C4=\*check\_double3C4;

if(\*check\_doubleC4>\*check\_double2C4)

{

\*check\_2db\_double\_C4=\*check\_doubleC4;

}

if(\*check\_double2C4>\*check\_doubleC4)

{

\*check\_2db\_double\_C4=\*check\_double2C4;

}

}

//-------------------COMPUTER5---------------------

if(\*check\_doubleC5>\*check\_double2C5&&\*check\_doubleC5>\*check\_double3C5)

{

\*check\_db\_double\_C5=\*check\_doubleC5;

if(\*check\_double2C5>\*check\_double3C5)

{

\*check\_2db\_double\_C5=\*check\_double2C5;

}

if(\*check\_double3C5>\*check\_double2C5)

{

\*check\_2db\_double\_C5=\*check\_double3C5;

}

}

if(\*check\_double2C5>\*check\_doubleC5&&\*check\_double2C5>\*check\_double3C5)

{

\*check\_db\_double\_C5=\*check\_double2C5;

if(\*check\_doubleC5>\*check\_double3C5)

{

\*check\_2db\_double\_C5=\*check\_doubleC5;

}

if(\*check\_double3C5>\*check\_doubleC5)

{

\*check\_2db\_double\_C5=\*check\_double3C5;

}

}

if(\*check\_double3C5>\*check\_doubleC5&&\*check\_double3C5>\*check\_double2C5)

{

\*check\_db\_double\_C5=\*check\_double3C5;

if(\*check\_doubleC5>\*check\_double2C5)

{

\*check\_2db\_double\_C5=\*check\_doubleC5;

}

if(\*check\_double2C5>\*check\_doubleC5)

{

\*check\_2db\_double\_C5=\*check\_double2C5;

}

}

//-------------------------------------------------

if(\*check\_db\_double\_P>\*check\_db\_double\_C1)

{

if(db\_max<\*check\_db\_double\_P)db\_max=\*check\_db\_double\_P;

}

else if(db\_max<\*check\_db\_double\_C1)db\_max=\*check\_db\_double\_C1;

if(\*check\_db\_double\_P>\*check\_db\_double\_C2)

{

if(db\_max<\*check\_db\_double\_P)db\_max=\*check\_db\_double\_P;

}

else if(db\_max<\*check\_db\_double\_C2)db\_max=\*check\_db\_double\_C2;

if(\*check\_db\_double\_P>\*check\_db\_double\_C3)

{

if(db\_max<\*check\_db\_double\_P)db\_max=\*check\_db\_double\_P;

}

else if(db\_max<\*check\_db\_double\_C3)db\_max=\*check\_db\_double\_C3;

if(\*check\_db\_double\_P>\*check\_db\_double\_C4)

{

if(db\_max<\*check\_db\_double\_P)db\_max=\*check\_db\_double\_P;

}

else if(db\_max<\*check\_db\_double\_C4)db\_max=\*check\_db\_double\_C4;

if(\*check\_db\_double\_P>\*check\_db\_double\_C5)

{

if(db\_max<\*check\_db\_double\_P)db\_max=\*check\_db\_double\_P;

}

else if(db\_max<\*check\_db\_double\_C5)db\_max=\*check\_db\_double\_C5;

//-------------------------------------------------

if(\*check\_2db\_double\_P>\*check\_2db\_double\_C1)

{

if(db2\_max<\*check\_2db\_double\_P)db2\_max=\*check\_2db\_double\_P;

}

else if(db2\_max<\*check\_2db\_double\_C1)db2\_max=\*check\_2db\_double\_C1;

if(\*check\_2db\_double\_P>\*check\_2db\_double\_C2)

{

if(db2\_max<\*check\_2db\_double\_P)db2\_max=\*check\_2db\_double\_P;

}

else if(db2\_max<\*check\_2db\_double\_C2)db2\_max=\*check\_2db\_double\_C2;

if(\*check\_2db\_double\_P>\*check\_2db\_double\_C3)

{

if(db2\_max<\*check\_2db\_double\_P)db2\_max=\*check\_2db\_double\_P;

}

else if(db2\_max<\*check\_2db\_double\_C3)db2\_max=\*check\_2db\_double\_C3;

if(\*check\_2db\_double\_P>\*check\_2db\_double\_C4)

{

if(db2\_max<\*check\_2db\_double\_P)db2\_max=\*check\_2db\_double\_P;

}

else if(db2\_max<\*check\_2db\_double\_C4)db2\_max=\*check\_2db\_double\_C4;

if(\*check\_2db\_double\_P>\*check\_2db\_double\_C5)

{

if(db2\_max<\*check\_2db\_double\_P)db2\_max=\*check\_2db\_double\_P;

}

else if(db2\_max<\*check\_2db\_double\_C5)db2\_max=\*check\_2db\_double\_C5;

//-------------------------------------------------

if((\*check\_db\_double\_P>=\*check\_db\_double\_C1&&\*check\_db\_double\_P>=\*check\_db\_double\_C2&&\*check\_db\_double\_P>=\*check\_db\_double\_C3&&\*check\_db\_double\_P>=\*check\_db\_double\_C4&&\*check\_db\_double\_P>=\*check\_db\_double\_C5)||(\*check\_2db\_double\_P>=\*check\_2db\_double\_C1&&\*check\_2db\_double\_P>=\*check\_2db\_double\_C2&&\*check\_2db\_double\_P>=\*check\_2db\_double\_C3&&\*check\_2db\_double\_P>=\*check\_2db\_double\_C4&&\*check\_2db\_double\_P>=\*check\_2db\_double\_C5))

{

if(\*check\_2db\_double\_P>0&&\*check\_2db\_double\_C1==0&&\*check\_2db\_double\_C2==0&&\*check\_2db\_double\_C3==0&&\*check\_2db\_double\_C4==0&&\*check\_2db\_double\_C5==0)

{

temp\_flhs=1;

\*win\_cardP=1;

}

if(\*check\_db\_double\_P>\*check\_db\_double\_C1&&\*check\_2db\_double\_P>0&&\*check\_db\_double\_P>\*check\_db\_double\_C2&&\*check\_2db\_double\_P>0&&\*check\_db\_double\_P>\*check\_db\_double\_C3&&\*check\_2db\_double\_P>0&&\*check\_db\_double\_P>\*check\_db\_double\_C4&&\*check\_2db\_double\_P>0&&\*check\_db\_double\_P>\*check\_db\_double\_C5&&\*check\_2db\_double\_P>0)

{

temp\_flhs=1;

\*win\_cardP=1;

}

if(\*check\_2db\_double\_P==0&&\*check\_2db\_double\_C1==0&&\*check\_2db\_double\_C2==0&&\*check\_2db\_double\_C3==0&&\*check\_2db\_double\_C4==0&&\*check\_2db\_double\_C5==0)

{

if(\*check\_db\_double\_P>\*check\_db\_double\_C1&&\*check\_db\_double\_P>\*check\_db\_double\_C2&&\*check\_db\_double\_P>\*check\_db\_double\_C3&&\*check\_db\_double\_P>\*check\_db\_double\_C4&&\*check\_db\_double\_P>\*check\_db\_double\_C5)

{

temp\_flhs=1;

\*win\_cardP=1;

}

if(\*check\_highP>=\*check\_highC1||\*check\_highP>=\*check\_highC2||\*check\_highP>=\*check\_highC3||\*check\_highP>=\*check\_highC4||\*check\_highP>=\*check\_highC5)

{

if(\*check\_db\_double\_P==0&&\*check\_2db\_double\_P==0)

{

db\_temp=1;

}

if((\*check\_highP<\*check\_highC1&&((\*check\_db\_double\_C1>=db2\_max&&\*check\_2db\_double\_C1>=db\_max)||(\*check\_2db\_double\_C1>=db2\_max&&\*check\_db\_double\_C1>=db\_max)))||(\*check\_highP<\*check\_highC2&&((\*check\_db\_double\_C2>=db2\_max&&\*check\_2db\_double\_C2>=db\_max)||(\*check\_2db\_double\_C2>=db2\_max&&\*check\_db\_double\_C2>=db\_max)))||(\*check\_highP<\*check\_highC3&&((\*check\_db\_double\_C3>=db2\_max&&\*check\_2db\_double\_C3>=db\_max)||(\*check\_2db\_double\_C3>=db2\_max&&\*check\_db\_double\_C3>=db\_max)))||(\*check\_highP<\*check\_highC4&&((\*check\_db\_double\_C4>=db2\_max&&\*check\_2db\_double\_C4>=db\_max)||(\*check\_2db\_double\_C4>=db2\_max&&\*check\_db\_double\_C4>=db\_max)))||(\*check\_highP<\*check\_highC5&&((\*check\_db\_double\_C5>=db2\_max&&\*check\_2db\_double\_C5>=db\_max)||(\*check\_2db\_double\_C5>=db2\_max&&\*check\_db\_double\_C5>=db\_max))))

{

db\_temp=1;

temp\_flhs=1;

}

if((\*check\_db\_double\_P>=db2\_max&&\*check\_2db\_double\_P>=db\_max)||(\*check\_2db\_double\_P>=db2\_max&&\*check\_db\_double\_P>=db\_max))

{

if(db\_temp==0)

{

if(\*check\_highP>\*check\_highC1&&\*check\_highP>\*check\_highC2&&\*check\_highP>\*check\_highC3&&\*check\_highP>\*check\_highC4&&\*check\_highP>\*check\_highC5)

{

temp\_flhs=1;

}

if(\*check\_highP==\*check\_highC1&&\*check\_db\_double\_P==\*check\_db\_double\_C1)

{

checl\_db++;

\*win\_cardC1=1;

}

if(\*check\_highP==\*check\_highC2&&\*check\_db\_double\_P==\*check\_db\_double\_C2)

{

checl\_db++;

\*win\_cardC2=1;

}

if(\*check\_highP==\*check\_highC3&&\*check\_db\_double\_P==\*check\_db\_double\_C3)

{

checl\_db++;

\*win\_cardC3=1;

}

if(\*check\_highP==\*check\_highC4&&\*check\_db\_double\_P==\*check\_db\_double\_C4)

{

checl\_db++;

\*win\_cardC4=1;

}

if(\*check\_highP==\*check\_highC5&&\*check\_db\_double\_P==\*check\_db\_double\_C5)

{

checl\_db++;

\*win\_cardC5=1;

}

temp\_flhs=1;

\*win\_cardP=1;

}

}

}

}

if(\*check\_db\_double\_P>0&&\*check\_2db\_double\_P>=\*check\_2db\_double\_C1&&\*check\_2db\_double\_P>=\*check\_2db\_double\_C2&&\*check\_2db\_double\_P>=\*check\_2db\_double\_C3&&\*check\_2db\_double\_P>=\*check\_2db\_double\_C4&&\*check\_2db\_double\_P>=\*check\_2db\_double\_C5)

{

if(\*check\_2db\_double\_P>0)

{

if((\*check\_db\_double\_P>=db2\_max&&\*check\_2db\_double\_P>=db\_max)||(\*check\_2db\_double\_P>=db2\_max&&\*check\_db\_double\_P>=db\_max))

{

if((\*check\_highP<\*check\_highC1&&((\*check\_db\_double\_C1>=db2\_max&&\*check\_2db\_double\_C1>=db\_max)||(\*check\_2db\_double\_C1>=db2\_max&&\*check\_db\_double\_C1>=db\_max)))||(\*check\_highP<\*check\_highC2&&((\*check\_db\_double\_C2>=db2\_max&&\*check\_2db\_double\_C2>=db\_max)||(\*check\_2db\_double\_C2>=db2\_max&&\*check\_db\_double\_C2>=db\_max)))||(\*check\_highP<\*check\_highC3&&((\*check\_db\_double\_C3>=db2\_max&&\*check\_2db\_double\_C3>=db\_max)||(\*check\_2db\_double\_C3>=db2\_max&&\*check\_db\_double\_C3>=db\_max)))||(\*check\_highP<\*check\_highC4&&((\*check\_db\_double\_C4>=db2\_max&&\*check\_2db\_double\_C4>=db\_max)||(\*check\_2db\_double\_C4>=db2\_max&&\*check\_db\_double\_C4>=db\_max)))||(\*check\_highP<\*check\_highC5&&((\*check\_db\_double\_C5>=db2\_max&&\*check\_2db\_double\_C5>=db\_max)||(\*check\_2db\_double\_C5>=db2\_max&&\*check\_db\_double\_C5>=db\_max))))

{

db2\_temp=1;

temp\_flhs=1;

}

if(\*check\_2db\_double\_P>\*check\_2db\_double\_C1&&\*check\_2db\_double\_P>\*check\_2db\_double\_C2&&\*check\_2db\_double\_P>\*check\_2db\_double\_C3&&\*check\_2db\_double\_P>\*check\_2db\_double\_C4&&\*check\_2db\_double\_P>\*check\_2db\_double\_C5)

{

temp\_flhs=1;

db1\_temp=1;

\*win\_cardP=1;

}

if(db2\_temp==0)

{

if(\*check\_2db\_double\_P==\*check\_2db\_double\_C1||\*check\_2db\_double\_P==\*check\_2db\_double\_C2||\*check\_2db\_double\_P==\*check\_2db\_double\_C3||\*check\_2db\_double\_P==\*check\_2db\_double\_C4||\*check\_2db\_double\_P==\*check\_2db\_double\_C5)

{

if(\*check\_highP>=\*check\_highC1||\*check\_highP>=\*check\_highC2||\*check\_highP>=\*check\_highC3||\*check\_highP>=\*check\_highC4||\*check\_highP>=\*check\_highC5)

{

if(db2\_temp==0)

{

if(\*check\_highP>\*check\_highC1&&\*check\_highP>\*check\_highC2&&\*check\_highP>\*check\_highC3&&\*check\_highP>\*check\_highC4&&\*check\_highP>\*check\_highC5)

{

temp\_flhs=1;

\*win\_cardP=1;

}

if(db2\_temp==0)

{

if(\*check\_highP==\*check\_highC1&&\*check\_2db\_double\_P==\*check\_2db\_double\_C1&&\*check\_db\_double\_P==\*check\_db\_double\_C1)

{

checl\_db++;

\*win\_cardC1=1;

}

if(\*check\_highP==\*check\_highC2&&\*check\_2db\_double\_P==\*check\_2db\_double\_C2&&\*check\_db\_double\_P==\*check\_db\_double\_C1)

{

checl\_db++;

\*win\_cardC2=1;

}

if(\*check\_highP==\*check\_highC3&&\*check\_2db\_double\_P==\*check\_2db\_double\_C3&&\*check\_db\_double\_P==\*check\_db\_double\_C1)

{

checl\_db++;

\*win\_cardC3=1;

}

if(\*check\_highP==\*check\_highC4&&\*check\_2db\_double\_P==\*check\_2db\_double\_C4&&\*check\_db\_double\_P==\*check\_db\_double\_C1)

{

checl\_db++;

\*win\_cardC4=1;

}

if(\*check\_highP==\*check\_highC5&&\*check\_2db\_double\_P==\*check\_2db\_double\_C5&&\*check\_db\_double\_P==\*check\_db\_double\_C1)

{

checl\_db++;

\*win\_cardC5=1;

}

temp\_flhs=1;

\*win\_cardP=1;

}

}

}

}

}

}

}

}

if(\*check\_2db\_double\_P<\*check\_2db\_double\_C1||\*check\_2db\_double\_P<\*check\_2db\_double\_C2||\*check\_2db\_double\_P<\*check\_2db\_double\_C3||\*check\_2db\_double\_P<\*check\_2db\_double\_C4||\*check\_2db\_double\_P<\*check\_2db\_double\_C5)

{

temp\_flhs=1;

}

if(\*check\_db\_double\_P<\*check\_db\_double\_C1||\*check\_db\_double\_P<\*check\_db\_double\_C2||\*check\_db\_double\_P<\*check\_db\_double\_C3||\*check\_db\_double\_P<\*check\_db\_double\_C4||\*check\_db\_double\_P<\*check\_db\_double\_C5)

{

temp\_flhs=1;

}

}

if(\*check\_db\_double\_P<\*check\_db\_double\_C1||\*check\_db\_double\_P<\*check\_db\_double\_C2||\*check\_db\_double\_P<\*check\_db\_double\_C3||\*check\_db\_double\_P<\*check\_db\_double\_C4||\*check\_db\_double\_P<\*check\_db\_double\_C5)

{

temp\_flhs=1;

}

}

}

}

//-------------------HIGH-CARD---------------------

else

{

if(\*check\_db\_double\_P==0&&\*check\_db\_double\_C1==0&&\*check\_db\_double\_C2==0&&\*check\_db\_double\_C3==0&&\*check\_db\_double\_C4==0&&\*check\_db\_double\_C5==0&&\*check\_2db\_double\_P==0&&\*check\_2db\_double\_C1==0&&\*check\_2db\_double\_C2==0&&\*check\_2db\_double\_C3==0&&\*check\_2db\_double\_C4==0&&\*check\_2db\_double\_C5==0&&\*check\_tr\_triple\_P==0&&\*check\_tr\_triple\_C1==0&&\*check\_tr\_triple\_C2==0&&\*check\_tr\_triple\_C3==0&&\*check\_tr\_triple\_C4==0&&\*check\_tr\_triple\_C5==0&&\*check\_straighP==0&&\*check\_straighC1==0&&\*check\_straighC2==0&&\*check\_straighC3==0&&\*check\_straighC4==0&&\*check\_straighC5==0&&\*check\_flushP==0&&\*check\_flushC1==0&&\*check\_flushC2==0&&\*check\_flushC3==0&&\*check\_flushC4==0&&\*check\_flushC5==0&&\*check\_straight\_flushP==0&&\*check\_straight\_flushC1==0&&\*check\_straight\_flushC2==0&&\*check\_straight\_flushC3==0&&\*check\_straight\_flushC4==0&&\*check\_straight\_flushC5==0&&\*check\_flush\_royalP==0&&\*check\_flush\_royalC1==0&&\*check\_flush\_royalC2==0&&\*check\_flush\_royalC3==0&&\*check\_flush\_royalC4==0&&\*check\_flush\_royalC5==0)

{

if(\*check\_highP<\*check\_highC1||\*check\_highP<\*check\_highC2||\*check\_highP<\*check\_highC3||\*check\_highP<\*check\_highC4||\*check\_highP<\*check\_highC5)

{

temp\_flhs=1;

hg\_temp=1;

}

if(\*check\_highP>=\*check\_highC1&&\*check\_highP>=\*check\_highC2&&\*check\_highP>=\*check\_highC3&&\*check\_highP>=\*check\_highC4&&\*check\_highP>=\*check\_highC5)

{

if(hg\_temp==0)

{

if(\*check\_highP>\*check\_highC1&&\*check\_highP>\*check\_highC2&&\*check\_highP>\*check\_highC3&&\*check\_highP>\*check\_highC4&&\*check\_highP>\*check\_highC5)

{

temp\_flhs=1;

\*win\_cardP=1;

}

if(\*check\_highP==\*check\_highC1)

{

chek\_hg++;

\*win\_cardC1=1;

}

if(\*check\_highP==\*check\_highC2)

{

chek\_hg++;

\*win\_cardC2=1;

}

if(\*check\_highP==\*check\_highC3)

{

chek\_hg++;

\*win\_cardC3=1;

}

if(\*check\_highP==\*check\_highC4)

{

chek\_hg++;

\*win\_cardC4=1;

}

if(\*check\_highP==\*check\_highC5)

{

chek\_hg++;

\*win\_cardC5=1;

}

temp\_flhs=1;

\*win\_cardP=1;

}

}

}

}

}

}

}

}

}

}

}

//-------------------------------------------------

check\_rfl=1;

check\_fl=1;

check\_tr=1;

checl\_db=1;

chek\_hg=1;

check\_str=1;

temp\_flhs=0;

str\_tp=0;

db\_temp=0;

db1\_temp=0;

db2\_temp=0;

hg\_temp=0;

db\_max=0;

db2\_max=0;

\*check\_fh\_double\_P=0;

\*check\_fh\_triple\_P=0;

\*check\_tr\_triple\_P=0;

\*check\_db\_double\_P=0;

\*check\_2db\_double\_P=0;

\*check\_fh\_double\_C1=0;

\*check\_fh\_triple\_C1=0;

\*check\_tr\_triple\_C1=0;

\*check\_db\_double\_C1=0;

\*check\_2db\_double\_C1=0;

\*check\_fh\_double\_C2=0;

\*check\_fh\_triple\_C2=0;

\*check\_tr\_triple\_C2=0;

\*check\_db\_double\_C2=0;

\*check\_2db\_double\_C2=0;

\*check\_fh\_double\_C3=0;

\*check\_fh\_triple\_C3=0;

\*check\_tr\_triple\_C3=0;

\*check\_db\_double\_C3=0;

\*check\_2db\_double\_C3=0;

\*check\_fh\_double\_C4=0;

\*check\_fh\_triple\_C4=0;

\*check\_tr\_triple\_C4=0;

\*check\_db\_double\_C4=0;

\*check\_2db\_double\_C4=0;

\*check\_fh\_double\_C5=0;

\*check\_fh\_triple\_C5=0;

\*check\_tr\_triple\_C5=0;

\*check\_db\_double\_C5=0;

\*check\_2db\_double\_C5=0;

}

}

**CApp\_onEvent.cpp:**

#include"CApp.h"

void CApp::onEvent(SDL\_Event\* e)

{

CEvent::onEvent(e);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CApp::onExit()

{

running = false;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CApp::onLButtonDown(int xm, int ym)

{

if(temp\_card==0)

{

temp\_card++;

Mix\_PlayChannel(-1,snd\_place,0);

}

if(temp\_card==3||temp\_card==5||temp\_card==7)

{

Mix\_PlayChannel(-1,snd\_place,0);

int i1 = floor((xm - CARD\_LEFT) / 63);

int j1 = floor((ym - CARD\_TOP) / 31);

if(i1 < 0 || i1 > 2 || j1 < 0 || j1 > 1)

return;

if (i1 == 0)

{

fold\_card();

}

if (i1 == 1)

{

check\_card();

}

if (i1 == 2)

{

bet\_card();

}

}

if(temp\_card>10)

{

temp\_card=0;

Mix\_PlayChannel(-1,snd\_place,0);

check\_temp\_card\_straight=0;

check\_temp\_high\_card=0;

moneybox=0;

//--------------player--------------------

check\_flush\_royalP=0;

check\_straight\_flushP=0;

check\_kareP=0;

check\_tripleP=0;

check\_triple2P=0;

check\_doubleP=0;

check\_double2P=0;

check\_double3P=0;

check\_straighP=0;

check\_flushP=0;

check\_full\_houseP=0;

check\_highP=0;

straight\_tempP=10;

proverkaP=0;

check\_fh\_double\_P=0;

check\_fh\_triple\_P=0;

check\_tr\_triple\_P=0;

check\_db\_double\_P=0;

check\_2db\_double\_P=0;

win\_cardP=0;

//--------------player(END)--------------

//--------------computer-1---------------

check\_flush\_royalC1=0;

check\_straight\_flushC1=0;

check\_kareC1=0;

check\_tripleC1=0;

check\_triple2C1=0;

check\_doubleC1=0;

check\_double2C1=0;

check\_double3C1=0;

check\_straighC1=0;

check\_flushC1=0;

check\_full\_houseC1=0;

check\_highC1=0;

straight\_tempC1=10;

check\_fh\_double\_C1=0;

check\_fh\_triple\_C1=0;

check\_tr\_triple\_C1=0;

check\_db\_double\_C1=0;

check\_2db\_double\_C1=0;

win\_cardC1=0;

//--------------computer-1(END)----------

//--------------computer-2---------------

check\_flush\_royalC2=0;

check\_straight\_flushC2=0;

check\_kareC2=0;

check\_tripleC2=0;

check\_triple2C2=0;

check\_doubleC2=0;

check\_double2C2=0;

check\_double3C2=0;

check\_straighC2=0;

check\_flushC2=0;

check\_full\_houseC2=0;

check\_highC2=0;

straight\_tempC2=10;

check\_fh\_double\_C2=0;

check\_fh\_triple\_C2=0;

check\_tr\_triple\_C2=0;

check\_db\_double\_C2=0;

check\_2db\_double\_C2=0;

win\_cardC2=0;

//--------------computer-2(END)----------

//--------------computer-3---------------

check\_flush\_royalC3=0;

check\_straight\_flushC3=0;

check\_kareC3=0;

check\_tripleC3=0;

check\_triple2C3=0;

check\_doubleC3=0;

check\_double2C3=0;

check\_double3C3=0;

check\_straighC3=0;

check\_flushC3=0;

check\_full\_houseC3=0;

check\_highC3=0;

straight\_tempC3=10;

check\_fh\_double\_C3=0;

check\_fh\_triple\_C3=0;

check\_tr\_triple\_C3=0;

check\_db\_double\_C3=0;

check\_2db\_double\_C3=0;

win\_cardC3=0;

//--------------computer-3(END)----------

//--------------computer-4---------------

check\_flush\_royalC4=0;

check\_straight\_flushC4=0;

check\_kareC4=0;

check\_tripleC4=0;

check\_triple2C4=0;

check\_doubleC4=0;

check\_double2C4=0;

check\_double3C4=0;

check\_straighC4=0;

check\_flushC4=0;

check\_full\_houseC4=0;

check\_highC4=0;

straight\_tempC4=10;

check\_fh\_double\_C4=0;

check\_fh\_triple\_C4=0;

check\_tr\_triple\_C4=0;

check\_db\_double\_C4=0;

check\_2db\_double\_C4=0;

win\_cardC4=0;

//--------------computer-4(END)----------

//--------------computer-5---------------

check\_flush\_royalC5=0;

check\_straight\_flushC5=0;

check\_kareC5=0;

check\_tripleC5=0;

check\_triple2C5=0;

check\_doubleC5=0;

check\_double2C5=0;

check\_double3C5=0;

check\_straighC5=0;

check\_flushC5=0;

check\_full\_houseC5=0;

check\_highC5=0;

straight\_tempC5=10;

check\_fh\_double\_C5=0;

check\_fh\_triple\_C5=0;

check\_tr\_triple\_C5=0;

check\_db\_double\_C5=0;

check\_2db\_double\_C5=0;

win\_cardC5=0;

//--------------computer-5(END)----------

}

if(temp\_card<-15)

{

money=2000;

temp\_card=0;

temp\_lose=0;

temp\_win=0;

}

}

**Capp\_onInit.cpp:**

#include "CApp.h"

bool CApp::onInit()

{

if(SDL\_Init(SDL\_INIT\_EVERYTHING) < 0)

{

return false;

}

//----------------------------------

window = SDL\_CreateWindow(WINDOW\_CAPTION, 450, 200, WINDOW\_WIDTH, WINDOW\_HEIGHT, SDL\_WINDOW\_SHOWN);

if(window == NULL)

{

return false;

}

//----------------------------------

screen = SDL\_GetWindowSurface(window);

if(screen == NULL)

return false;

//----------------------------------

SDL\_SetSurfaceRLE(screen,1);

//----------------------------------

background = CSurface::loadSursface("Backgrounds/Background.bmp",screen);

//--------------CARDS--------------------

queue\_cards[0] = cart\_2\_spades = CSurface::loadSursface("Cards/2Spades.bmp",screen);

queue\_cards[1] = cart\_2\_clubs = CSurface::loadSursface("Cards/2Clubs.bmp",screen);

queue\_cards[2] = cart\_2\_hearts = CSurface::loadSursface("Cards/2Hearts.bmp",screen);

queue\_cards[3] = cart\_2\_diamonds = CSurface::loadSursface("Cards/2Diamonds.bmp",screen);

queue\_cards[4] = cart\_3\_spades = CSurface::loadSursface("Cards/3Spades.bmp",screen);

queue\_cards[5] = cart\_3\_clubs = CSurface::loadSursface("Cards/3Clubs.bmp",screen);

queue\_cards[6] = cart\_3\_hearts = CSurface::loadSursface("Cards/3Hearts.bmp",screen);

queue\_cards[7] = cart\_3\_diamonds = CSurface::loadSursface("Cards/3Diamonds.bmp",screen);

queue\_cards[8] = cart\_4\_spades = CSurface::loadSursface("Cards/4Spades.bmp",screen);

queue\_cards[9] = cart\_4\_clubs = CSurface::loadSursface("Cards/4Clubs.bmp",screen);

queue\_cards[10] = cart\_4\_hearts = CSurface::loadSursface("Cards/4Hearts.bmp",screen);

queue\_cards[11] = cart\_4\_diamonds = CSurface::loadSursface("Cards/4Diamonds.bmp",screen);

queue\_cards[12] = cart\_5\_spades = CSurface::loadSursface("Cards/5Spades.bmp",screen);

queue\_cards[13] = cart\_5\_clubs = CSurface::loadSursface("Cards/5Clubs.bmp",screen);

queue\_cards[14] = cart\_5\_hearts = CSurface::loadSursface("Cards/5Hearts.bmp",screen);

queue\_cards[15] = cart\_5\_diamonds = CSurface::loadSursface("Cards/5Diamonds.bmp",screen);

queue\_cards[16] = cart\_6\_spades = CSurface::loadSursface("Cards/6Spades.bmp",screen);

queue\_cards[17] = cart\_6\_clubs = CSurface::loadSursface("Cards/6Clubs.bmp",screen);

queue\_cards[18] = cart\_6\_hearts = CSurface::loadSursface("Cards/6Hearts.bmp",screen);

queue\_cards[19] = cart\_6\_diamonds = CSurface::loadSursface("Cards/6Diamonds.bmp",screen);

queue\_cards[20] = cart\_7\_spades = CSurface::loadSursface("Cards/7Spades.bmp",screen);

queue\_cards[21] = cart\_7\_clubs = CSurface::loadSursface("Cards/7Clubs.bmp",screen);

queue\_cards[22] = cart\_7\_hearts = CSurface::loadSursface("Cards/7Hearts.bmp",screen);

queue\_cards[23] = cart\_7\_diamonds = CSurface::loadSursface("Cards/7Diamonds.bmp",screen);

queue\_cards[24] = cart\_8\_spades = CSurface::loadSursface("Cards/8Spades.bmp",screen);

queue\_cards[25] = cart\_8\_clubs = CSurface::loadSursface("Cards/8Clubs.bmp",screen);

queue\_cards[26] = cart\_8\_hearts = CSurface::loadSursface("Cards/8Hearts.bmp",screen);

queue\_cards[27] = cart\_8\_diamonds = CSurface::loadSursface("Cards/8Diamonds.bmp",screen);

queue\_cards[28] = cart\_9\_spades = CSurface::loadSursface("Cards/9Spades.bmp",screen);

queue\_cards[29] = cart\_9\_clubs = CSurface::loadSursface("Cards/9Clubs.bmp",screen);

queue\_cards[30] = cart\_9\_hearts = CSurface::loadSursface("Cards/9Hearts.bmp",screen);

queue\_cards[31] = cart\_9\_diamonds = CSurface::loadSursface("Cards/9Diamonds.bmp",screen);

queue\_cards[32] = cart\_10\_spades = CSurface::loadSursface("Cards/10Spades.bmp",screen);

queue\_cards[33] = cart\_10\_clubs = CSurface::loadSursface("Cards/10Clubs.bmp",screen);

queue\_cards[34] = cart\_10\_hearts = CSurface::loadSursface("Cards/10Hearts.bmp",screen);

queue\_cards[35] = cart\_10\_diamonds = CSurface::loadSursface("Cards/10Diamonds.bmp",screen);

queue\_cards[36] = cart\_J\_spades = CSurface::loadSursface("Cards/JSpades.bmp",screen);

queue\_cards[37] = cart\_J\_clubs = CSurface::loadSursface("Cards/JClubs.bmp",screen);

queue\_cards[38] = cart\_J\_hearts = CSurface::loadSursface("Cards/JHearts.bmp",screen);

queue\_cards[39] = cart\_J\_diamonds = CSurface::loadSursface("Cards/JDiamonds.bmp",screen);

queue\_cards[40] = cart\_Q\_spades = CSurface::loadSursface("Cards/QSpades.bmp",screen);

queue\_cards[41] = cart\_Q\_clubs = CSurface::loadSursface("Cards/QClubs.bmp",screen);

queue\_cards[42] = cart\_Q\_hearts = CSurface::loadSursface("Cards/QHearts.bmp",screen);

queue\_cards[43] = cart\_Q\_diamonds = CSurface::loadSursface("Cards/QDiamonds.bmp",screen);

queue\_cards[44] = cart\_K\_spades = CSurface::loadSursface("Cards/KSpades.bmp",screen);

queue\_cards[45] = cart\_K\_clubs = CSurface::loadSursface("Cards/KClubs.bmp",screen);

queue\_cards[46] = cart\_K\_hearts = CSurface::loadSursface("Cards/KHearts.bmp",screen);

queue\_cards[47] = cart\_K\_diamonds = CSurface::loadSursface("Cards/KDiamonds.bmp",screen);

queue\_cards[48] = cart\_A\_spades = CSurface::loadSursface("Cards/ASpades.bmp",screen);

queue\_cards[49] = cart\_A\_clubs = CSurface::loadSursface("Cards/AClubs.bmp",screen);

queue\_cards[50] = cart\_A\_hearts = CSurface::loadSursface("Cards/AHearts.bmp",screen);

queue\_cards[51] = cart\_A\_diamonds = CSurface::loadSursface("Cards/ADiamonds.bmp",screen);

Back\_Card = CSurface::loadSursface("Cards/BackCard.bmp",screen);

//--------------CARDS--------------------

if(cart\_2\_spades == NULL || cart\_2\_clubs == NULL || cart\_2\_hearts == NULL || cart\_2\_diamonds == NULL)

{

return false;

}

if(cart\_3\_spades == NULL || cart\_3\_clubs == NULL || cart\_3\_hearts == NULL || cart\_3\_diamonds == NULL)

{

return false;

}

if(cart\_4\_spades == NULL || cart\_4\_clubs == NULL || cart\_4\_hearts == NULL || cart\_4\_diamonds == NULL)

{

return false;

}

if(cart\_5\_spades == NULL || cart\_5\_clubs == NULL || cart\_5\_hearts == NULL || cart\_5\_diamonds == NULL)

{

return false;

}

if(cart\_6\_spades == NULL || cart\_6\_clubs == NULL || cart\_6\_hearts == NULL || cart\_6\_diamonds == NULL)

{

return false;

}

if(cart\_7\_spades == NULL || cart\_7\_clubs == NULL || cart\_7\_hearts == NULL || cart\_7\_diamonds == NULL)

{

return false;

}

if(cart\_8\_spades == NULL || cart\_8\_clubs == NULL || cart\_8\_hearts == NULL || cart\_8\_diamonds == NULL)

{

return false;

}

if(cart\_9\_spades == NULL || cart\_9\_clubs == NULL || cart\_9\_hearts == NULL || cart\_9\_diamonds == NULL)

{

return false;

}

if(cart\_10\_spades == NULL || cart\_10\_clubs == NULL || cart\_10\_hearts == NULL || cart\_10\_diamonds == NULL)

{

return false;

}

if(cart\_J\_spades == NULL || cart\_J\_clubs == NULL || cart\_J\_hearts == NULL || cart\_J\_diamonds == NULL)

{

return false;

}

if(cart\_Q\_spades == NULL || cart\_Q\_clubs == NULL || cart\_Q\_hearts == NULL || cart\_Q\_diamonds == NULL)

{

return false;

}

if(cart\_K\_spades == NULL || cart\_K\_clubs == NULL || cart\_K\_hearts == NULL || cart\_K\_diamonds == NULL)

{

return false;

}

if(cart\_A\_spades == NULL || cart\_A\_clubs == NULL || cart\_A\_hearts == NULL || cart\_A\_diamonds == NULL)

{

return false;

}

if(Back\_Card == NULL)

{

return false;

}

//--------------CARDS--------------------

if(background == NULL)

{

return false;

}

//-----------------------------

if(Mix\_OpenAudio(44100, MIX\_DEFAULT\_FORMAT, 2, 2048)==-1)

{

return false;

}

//-----------------------------

if(!loadMedia())

return false;

return true;

}

bool CApp::loadMedia()

{

snd\_draw = Mix\_LoadWAV("Music/Draw.wav");

snd\_lose = Mix\_LoadWAV("Music/Lose.wav");

snd\_place = Mix\_LoadWAV("Music/Place.wav");

snd\_win = Mix\_LoadWAV("Music/Win.wav");

if(snd\_draw == NULL || snd\_lose == NULL || snd\_place == NULL || snd\_win == NULL)

return false;

//--------------------------------

if(TTF\_Init()== -1)

return false;

font = TTF\_OpenFont("Fonts/RobotoCondensed-Italic.ttf",24);

if(!font)

return false;

return true;

}

**CApp\_onLoop.cpp:**

#include "CApp.h"

void CApp::onLoop()

{

if(temp\_card==1)

{

queue\_cards[card\_1\_player=(rand() % 52)];

temp\_card++;

}

if(temp\_card==2)

{

random\_card\_2\_player();

queue\_cards[card\_2\_player];

random\_card\_1\_computer1();

queue\_cards[card\_1\_computer1];

random\_card\_2\_computer1();

queue\_cards[card\_2\_computer1];

random\_card\_1\_computer2();

queue\_cards[card\_1\_computer2];

random\_card\_2\_computer2();

queue\_cards[card\_2\_computer2];

random\_card\_1\_computer3();

queue\_cards[card\_1\_computer3];

random\_card\_2\_computer3();

queue\_cards[card\_2\_computer3];

random\_card\_1\_computer4();

queue\_cards[card\_1\_computer4];

random\_card\_2\_computer4();

queue\_cards[card\_2\_computer4];

random\_card\_1\_computer5();

queue\_cards[card\_1\_computer5];

random\_card\_2\_computer5();

queue\_cards[card\_2\_computer5];

temp\_card++;

}

if(temp\_card==4)

{

random\_card\_1\_all();

queue\_cards[card\_1\_all];

random\_card\_2\_all();

queue\_cards[card\_2\_all];

random\_card\_3\_all();

queue\_cards[card\_3\_all];

temp\_card++;

}

if(temp\_card==6)

{

random\_card\_4\_all();

queue\_cards[card\_4\_all];

temp\_card++;

}

if(temp\_card==8)

{

random\_card\_5\_all();

queue\_cards[card\_5\_all];

temp\_card++;

}

if(temp\_card==9)

{

combinations\_check(card\_1\_player, card\_2\_player, &check\_flush\_royalP, &check\_straight\_flushP, &check\_kareP, &check\_tripleP, &check\_triple2P, &check\_doubleP, &check\_double2P, &check\_double3P, &check\_straighP, &check\_flushP, &check\_full\_houseP, &check\_highP, &straight\_tempP);

check\_temp\_card\_straight=0;

check\_temp\_high\_card=0;

combinations\_check(card\_1\_computer1, card\_2\_computer1, &check\_flush\_royalC1, &check\_straight\_flushC1, &check\_kareC1, &check\_tripleC1, &check\_triple2C1, &check\_doubleC1, &check\_double2C1, &check\_double3C1, &check\_straighC1, &check\_flushC1, &check\_full\_houseC1, &check\_highC1, &straight\_tempC1);

check\_temp\_card\_straight=0;

check\_temp\_high\_card=0;

combinations\_check(card\_1\_computer2, card\_2\_computer2, &check\_flush\_royalC2, &check\_straight\_flushC2, &check\_kareC2, &check\_tripleC2, &check\_triple2C2, &check\_doubleC2, &check\_double2C2, &check\_double3C2, &check\_straighC2, &check\_flushC2, &check\_full\_houseC2, &check\_highC2, &straight\_tempC2);

check\_temp\_card\_straight=0;

check\_temp\_high\_card=0;

combinations\_check(card\_1\_computer3, card\_2\_computer3, &check\_flush\_royalC3, &check\_straight\_flushC3, &check\_kareC3, &check\_tripleC3, &check\_triple2C3, &check\_doubleC3, &check\_double2C3, &check\_double3C3, &check\_straighC3, &check\_flushC3, &check\_full\_houseC3, &check\_highC3, &straight\_tempC3);

check\_temp\_card\_straight=0;

check\_temp\_high\_card=0;

combinations\_check(card\_1\_computer4, card\_2\_computer4, &check\_flush\_royalC4, &check\_straight\_flushC4, &check\_kareC4, &check\_tripleC4, &check\_triple2C4, &check\_doubleC4, &check\_double2C4, &check\_double3C4, &check\_straighC4, &check\_flushC4, &check\_full\_houseC4, &check\_highC4, &straight\_tempC4);

check\_temp\_card\_straight=0;

check\_temp\_high\_card=0;

combinations\_check(card\_1\_computer5, card\_2\_computer5, &check\_flush\_royalC5, &check\_straight\_flushC5, &check\_kareC5, &check\_tripleC5, &check\_triple2C5, &check\_doubleC5, &check\_double2C5, &check\_double3C5, &check\_straighC5, &check\_flushC5, &check\_full\_houseC5, &check\_highC5, &straight\_tempC5);

}

}

**CApp\_onQuit.cpp:**

#include "CApp.h"

void CApp::onQuit()

{

TTF\_CloseFont(font);

font = NULL;

Mix\_FreeChunk(snd\_draw);

snd\_draw = NULL;

Mix\_FreeChunk(snd\_lose);

snd\_lose = NULL;

Mix\_FreeChunk(snd\_place);

snd\_place = NULL;

Mix\_FreeChunk(snd\_win);

snd\_win = NULL;

SDL\_FreeSurface(screen);

screen = NULL;

SDL\_FreeSurface(background);

background = NULL;

SDL\_DestroyWindow(window);

window = NULL;

//--------------CARDS--------------------

SDL\_FreeSurface(cart\_2\_spades);

cart\_2\_spades = NULL;

SDL\_FreeSurface(cart\_2\_clubs);

cart\_2\_clubs = NULL;

SDL\_FreeSurface(cart\_2\_hearts);

cart\_2\_hearts = NULL;

SDL\_FreeSurface(cart\_2\_diamonds);

cart\_2\_diamonds = NULL;

SDL\_FreeSurface(cart\_3\_spades);

cart\_3\_spades = NULL;

SDL\_FreeSurface(cart\_3\_clubs);

cart\_3\_clubs = NULL;

SDL\_FreeSurface(cart\_3\_hearts);

cart\_3\_hearts = NULL;

SDL\_FreeSurface(cart\_3\_diamonds);

cart\_3\_diamonds = NULL;

SDL\_FreeSurface(cart\_4\_spades);

cart\_4\_spades = NULL;

SDL\_FreeSurface(cart\_4\_clubs);

cart\_4\_clubs = NULL;

SDL\_FreeSurface(cart\_4\_hearts);

cart\_4\_hearts = NULL;

SDL\_FreeSurface(cart\_4\_diamonds);

cart\_4\_diamonds = NULL;

SDL\_FreeSurface(cart\_5\_spades);

cart\_5\_spades = NULL;

SDL\_FreeSurface(cart\_5\_clubs);

cart\_5\_clubs = NULL;

SDL\_FreeSurface(cart\_5\_hearts);

cart\_5\_hearts = NULL;

SDL\_FreeSurface(cart\_5\_diamonds);

cart\_5\_diamonds = NULL;

SDL\_FreeSurface(cart\_6\_spades);

cart\_6\_spades = NULL;

SDL\_FreeSurface(cart\_6\_clubs);

cart\_6\_clubs = NULL;

SDL\_FreeSurface(cart\_6\_hearts);

cart\_6\_hearts = NULL;

SDL\_FreeSurface(cart\_6\_diamonds);

cart\_6\_diamonds = NULL;

SDL\_FreeSurface(cart\_7\_spades);

cart\_7\_spades = NULL;

SDL\_FreeSurface(cart\_7\_clubs);

cart\_7\_clubs = NULL;

SDL\_FreeSurface(cart\_7\_hearts);

cart\_7\_hearts = NULL;

SDL\_FreeSurface(cart\_7\_diamonds);

cart\_7\_diamonds = NULL;

SDL\_FreeSurface(cart\_8\_spades);

cart\_8\_spades = NULL;

SDL\_FreeSurface(cart\_8\_clubs);

cart\_8\_clubs = NULL;

SDL\_FreeSurface(cart\_8\_hearts);

cart\_8\_hearts = NULL;

SDL\_FreeSurface(cart\_8\_diamonds);

cart\_8\_diamonds = NULL;

SDL\_FreeSurface(cart\_9\_spades);

cart\_9\_spades = NULL;

SDL\_FreeSurface(cart\_9\_clubs);

cart\_9\_clubs = NULL;

SDL\_FreeSurface(cart\_9\_hearts);

cart\_9\_hearts = NULL;

SDL\_FreeSurface(cart\_9\_diamonds);

cart\_9\_diamonds = NULL;

SDL\_FreeSurface(cart\_10\_spades);

cart\_10\_spades = NULL;

SDL\_FreeSurface(cart\_10\_clubs);

cart\_10\_clubs = NULL;

SDL\_FreeSurface(cart\_10\_hearts);

cart\_10\_hearts = NULL;

SDL\_FreeSurface(cart\_10\_diamonds);

cart\_10\_diamonds = NULL;

SDL\_FreeSurface(cart\_J\_spades);

cart\_J\_spades = NULL;

SDL\_FreeSurface(cart\_J\_clubs);

cart\_J\_clubs = NULL;

SDL\_FreeSurface(cart\_J\_hearts);

cart\_J\_hearts = NULL;

SDL\_FreeSurface(cart\_J\_diamonds);

cart\_J\_diamonds = NULL;

SDL\_FreeSurface(cart\_Q\_spades);

cart\_Q\_spades = NULL;

SDL\_FreeSurface(cart\_Q\_clubs);

cart\_Q\_clubs = NULL;

SDL\_FreeSurface(cart\_Q\_hearts);

cart\_Q\_hearts = NULL;

SDL\_FreeSurface(cart\_Q\_diamonds);

cart\_Q\_diamonds = NULL;

SDL\_FreeSurface(cart\_K\_spades);

cart\_K\_spades = NULL;

SDL\_FreeSurface(cart\_K\_clubs);

cart\_K\_clubs = NULL;

SDL\_FreeSurface(cart\_K\_hearts);

cart\_K\_hearts = NULL;

SDL\_FreeSurface(cart\_K\_diamonds);

cart\_K\_diamonds = NULL;

SDL\_FreeSurface(cart\_A\_spades);

cart\_A\_spades = NULL;

SDL\_FreeSurface(cart\_A\_clubs);

cart\_A\_clubs = NULL;

SDL\_FreeSurface(cart\_A\_hearts);

cart\_A\_hearts = NULL;

SDL\_FreeSurface(cart\_A\_diamonds);

cart\_A\_diamonds = NULL;

//--------------CARDS--------------------

Mix\_Quit();

TTF\_Quit();

SDL\_Quit();

}

**CApp\_onRender.cpp:**

#include "CApp.h"

void CApp::onRender()

{

CSurface::drawSurface(background,screen, 0,0);

field\_draw();

SDL\_UpdateWindowSurface(window);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CApp::field\_draw()

{

if(temp\_card>10)

{if(money<0)temp\_card=-10;}

std::string szPlayer = "Player";

std::string szComputer1 = "Computer 1";

std::string szComputer2 = "Computer 2";

std::string szComputer3 = "Computer 3";

std::string szComputer4 = "Computer 4";

std::string szComputer5 = "Computer 5";

std::string szWin1 = "WINNER";

std::string szLose = "GAME OVER";

std::string szWin = "CONGRATULATIONS";

std::string szComp = "Press LBM to start the game...";

std::string szDraw = "You have: " + std::to\_string(money);

//-----------------------------

if(temp\_card>0&&temp\_card<9)

{

CSurface::drawSurface(Back\_Card, screen, 0, 275);

CSurface::drawSurface(Back\_Card, screen, 55, 275);

CSurface::drawSurface(Back\_Card, screen, 20, 5);

CSurface::drawSurface(Back\_Card, screen, 75, 5);

CSurface::drawSurface(Back\_Card, screen, 260, 0);

CSurface::drawSurface(Back\_Card, screen, 315, 0);

CSurface::drawSurface(Back\_Card, screen, 510, 5);

CSurface::drawSurface(Back\_Card, screen, 565, 5);

CSurface::drawSurface(Back\_Card, screen, 535, 275);

CSurface::drawSurface(Back\_Card, screen, 590, 275);

}

if(win\_cardP==1)

{

CSurface::drawTextSurface(font,screen,szWin1.c\_str(),280,245,255,0,0);

}

if(win\_cardC1==1)

{

CSurface::drawTextSurface(font,screen,szWin1.c\_str(),0,215,255,0,0);

}

if(win\_cardC2==1)

{

CSurface::drawTextSurface(font,screen,szWin1.c\_str(),20,105,255,0,0);

}

if(win\_cardC3==1)

{

CSurface::drawTextSurface(font,screen,szWin1.c\_str(),260,95,255,0,0);

}

if(win\_cardC4==1)

{

CSurface::drawTextSurface(font,screen,szWin1.c\_str(),505,105,255,0,0);

}

if(win\_cardC5==1)

{

CSurface::drawTextSurface(font,screen,szWin1.c\_str(),535,215,255,0,0);

}

if(temp\_card>0)

{

CSurface::drawTextSurface(font,screen,szPlayer.c\_str(),280,265,255,155,0);

CSurface::drawTextSurface(font,screen,szComputer1.c\_str(),0,235,0,255,255);

CSurface::drawTextSurface(font,screen,szComputer2.c\_str(),20,80,0,255,255);

CSurface::drawTextSurface(font,screen,szComputer3.c\_str(),260,70,0,255,255);

CSurface::drawTextSurface(font,screen,szComputer4.c\_str(),505,80,0,255,255);

CSurface::drawTextSurface(font,screen,szComputer5.c\_str(),535,235,0,255,255);

}

if(temp\_card==0)

{

CSurface::drawTextSurface(font,screen,szComp.c\_str(),200,175,255,0,0);

}

CSurface::drawTextSurface(font,screen,szDraw.c\_str(),8,410,0,0,0);

//-----------------------------

if(temp\_card>0)

{

CSurface::drawSurface(queue\_cards[card\_1\_player], screen, 260, 295);

}

if(temp\_card>1)

{

CSurface::drawSurface(queue\_cards[card\_2\_player], screen, 315, 295);

}

if(temp\_card>4)

{

CSurface::drawSurface(queue\_cards[card\_1\_all], screen, 120, 160);

CSurface::drawSurface(queue\_cards[card\_2\_all], screen, 210, 160);

CSurface::drawSurface(queue\_cards[card\_3\_all], screen, 300, 160);

}

if(temp\_card>6)

{

CSurface::drawSurface(queue\_cards[card\_4\_all], screen, 390, 160);

}

if(temp\_card>8)

{

CSurface::drawSurface(queue\_cards[card\_5\_all], screen, 480, 160);

}

if(temp\_card>8)

{

CSurface::drawSurface(queue\_cards[card\_1\_computer1], screen, 0, 275);

CSurface::drawSurface(queue\_cards[card\_2\_computer1], screen, 55, 275);

CSurface::drawSurface(queue\_cards[card\_1\_computer2], screen, 20, 5);

CSurface::drawSurface(queue\_cards[card\_2\_computer2], screen, 75, 5);

CSurface::drawSurface(queue\_cards[card\_1\_computer3], screen, 260, 0);

CSurface::drawSurface(queue\_cards[card\_2\_computer3], screen, 315, 0);

CSurface::drawSurface(queue\_cards[card\_1\_computer4], screen, 510, 5);

CSurface::drawSurface(queue\_cards[card\_2\_computer4], screen, 565, 5);

CSurface::drawSurface(queue\_cards[card\_1\_computer5], screen, 535, 275);

CSurface::drawSurface(queue\_cards[card\_2\_computer5], screen, 590, 275);

}

if(temp\_card==9)

{

round\_card++;

check\_card\_win();

//====================================================================================================================================================================

check\_card\_win( &check\_flush\_royalC1, &check\_straight\_flushC1, &check\_kareC1, &check\_tripleC1, &check\_triple2C1, &check\_doubleC1,

&check\_double2C1, &check\_double3C1, &check\_straighC1, &check\_flushC1, &check\_full\_houseC1, &check\_highC1, &straight\_tempC1, &check\_fh\_double\_C1,

&check\_fh\_triple\_C1, &check\_tr\_triple\_C1, &check\_db\_double\_C1, &check\_2db\_double\_C1, &win\_cardC1,

&check\_flush\_royalP, &check\_straight\_flushP, &check\_kareP, &check\_tripleP, &check\_triple2P, &check\_doubleP, &check\_double2P,

&check\_double3P, &check\_straighP, &check\_flushP, &check\_full\_houseP, &check\_highP, &straight\_tempP, &check\_fh\_double\_P, &check\_fh\_triple\_P,

&check\_tr\_triple\_P, &check\_db\_double\_P, &check\_2db\_double\_P, &win\_cardP,

&check\_flush\_royalC2, &check\_straight\_flushC2, &check\_kareC2, &check\_tripleC2, &check\_triple2C2, &check\_doubleC2, &check\_double2C2, &check\_double3C2,

&check\_straighC2, &check\_flushC2, &check\_full\_houseC2, &check\_highC2, &straight\_tempC2, &check\_fh\_double\_C2, &check\_fh\_triple\_C2, &check\_tr\_triple\_C2,

&check\_db\_double\_C2, &check\_2db\_double\_C2, &win\_cardC2,

&check\_flush\_royalC3, &check\_straight\_flushC3, &check\_kareC3, &check\_tripleC3, &check\_triple2C3, &check\_doubleC3, &check\_double2C3,

&check\_double3C3, &check\_straighC3, &check\_flushC3, &check\_full\_houseC3, &check\_highC3, &straight\_tempC3, &check\_fh\_double\_C3, &check\_fh\_triple\_C3, &check\_tr\_triple\_C3,

&check\_db\_double\_C3, &check\_2db\_double\_C3, &win\_cardC3,

&check\_flush\_royalC4, &check\_straight\_flushC4, &check\_kareC4, &check\_tripleC4, &check\_triple2C4, &check\_doubleC4, &check\_double2C4, &check\_double3C4,

&check\_straighC4, &check\_flushC4, &check\_full\_houseC4, &check\_highC4, &straight\_tempC4, &check\_fh\_double\_C4, &check\_fh\_triple\_C4, &check\_tr\_triple\_C4,

&check\_db\_double\_C4, &check\_2db\_double\_C4, &win\_cardC4,

&check\_flush\_royalC5, &check\_straight\_flushC5, &check\_kareC5, &check\_tripleC5, &check\_triple2C5, &check\_doubleC5, &check\_double2C5, &check\_double3C5, &check\_straighC5,

&check\_flushC5, &check\_full\_houseC5, &check\_highC5, &straight\_tempC5, &check\_fh\_double\_C5, &check\_fh\_triple\_C5, &check\_tr\_triple\_C5,

&check\_db\_double\_C5, &check\_2db\_double\_C5, &win\_cardC5);

//====================================================================================================================================================================

check\_card\_win( &check\_flush\_royalC2, &check\_straight\_flushC2, &check\_kareC2, &check\_tripleC2, &check\_triple2C2, &check\_doubleC2, &check\_double2C2, &check\_double3C2,

&check\_straighC2, &check\_flushC2, &check\_full\_houseC2, &check\_highC2, &straight\_tempC2, &check\_fh\_double\_C2, &check\_fh\_triple\_C2, &check\_tr\_triple\_C2,

&check\_db\_double\_C2, &check\_2db\_double\_C2, &win\_cardC2,

&check\_flush\_royalC1, &check\_straight\_flushC1, &check\_kareC1, &check\_tripleC1, &check\_triple2C1, &check\_doubleC1,

&check\_double2C1, &check\_double3C1, &check\_straighC1, &check\_flushC1, &check\_full\_houseC1, &check\_highC1, &straight\_tempC1, &check\_fh\_double\_C1,

&check\_fh\_triple\_C1, &check\_tr\_triple\_C1, &check\_db\_double\_C1, &check\_2db\_double\_C1, &win\_cardC1,

&check\_flush\_royalP, &check\_straight\_flushP, &check\_kareP, &check\_tripleP, &check\_triple2P, &check\_doubleP, &check\_double2P,

&check\_double3P, &check\_straighP, &check\_flushP, &check\_full\_houseP, &check\_highP, &straight\_tempP, &check\_fh\_double\_P, &check\_fh\_triple\_P,

&check\_tr\_triple\_P, &check\_db\_double\_P, &check\_2db\_double\_P, &win\_cardP,

&check\_flush\_royalC3, &check\_straight\_flushC3, &check\_kareC3, &check\_tripleC3, &check\_triple2C3, &check\_doubleC3, &check\_double2C3,

&check\_double3C3, &check\_straighC3, &check\_flushC3, &check\_full\_houseC3, &check\_highC3, &straight\_tempC3, &check\_fh\_double\_C3, &check\_fh\_triple\_C3, &check\_tr\_triple\_C3,

&check\_db\_double\_C3, &check\_2db\_double\_C3, &win\_cardC3,

&check\_flush\_royalC4, &check\_straight\_flushC4, &check\_kareC4, &check\_tripleC4, &check\_triple2C4, &check\_doubleC4, &check\_double2C4, &check\_double3C4,

&check\_straighC4, &check\_flushC4, &check\_full\_houseC4, &check\_highC4, &straight\_tempC4, &check\_fh\_double\_C4, &check\_fh\_triple\_C4, &check\_tr\_triple\_C4,

&check\_db\_double\_C4, &check\_2db\_double\_C4, &win\_cardC4,

&check\_flush\_royalC5, &check\_straight\_flushC5, &check\_kareC5, &check\_tripleC5, &check\_triple2C5, &check\_doubleC5, &check\_double2C5, &check\_double3C5, &check\_straighC5,

&check\_flushC5, &check\_full\_houseC5, &check\_highC5, &straight\_tempC5, &check\_fh\_double\_C5, &check\_fh\_triple\_C5, &check\_tr\_triple\_C5,

&check\_db\_double\_C5, &check\_2db\_double\_C5, &win\_cardC5);

//====================================================================================================================================================================

check\_card\_win(&check\_flush\_royalC3, &check\_straight\_flushC3, &check\_kareC3, &check\_tripleC3, &check\_triple2C3, &check\_doubleC3, &check\_double2C3,

&check\_double3C3, &check\_straighC3, &check\_flushC3, &check\_full\_houseC3, &check\_highC3, &straight\_tempC3, &check\_fh\_double\_C3, &check\_fh\_triple\_C3, &check\_tr\_triple\_C3,

&check\_db\_double\_C3, &check\_2db\_double\_C3, &win\_cardC3,

&check\_flush\_royalC1, &check\_straight\_flushC1, &check\_kareC1, &check\_tripleC1, &check\_triple2C1, &check\_doubleC1,

&check\_double2C1, &check\_double3C1, &check\_straighC1, &check\_flushC1, &check\_full\_houseC1, &check\_highC1, &straight\_tempC1, &check\_fh\_double\_C1,

&check\_fh\_triple\_C1, &check\_tr\_triple\_C1, &check\_db\_double\_C1, &check\_2db\_double\_C1, &win\_cardC1,

&check\_flush\_royalC2, &check\_straight\_flushC2, &check\_kareC2, &check\_tripleC2, &check\_triple2C2, &check\_doubleC2, &check\_double2C2, &check\_double3C2,

&check\_straighC2, &check\_flushC2, &check\_full\_houseC2, &check\_highC2, &straight\_tempC2, &check\_fh\_double\_C2, &check\_fh\_triple\_C2, &check\_tr\_triple\_C2,

&check\_db\_double\_C2, &check\_2db\_double\_C2, &win\_cardC2,

&check\_flush\_royalP, &check\_straight\_flushP, &check\_kareP, &check\_tripleP, &check\_triple2P, &check\_doubleP, &check\_double2P,

&check\_double3P, &check\_straighP, &check\_flushP, &check\_full\_houseP, &check\_highP, &straight\_tempP, &check\_fh\_double\_P, &check\_fh\_triple\_P,

&check\_tr\_triple\_P, &check\_db\_double\_P, &check\_2db\_double\_P, &win\_cardP,

&check\_flush\_royalC4, &check\_straight\_flushC4, &check\_kareC4, &check\_tripleC4, &check\_triple2C4, &check\_doubleC4, &check\_double2C4, &check\_double3C4,

&check\_straighC4, &check\_flushC4, &check\_full\_houseC4, &check\_highC4, &straight\_tempC4, &check\_fh\_double\_C4, &check\_fh\_triple\_C4, &check\_tr\_triple\_C4,

&check\_db\_double\_C4, &check\_2db\_double\_C4, &win\_cardC4,

&check\_flush\_royalC5, &check\_straight\_flushC5, &check\_kareC5, &check\_tripleC5, &check\_triple2C5, &check\_doubleC5, &check\_double2C5, &check\_double3C5, &check\_straighC5,

&check\_flushC5, &check\_full\_houseC5, &check\_highC5, &straight\_tempC5, &check\_fh\_double\_C5, &check\_fh\_triple\_C5, &check\_tr\_triple\_C5,

&check\_db\_double\_C5, &check\_2db\_double\_C5, &win\_cardC5);

//====================================================================================================================================================================

check\_card\_win(&check\_flush\_royalC4, &check\_straight\_flushC4, &check\_kareC4, &check\_tripleC4, &check\_triple2C4, &check\_doubleC4, &check\_double2C4, &check\_double3C4,

&check\_straighC4, &check\_flushC4, &check\_full\_houseC4, &check\_highC4, &straight\_tempC4, &check\_fh\_double\_C4, &check\_fh\_triple\_C4, &check\_tr\_triple\_C4,

&check\_db\_double\_C4, &check\_2db\_double\_C4, &win\_cardC4,

&check\_flush\_royalC1, &check\_straight\_flushC1, &check\_kareC1, &check\_tripleC1, &check\_triple2C1, &check\_doubleC1,

&check\_double2C1, &check\_double3C1, &check\_straighC1, &check\_flushC1, &check\_full\_houseC1, &check\_highC1, &straight\_tempC1, &check\_fh\_double\_C1,

&check\_fh\_triple\_C1, &check\_tr\_triple\_C1, &check\_db\_double\_C1, &check\_2db\_double\_C1, &win\_cardC1,

&check\_flush\_royalC2, &check\_straight\_flushC2, &check\_kareC2, &check\_tripleC2, &check\_triple2C2, &check\_doubleC2, &check\_double2C2, &check\_double3C2,

&check\_straighC2, &check\_flushC2, &check\_full\_houseC2, &check\_highC2, &straight\_tempC2, &check\_fh\_double\_C2, &check\_fh\_triple\_C2, &check\_tr\_triple\_C2,

&check\_db\_double\_C2, &check\_2db\_double\_C2, &win\_cardC2,

&check\_flush\_royalC3, &check\_straight\_flushC3, &check\_kareC3, &check\_tripleC3, &check\_triple2C3, &check\_doubleC3, &check\_double2C3,

&check\_double3C3, &check\_straighC3, &check\_flushC3, &check\_full\_houseC3, &check\_highC3, &straight\_tempC3, &check\_fh\_double\_C3, &check\_fh\_triple\_C3, &check\_tr\_triple\_C3,

&check\_db\_double\_C3, &check\_2db\_double\_C3, &win\_cardC3,

&check\_flush\_royalP, &check\_straight\_flushP, &check\_kareP, &check\_tripleP, &check\_triple2P, &check\_doubleP, &check\_double2P,

&check\_double3P, &check\_straighP, &check\_flushP, &check\_full\_houseP, &check\_highP, &straight\_tempP, &check\_fh\_double\_P, &check\_fh\_triple\_P,

&check\_tr\_triple\_P, &check\_db\_double\_P, &check\_2db\_double\_P, &win\_cardP,

&check\_flush\_royalC5, &check\_straight\_flushC5, &check\_kareC5, &check\_tripleC5, &check\_triple2C5, &check\_doubleC5, &check\_double2C5, &check\_double3C5, &check\_straighC5,

&check\_flushC5, &check\_full\_houseC5, &check\_highC5, &straight\_tempC5, &check\_fh\_double\_C5, &check\_fh\_triple\_C5, &check\_tr\_triple\_C5,

&check\_db\_double\_C5, &check\_2db\_double\_C5, &win\_cardC5);

//====================================================================================================================================================================

check\_card\_win(&check\_flush\_royalC5, &check\_straight\_flushC5, &check\_kareC5, &check\_tripleC5, &check\_triple2C5, &check\_doubleC5, &check\_double2C5, &check\_double3C5, &check\_straighC5,

&check\_flushC5, &check\_full\_houseC5, &check\_highC5, &straight\_tempC5, &check\_fh\_double\_C5, &check\_fh\_triple\_C5, &check\_tr\_triple\_C5,

&check\_db\_double\_C5, &check\_2db\_double\_C5, &win\_cardC5,

&check\_flush\_royalC1, &check\_straight\_flushC1, &check\_kareC1, &check\_tripleC1, &check\_triple2C1, &check\_doubleC1,

&check\_double2C1, &check\_double3C1, &check\_straighC1, &check\_flushC1, &check\_full\_houseC1, &check\_highC1, &straight\_tempC1, &check\_fh\_double\_C1,

&check\_fh\_triple\_C1, &check\_tr\_triple\_C1, &check\_db\_double\_C1, &check\_2db\_double\_C1, &win\_cardC1,

&check\_flush\_royalC2, &check\_straight\_flushC2, &check\_kareC2, &check\_tripleC2, &check\_triple2C2, &check\_doubleC2, &check\_double2C2, &check\_double3C2,

&check\_straighC2, &check\_flushC2, &check\_full\_houseC2, &check\_highC2, &straight\_tempC2, &check\_fh\_double\_C2, &check\_fh\_triple\_C2, &check\_tr\_triple\_C2,

&check\_db\_double\_C2, &check\_2db\_double\_C2, &win\_cardC2,

&check\_flush\_royalC3, &check\_straight\_flushC3, &check\_kareC3, &check\_tripleC3, &check\_triple2C3, &check\_doubleC3, &check\_double2C3,

&check\_double3C3, &check\_straighC3, &check\_flushC3, &check\_full\_houseC3, &check\_highC3, &straight\_tempC3, &check\_fh\_double\_C3, &check\_fh\_triple\_C3, &check\_tr\_triple\_C3,

&check\_db\_double\_C3, &check\_2db\_double\_C3, &win\_cardC3,

&check\_flush\_royalC4, &check\_straight\_flushC4, &check\_kareC4, &check\_tripleC4, &check\_triple2C4, &check\_doubleC4, &check\_double2C4, &check\_double3C4,

&check\_straighC4, &check\_flushC4, &check\_full\_houseC4, &check\_highC4, &straight\_tempC4, &check\_fh\_double\_C4, &check\_fh\_triple\_C4, &check\_tr\_triple\_C4,

&check\_db\_double\_C4, &check\_2db\_double\_C4, &win\_cardC4,

&check\_flush\_royalP, &check\_straight\_flushP, &check\_kareP, &check\_tripleP, &check\_triple2P, &check\_doubleP, &check\_double2P,

&check\_double3P, &check\_straighP, &check\_flushP, &check\_full\_houseP, &check\_highP, &straight\_tempP, &check\_fh\_double\_P, &check\_fh\_triple\_P,

&check\_tr\_triple\_P, &check\_db\_double\_P, &check\_2db\_double\_P, &win\_cardP);

//====================================================================================================================================================================

temp\_card++;

}

if(temp\_card<-3&&temp\_card>-100)

{

if(money<0)

{

CSurface::drawTextSurface(font,screen,szLose.c\_str(),265,175,255,0,0);

if(temp\_lose==0)

{

temp\_card=-25;

Mix\_PlayChannel(-1,snd\_draw,0);

SDL\_Delay(4000);

Mix\_PlayChannel(-1,snd\_lose,0);

SDL\_Delay(1000);

temp\_lose++;

}

//--------------player--------------------

check\_flush\_royalP=0;

check\_straight\_flushP=0;

check\_kareP=0;

check\_tripleP=0;

check\_triple2P=0;

check\_doubleP=0;

check\_double2P=0;

check\_double3P=0;

check\_straighP=0;

check\_flushP=0;

check\_full\_houseP=0;

check\_highP=0;

straight\_tempP=10;

proverkaP=0;

check\_fh\_double\_P=0;

check\_fh\_triple\_P=0;

check\_tr\_triple\_P=0;

check\_db\_double\_P=0;

check\_2db\_double\_P=0;

win\_cardP=0;

//--------------player(END)--------------

//--------------computer-1---------------

check\_flush\_royalC1=0;

check\_straight\_flushC1=0;

check\_kareC1=0;

check\_tripleC1=0;

check\_triple2C1=0;

check\_doubleC1=0;

check\_double2C1=0;

check\_double3C1=0;

check\_straighC1=0;

check\_flushC1=0;

check\_full\_houseC1=0;

check\_highC1=0;

straight\_tempC1=10;

check\_fh\_double\_C1=0;

check\_fh\_triple\_C1=0;

check\_tr\_triple\_C1=0;

check\_db\_double\_C1=0;

check\_2db\_double\_C1=0;

win\_cardC1=0;

//--------------computer-1(END)----------

//--------------computer-2---------------

check\_flush\_royalC2=0;

check\_straight\_flushC2=0;

check\_kareC2=0;

check\_tripleC2=0;

check\_triple2C2=0;

check\_doubleC2=0;

check\_double2C2=0;

check\_double3C2=0;

check\_straighC2=0;

check\_flushC2=0;

check\_full\_houseC2=0;

check\_highC2=0;

straight\_tempC2=10;

check\_fh\_double\_C2=0;

check\_fh\_triple\_C2=0;

check\_tr\_triple\_C2=0;

check\_db\_double\_C2=0;

check\_2db\_double\_C2=0;

win\_cardC2=0;

//--------------computer-2(END)----------

//--------------computer-3---------------

check\_flush\_royalC3=0;

check\_straight\_flushC3=0;

check\_kareC3=0;

check\_tripleC3=0;

check\_triple2C3=0;

check\_doubleC3=0;

check\_double2C3=0;

check\_double3C3=0;

check\_straighC3=0;

check\_flushC3=0;

check\_full\_houseC3=0;

check\_highC3=0;

straight\_tempC3=10;

check\_fh\_double\_C3=0;

check\_fh\_triple\_C3=0;

check\_tr\_triple\_C3=0;

check\_db\_double\_C3=0;

check\_2db\_double\_C3=0;

win\_cardC3=0;

//--------------computer-3(END)----------

//--------------computer-4---------------

check\_flush\_royalC4=0;

check\_straight\_flushC4=0;

check\_kareC4=0;

check\_tripleC4=0;

check\_triple2C4=0;

check\_doubleC4=0;

check\_double2C4=0;

check\_double3C4=0;

check\_straighC4=0;

check\_flushC4=0;

check\_full\_houseC4=0;

check\_highC4=0;

straight\_tempC4=10;

check\_fh\_double\_C4=0;

check\_fh\_triple\_C4=0;

check\_tr\_triple\_C4=0;

check\_db\_double\_C4=0;

check\_2db\_double\_C4=0;

win\_cardC4=0;

//--------------computer-4(END)----------

//--------------computer-5---------------

check\_flush\_royalC5=0;

check\_straight\_flushC5=0;

check\_kareC5=0;

check\_tripleC5=0;

check\_triple2C5=0;

check\_doubleC5=0;

check\_double2C5=0;

check\_double3C5=0;

check\_straighC5=0;

check\_flushC5=0;

check\_full\_houseC5=0;

check\_highC5=0;

straight\_tempC5=10;

check\_fh\_double\_C5=0;

check\_fh\_triple\_C5=0;

check\_tr\_triple\_C5=0;

check\_db\_double\_C5=0;

check\_2db\_double\_C5=0;

win\_cardC5=0;

//--------------computer-5(END)----------

counts\_combinations=0;

check\_temp\_card\_straight=0;

check\_temp\_high\_card=0;

moneybox=0;

}

}

//if(temp\_card==11)money=10000;

if(money>4000)

{

temp\_card=-1000;

if(temp\_card<-950)

{

CSurface::drawTextSurface(font,screen,szWin.c\_str(),230,175,255,255,0);

if(temp\_win==0)

{

Mix\_PlayChannel(-1,snd\_win,0);

temp\_win++;

}

}

//--------------player--------------------

check\_flush\_royalP=0;

check\_straight\_flushP=0;

check\_kareP=0;

check\_tripleP=0;

check\_triple2P=0;

check\_doubleP=0;

check\_double2P=0;

check\_double3P=0;

check\_straighP=0;

check\_flushP=0;

check\_full\_houseP=0;

check\_highP=0;

straight\_tempP=10;

proverkaP=0;

check\_fh\_double\_P=0;

check\_fh\_triple\_P=0;

check\_tr\_triple\_P=0;

check\_db\_double\_P=0;

check\_2db\_double\_P=0;

win\_cardP=0;

//--------------player(END)--------------

//--------------computer-1---------------

check\_flush\_royalC1=0;

check\_straight\_flushC1=0;

check\_kareC1=0;

check\_tripleC1=0;

check\_triple2C1=0;

check\_doubleC1=0;

check\_double2C1=0;

check\_double3C1=0;

check\_straighC1=0;

check\_flushC1=0;

check\_full\_houseC1=0;

check\_highC1=0;

straight\_tempC1=10;

check\_fh\_double\_C1=0;

check\_fh\_triple\_C1=0;

check\_tr\_triple\_C1=0;

check\_db\_double\_C1=0;

check\_2db\_double\_C1=0;

win\_cardC1=0;

//--------------computer-1(END)----------

//--------------computer-2---------------

check\_flush\_royalC2=0;

check\_straight\_flushC2=0;

check\_kareC2=0;

check\_tripleC2=0;

check\_triple2C2=0;

check\_doubleC2=0;

check\_double2C2=0;

check\_double3C2=0;

check\_straighC2=0;

check\_flushC2=0;

check\_full\_houseC2=0;

check\_highC2=0;

straight\_tempC2=10;

check\_fh\_double\_C2=0;

check\_fh\_triple\_C2=0;

check\_tr\_triple\_C2=0;

check\_db\_double\_C2=0;

check\_2db\_double\_C2=0;

win\_cardC2=0;

//--------------computer-2(END)----------

//--------------computer-3---------------

check\_flush\_royalC3=0;

check\_straight\_flushC3=0;

check\_kareC3=0;

check\_tripleC3=0;

check\_triple2C3=0;

check\_doubleC3=0;

check\_double2C3=0;

check\_double3C3=0;

check\_straighC3=0;

check\_flushC3=0;

check\_full\_houseC3=0;

check\_highC3=0;

straight\_tempC3=10;

check\_fh\_double\_C3=0;

check\_fh\_triple\_C3=0;

check\_tr\_triple\_C3=0;

check\_db\_double\_C3=0;

check\_2db\_double\_C3=0;

win\_cardC3=0;

//--------------computer-3(END)----------

//--------------computer-4---------------

check\_flush\_royalC4=0;

check\_straight\_flushC4=0;

check\_kareC4=0;

check\_tripleC4=0;

check\_triple2C4=0;

check\_doubleC4=0;

check\_double2C4=0;

check\_double3C4=0;

check\_straighC4=0;

check\_flushC4=0;

check\_full\_houseC4=0;

check\_highC4=0;

straight\_tempC4=10;

check\_fh\_double\_C4=0;

check\_fh\_triple\_C4=0;

check\_tr\_triple\_C4=0;

check\_db\_double\_C4=0;

check\_2db\_double\_C4=0;

win\_cardC4=0;

//--------------computer-4(END)----------

//--------------computer-5---------------

check\_flush\_royalC5=0;

check\_straight\_flushC5=0;

check\_kareC5=0;

check\_tripleC5=0;

check\_triple2C5=0;

check\_doubleC5=0;

check\_double2C5=0;

check\_double3C5=0;

check\_straighC5=0;

check\_flushC5=0;

check\_full\_houseC5=0;

check\_highC5=0;

straight\_tempC5=10;

check\_fh\_double\_C5=0;

check\_fh\_triple\_C5=0;

check\_tr\_triple\_C5=0;

check\_db\_double\_C5=0;

check\_2db\_double\_C5=0;

win\_cardC5=0;

//--------------computer-5(END)----------

counts\_combinations=0;

check\_temp\_card\_straight=0;

check\_temp\_high\_card=0;

moneybox=0;

}

}

**CEvent.cpp:**

#include "CEvent.h"

CEvent::CEvent()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CEvent::~CEvent()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onEvent(SDL\_Event\* e)

{

switch(e->type)

{

case SDL\_QUIT:

{

onExit();

break;

}

case SDL\_KEYDOWN:

{

onKeyDown(e->key.keysym.sym,e->key.keysym.mod,e->key.keysym.scancode);

break;

}

case SDL\_KEYUP:

{

onKeyUp(e->key.keysym.sym,e->key.keysym.mod,e->key.keysym.scancode);

break;

}

case SDL\_MOUSEBUTTONDOWN:

{

switch(e->button.button)

{

case SDL\_BUTTON\_LEFT:

{

onLButtonDown(e->button.x,e->button.y);

break;

}

case SDL\_BUTTON\_RIGHT:

{

onRButtonDown(e->button.x,e->button.y);

break;

}

case SDL\_BUTTON\_MIDDLE:

{

onMButtonDown(e->button.x,e->button.y);

break;

}

}

break;

}

case SDL\_MOUSEMOTION:

{

onMouseMove(e->motion.x,e->motion.y,e->motion.xrel,e->motion.yrel,(e->motion.state&SDL\_BUTTON(SDL\_BUTTON\_LEFT)) != 0,(e->motion.state&SDL\_BUTTON(SDL\_BUTTON\_RIGHT)) != 0, (e->motion.state&SDL\_BUTTON(SDL\_BUTTON\_MIDDLE)) != 0);

break;

}

case SDL\_MOUSEBUTTONUP:

{

switch(e->button.button)

{

case SDL\_BUTTON\_LEFT:

{

onLButtonUp(e->button.x,e->button.y);

break;

}

case SDL\_BUTTON\_RIGHT:

{

onRButtonUp(e->button.x,e->button.y);

break;

}

case SDL\_BUTTON\_MIDDLE:

{

onMButtonUp(e->button.x,e->button.y);

break;

}

}

break;

}

case SDL\_WINDOWEVENT:

{

switch(e->window.event)

{

case SDL\_WINDOWEVENT\_SHOWN:

{

onWindowShown(e->window.windowID);

break;

}

case SDL\_WINDOWEVENT\_HIDDEN:

{

onWindowHidden(e->window.event);

break;

}

case SDL\_WINDOWEVENT\_RESIZED:

{

onWindowResized(e->window.windowID,e->window.data1,e->window.data2);

break;

}

case SDL\_WINDOWEVENT\_SIZE\_CHANGED:

{

onWindowSizeChanged(e->window.windowID,e->window.data1,e->window.data2);

break;

}

case SDL\_WINDOWEVENT\_ENTER:

{

onMouseEnter();

break;

}

case SDL\_WINDOWEVENT\_LEAVE:

{

onMouseLeave();

break;

}

case SDL\_WINDOWEVENT\_EXPOSED:

{

onWindowExposed();

break;

}

case SDL\_WINDOWEVENT\_MOVED:

{

onWindowMove(e->window.windowID,e->window.data1,e->window.data2);

break;

}

case SDL\_WINDOWEVENT\_MAXIMIZED:

{

onWindowMaximize(e->window.windowID,e->window.data1,e->window.data2);

break;

}

case SDL\_WINDOWEVENT\_MINIMIZED:

{

onWindowMinimize(e->window.windowID,e->window.data1,e->window.data2);

break;

}

case SDL\_WINDOWEVENT\_CLOSE:

{

e->type = SDL\_QUIT;

SDL\_PushEvent(e);

break;

}

case SDL\_WINDOWEVENT\_TAKE\_FOCUS:

{

onTakeFocus();

break;

}

case SDL\_WINDOWEVENT\_RESTORED:

{

onWindowRestored();

break;

}

case SDL\_WINDOWEVENT\_FOCUS\_GAINED:

{

onInputFocus();

break;

}

case SDL\_WINDOWEVENT\_FOCUS\_LOST:

{

onInputFocusLost();

break;

}

}

break;

}

default:

{

onUser(e->user.type,e->user.code,e->user.data1,e->user.data2);

break;

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onExit()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onKeyDown(SDL\_Keycode sym, Uint16 mod, Uint16 scancode)

{

switch(sym)

{

case SDLK\_ESCAPE:

{

onExit();

break;

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onKeyUp(SDL\_Keycode sym, Uint16 mod, Uint16 scancode)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onLButtonDown(int xm,int ym)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onRButtonDown(int xm,int ym)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onMButtonDown(int xm,int ym)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onMouseMove(int xm, int ym, int relx, int rely, bool left, bool right, bool middle)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onLButtonUp(int xm,int ym)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onRButtonUp(int xm,int ym)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onMButtonUp(int xm,int ym)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onWindowShown(int winId)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onWindowHidden(int winId)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onWindowResized(int winId, int data1, int data2)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onWindowSizeChanged(int winId, int data1, int data2)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onMouseEnter()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onMouseLeave()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onWindowExposed()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onWindowMove(int winId, int data1, int data2)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onWindowMinimize(int winId, int data1, int data2)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onWindowMaximize(int winId, int data1, int data2)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onTakeFocus()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onWindowRestored()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onInputFocus()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onInputFocusLost()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CEvent::onUser(Uint8 type, int code, void\* data1, void\* data2)

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CSurface.cpp:**

#include "CSurface.h"

CSurface::CSurface()

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SDL\_Surface\* CSurface::loadSursface(char\* fname, SDL\_Surface\* screen)

{

SDL\_Surface\* temp = SDL\_LoadBMP(fname);

if(temp == NULL)

{

return NULL;

}

//----------------------------------------

SDL\_Surface\* newSurface = SDL\_ConvertSurface(temp,screen->format,0);

//------------------------------------------

SDL\_FreeSurface(temp);

return newSurface;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SDL\_Surface\* CSurface::loadSursface(char\* fname, SDL\_Surface\* screen,int r, int g, int b)

{

SDL\_Surface\* temp = SDL\_LoadBMP(fname);

if(temp == NULL)

{

return NULL;

}

//----------------------------------------

SDL\_Surface\* newSurface = SDL\_ConvertSurface(temp,screen->format,0);

SDL\_SetColorKey(newSurface, SDL\_TRUE, SDL\_MapRGB(newSurface->format,r,g,b));

//------------------------------------------

SDL\_FreeSurface(temp);

return newSurface;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CSurface::drawSurface(SDL\_Surface\* src, SDL\_Surface\* dst, int x, int y)

{

if(src == NULL || dst == NULL)

return;

SDL\_Rect rcDest;

rcDest.x = x;

rcDest.y = y;

SDL\_BlitSurface(src,NULL,dst,&rcDest);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CSurface::drawSurface(SDL\_Surface\* src, SDL\_Surface\* dst, SDL\_Rect\* clip, int x, int y)

{

if(src == NULL || dst == NULL)

return;

SDL\_Rect rcDest;

rcDest.x = x;

rcDest.y = y;

SDL\_BlitSurface(src,clip,dst,&rcDest);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CSurface::drawSurface(SDL\_Surface\* src, SDL\_Surface\* dst,SDL\_Rect\* clip, SDL\_Rect\* scale)

{

if(src == NULL || dst == NULL)

return;

SDL\_BlitSurface(src,NULL,dst,scale);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CSurface::drawTextSurface(TTF\_Font\* font,SDL\_Surface\* screen, const char\* str, int x, int y, int r, int g, int b)

{

if(font == NULL || screen == NULL)

return;

SDL\_Color color = {r,g,b};

SDL\_Rect rcDest;

rcDest.x = x;

rcDest.y = y;

SDL\_Surface\* txtSurface = TTF\_RenderText\_Solid(font,str,color);

if(txtSurface == NULL)

return;

SDL\_BlitSurface(txtSurface,NULL,screen,&rcDest);

SDL\_FreeSurface(txtSurface);

}

**Baza\_Dannih.h:**

#ifndef BAZA\_DANNIH\_H\_INCLUDED

#define BAZA\_DANNIH\_H\_INCLUDED

#include <stdio.h>

#include <cstdlib>

#include <conio.h>

#include <locale.h>

#include <string.h>

#include <windows.h>

#include <stdio.h>

#include <cstdlib>

#include <conio.h>

#include <locale.h>

#include <string.h>

#include <windows.h>

using namespace std;

void XY1(int x, int y)

{

COORD a = { x, y };

HANDLE form = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleCursorPosition(form, a);

}

void Color1(int text, int background)

{

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hStdOut, (WORD)((background << 4) | text));

}

struct grajdanin

{

struct

{

char name [15];

char fam [15];

char ot [15];

char ad [15];

char ad1 [15];

unsigned int dd,mm,gg;

char nb [15];

} gr;

} ;

struct grajdanin1

{

struct

{

char name [15];

char fam [15];

char ot [15];

char ad [15];

char ad1 [15];

char nb [15];

unsigned int dd,mm,gg;

} gr1;

} ;

char pq[ 15 ];

char pname[ 15 ];

char pustoi1[ 15 ]={'\0'};

char pustoi[ 15 ]={' '};

char null[ 15 ]={'0'};

char fff[ 15 ];

char ott[ 15 ];

char add[ 15 ];

char add1[ 15 ];

char nnb[ 15 ];

int ddd,mmm,ggg;

char finame[ 15 ];

char finame1[ 15 ];

char finame2[ 15 ];

struct grajdanin z[50];

struct grajdanin1 c[50];

unsigned short g,n,w,e,T,y=1,h,l,vv,rtt=2,ud=0;

int o=1;

void clrscr()

{

system("CLS");

}

void delete\_()

{

//ПОРЯДОЧНОСТЬ

for(int ii=1;ii<31;ii++)

{

for(int i=1;i<31;i++)

{

if(z[i].gr.dd==ii)

{

c[o].gr1.dd=z[i].gr.dd;

c[o].gr1.gg=z[i].gr.gg;

c[o].gr1.mm=z[i].gr.mm;

strcpy(c[o].gr1.name,z[i].gr.name);

strcpy(c[o].gr1.ad,z[i].gr.ad);

strcpy(c[o].gr1.ad1,z[i].gr.ad1);

strcpy(c[o].gr1.nb,z[i].gr.nb);

strcpy(c[o].gr1.ot,z[i].gr.ot);

strcpy(c[o].gr1.fam,z[i].gr.fam);

o++;

}

}

}

o=1;

//Вывод полный таблицей

clrscr();

printf ("Если телефона нет - в столбце | Номер телефона | будет 0\n");

printf ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf ("|Номер| Фамилия | Имя | Отчество | Адрес | Номер телефона |Дата рождения|\n");

for (int i=1; i<40; ++i)

{

if(c[i].gr1.dd>31||c[i].gr1.dd<1||c[i].gr1.mm>12||c[i].gr1.mm<1||c[i].gr1.gg<1||c[i].gr1.gg>2019)break;

else

{

if(!strcmp (c[i].gr1.ad,pustoi)||!strcmp (c[i].gr1.nb,pustoi)||!strcmp (c[i].gr1.ad1,pustoi)||!strcmp (c[i].gr1.name,pustoi)){}

else

{

printf ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf ("| %3i |%15s|%13s|%20s|%13s%4s|%16s| %3i %3i %4i|", i, &c[i].gr1.fam, &c[i].gr1.name, &c[i].gr1.ot, &c[i].gr1.ad, &c[i].gr1.ad1, &c[i].gr1.nb, c[i].gr1.dd, c[i].gr1.mm, c[i].gr1.gg);

printf ("\n", &i);

}

}

}

printf ("Введите номер человека которого хотите удалить:");

scanf ("%u", &vv);

strcpy(c[vv].gr1.name,pustoi);

strcpy(c[vv].gr1.fam,pustoi);

strcpy(c[vv].gr1.ot,pustoi);

strcpy(c[vv].gr1.ad,pustoi);

strcpy(c[vv].gr1.ad1,pustoi);

strcpy(c[vv].gr1.nb,pustoi);

c[vv].gr1.dd=0;

c[vv].gr1.mm=0;

c[vv].gr1.gg=0;

clrscr();

}

void save()

{

//Сохранение нового списка

printf( "Введите имя файла, в который будет записываться новая информация: " );

scanf( "%s", finame2 );

FILE \*f2 = fopen( finame2, "w" );

for (int i=1; i<15; ++i)

{

fprintf(f2, "%s\n", &c[i].gr1.name);

fprintf(f2, "%s\n", &c[i].gr1.fam);

fprintf(f2, "%s\n", &c[i].gr1.ot);

fprintf(f2, "%s\n", &c[i].gr1.ad);

fprintf(f2, "%s\n", &c[i].gr1.ad1);

fprintf(f2, "%s\n", &c[i].gr1.nb);

fprintf(f2, "%i\n", c[i].gr1.dd);

fprintf(f2, "%i\n", c[i].gr1.mm);

fprintf(f2, "%i\n", c[i].gr1.gg);

}

fclose(f2);

}

void print\_()

{

//ПОРЯДОЧНОСТЬ

for(int ii=1;ii<31;ii++)

{

for(int i=1;i<31;i++)

{

if(z[i].gr.dd==ii)

{

c[o].gr1.dd=z[i].gr.dd;

c[o].gr1.gg=z[i].gr.gg;

c[o].gr1.mm=z[i].gr.mm;

strcpy(c[o].gr1.name,z[i].gr.name);

strcpy(c[o].gr1.ad,z[i].gr.ad);

strcpy(c[o].gr1.ad1,z[i].gr.ad1);

strcpy(c[o].gr1.nb,z[i].gr.nb);

strcpy(c[o].gr1.ot,z[i].gr.ot);

strcpy(c[o].gr1.fam,z[i].gr.fam);

o++;

}

}

}

o=1;

//Вывод полный таблицей

clrscr();

printf ("Если телефона нет - в столбце | Номер телефона | будет 0\n");

printf ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf ("|Номер| Фамилия | Имя | Отчество | Адрес | Номер телефона |Дата рождения|\n");

for (int i=1; i<40; ++i)

{

if(c[i].gr1.dd>31||c[i].gr1.dd<1||c[i].gr1.mm>12||c[i].gr1.mm<1||c[i].gr1.gg<1||c[i].gr1.gg>2019)break;

else

{

if(!strcmp (c[i].gr1.ad,pustoi)||!strcmp (c[i].gr1.nb,pustoi)||!strcmp (c[i].gr1.ad1,pustoi)||!strcmp (c[i].gr1.name,pustoi)){}

else

{

printf ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf ("| %3i |%15s|%13s|%20s|%13s%4s|%16s| %3i %3i %4i|", i, &c[i].gr1.fam, &c[i].gr1.name, &c[i].gr1.ot, &c[i].gr1.ad, &c[i].gr1.ad1, &c[i].gr1.nb, c[i].gr1.dd, c[i].gr1.mm, c[i].gr1.gg);

printf ("\n", &i);

}

}

}

printf ("Для продолжения нажмите любую клавишу...\n");

\_getch();

}

void search\_()

{

//ПОРЯДОЧНОСТЬ

for(int ii=1;ii<31;ii++)

{

for(int i=1;i<31;i++)

{

if(z[i].gr.dd==ii)

{

c[o].gr1.dd=z[i].gr.dd;

c[o].gr1.gg=z[i].gr.gg;

c[o].gr1.mm=z[i].gr.mm;

strcpy(c[o].gr1.name,z[i].gr.name);

strcpy(c[o].gr1.ad,z[i].gr.ad);

strcpy(c[o].gr1.ad1,z[i].gr.ad1);

strcpy(c[o].gr1.nb,z[i].gr.nb);

strcpy(c[o].gr1.ot,z[i].gr.ot);

strcpy(c[o].gr1.fam,z[i].gr.fam);

o++;

}

}

}

o=1;

for(int t=0; t<1; t++)

{

clrscr();

printf("Введите сколько раз будет проходить поиск человека по списку: ");

scanf ("%u", &w);

if(w!=0){

printf ("Выберите из списка нужный вам вид поиска:\n1-Поиск по имени.\n2-Поиск по фамилии.\n3-Поиск по отчеству.\n4-Поиск по адресу.\n5-Поиск по номеру дома.\n6-Поиск по номеру телефона.\n7-Поиск по дню рождения.\n8-Поиск по месяцу рождения.\n9-Поиск по году рождения.\nВаш выбор: ");

scanf ("%d",&T);

clrscr();

switch (T)

{

case 1:

{

printf ("Поиск по имени:\n");

gets(pname);

for (int i=0; i<w; ++i)

{

printf( "Введите имя человека для поиска: " );

gets(pname);

for (int i=0; i<50; ++i)

{

if (!strcmp (c[i].gr1.name,pname))

{

if(!strcmp (c[i].gr1.ad,pustoi)||!strcmp (c[i].gr1.nb,pustoi)||!strcmp (c[i].gr1.ad1,pustoi)||!strcmp (c[i].gr1.name,pustoi)){}

else

{

printf ("ФИО: %s %s %s\n", &c[i].gr1.fam, &c[i].gr1.name, &c[i].gr1.ot);

printf ("Адрес: %s %s\n", &c[i].gr1.ad, &c[i].gr1.ad1);

printf ("Номер телефона: %s\n", &c[i].gr1.nb);

printf ("Дата рождения: %i %i %i\n", c[i].gr1.dd, c[i].gr1.mm, c[i].gr1.gg);

printf ("\n", &i);

}

}

}

}

break;

}

case 2:

{

printf ("Поиск по фамилии:\n");

gets(fff);

for (int i=0; i<w; ++i)

{

printf( "Введите фамилию человека для поиска: " );

gets(fff);

for (int i=0; i<50; ++i)

{

if (!strcmp (c[i].gr1.fam,fff))

{

if(!strcmp (c[i].gr1.ad,pustoi)||!strcmp (c[i].gr1.nb,pustoi)||!strcmp (c[i].gr1.ad1,pustoi)||!strcmp (c[i].gr1.name,pustoi)){}

else

{

printf ("ФИО: %s %s %s\n", &c[i].gr1.fam, &c[i].gr1.name, &c[i].gr1.ot);

printf ("Адрес: %s %s\n", &c[i].gr1.ad, &c[i].gr1.ad1);

printf ("Номер телефона: %s\n", &c[i].gr1.nb);

printf ("Дата рождения: %i %i %i\n", c[i].gr1.dd, c[i].gr1.mm, c[i].gr1.gg);

printf ("\n", &i);

}

}

}

}

break;

}

case 3:

{

printf ("Поиск по отчеству:\n");

gets(ott);

for (int i=0; i<w; ++i)

{

printf( "Введите отчество человека для поиска: " );

gets(ott);

for (int i=0; i<50; ++i)

{

if (!strcmp (c[i].gr1.ot,ott))

{

if(!strcmp (c[i].gr1.ad,pustoi)||!strcmp (c[i].gr1.nb,pustoi)||!strcmp (c[i].gr1.ad1,pustoi)||!strcmp (c[i].gr1.name,pustoi)){}

else

{

printf ("ФИО: %s %s %s\n", &c[i].gr1.fam, &c[i].gr1.name, &c[i].gr1.ot);

printf ("Адрес: %s %s\n", &c[i].gr1.ad, &c[i].gr1.ad1);

printf ("Номер телефона: %s\n", &c[i].gr1.nb);

printf ("Дата рождения: %i %i %i\n", c[i].gr1.dd, c[i].gr1.mm, c[i].gr1.gg);

printf ("\n", &i);

}

}

}

}

break;

}

case 4:

{

printf ("Поиск по адресу:\n");

gets(add);

for (int i=0; i<w; ++i)

{

printf( "Введите адрес человека для поиска: " );

gets(add);

for (int i=0; i<50; ++i)

{

if (!strcmp (c[i].gr1.ad,add))

{

if(!strcmp (c[i].gr1.ad,pustoi)||!strcmp (c[i].gr1.nb,pustoi)||!strcmp (c[i].gr1.ad1,pustoi)||!strcmp (c[i].gr1.name,pustoi)){}

else

{

printf ("ФИО: %s %s %s\n", &c[i].gr1.fam, &c[i].gr1.name, &c[i].gr1.ot);

printf ("Адрес: %s %s\n", &c[i].gr1.ad, &c[i].gr1.ad1);

printf ("Номер телефона: %s\n", &c[i].gr1.nb);

printf ("Дата рождения: %i %i %i\n", c[i].gr1.dd, c[i].gr1.mm, c[i].gr1.gg);

printf ("\n", &i);

}

}

}

}

break;

}

case 5:

{

printf ("Поиск по номеру дома:\n");

gets(add1);

for (int i=0; i<w; ++i)

{

printf( "Введите номер дома для поиска: " );

gets(add1);

for (int i=0; i<50; ++i)

{

if (!strcmp (c[i].gr1.ad1,add1))

{

if(!strcmp (c[i].gr1.ad,pustoi)||!strcmp (c[i].gr1.nb,pustoi)||!strcmp (c[i].gr1.ad1,pustoi)||!strcmp (c[i].gr1.name,pustoi)){}

else

{

printf ("ФИО: %s %s %s\n", &c[i].gr1.fam, &c[i].gr1.name, &c[i].gr1.ot);

printf ("Адрес: %s %s\n", &c[i].gr1.ad, &c[i].gr1.ad1);

printf ("Номер телефона: %s\n", &c[i].gr1.nb);

printf ("Дата рождения: %i %i %i\n", c[i].gr1.dd, c[i].gr1.mm, c[i].gr1.gg);

printf ("\n", &i);

}

}

}

}

break;

}

case 6:

{

printf ("Поиск по номеру телефона:\n");

gets(nnb);

for (int i=0; i<w; ++i)

{

printf( "Введите номер телефона для поиска: " );

gets(nnb);

for (int i=0; i<50; ++i)

{

if (!strcmp (c[i].gr1.nb,nnb))

{

if(!strcmp (c[i].gr1.ad,pustoi)||!strcmp (c[i].gr1.nb,pustoi)||!strcmp (c[i].gr1.ad1,pustoi)||!strcmp (c[i].gr1.name,pustoi)){}

else

{

printf ("ФИО: %s %s %s\n", &c[i].gr1.fam, &c[i].gr1.name, &c[i].gr1.ot);

printf ("Адрес: %s %s\n", &c[i].gr1.ad, &c[i].gr1.ad1);

printf ("Номер телефона: %s\n", &c[i].gr1.nb);

printf ("Дата рождения: %i %i %i\n", c[i].gr1.dd, c[i].gr1.mm, c[i].gr1.gg);

printf ("\n", &i);

}

}

}

}

break;

}

case 7:

{

printf ("Поиск дню рождения:\n");

gets(nnb);

for (int i=0; i<w; ++i)

{

printf( "Введите день рождения для поиска: " );

scanf("%d", &ddd);

for (int i=0; i<50; ++i)

{

if (ddd<1||ddd>31)

{

printf("Данного дня рождения не существует.\n");break;

}

else

{

if (c[i].gr1.dd==ddd)

{

if (ddd==c[i].gr1.dd){if(c[i].gr1.mm>12||c[i].gr1.mm<1||c[i].gr1.gg<1||c[i].gr1.gg>2019)break;

else

{

if(!strcmp (c[i].gr1.ad,pustoi)||!strcmp (c[i].gr1.nb,pustoi)||!strcmp (c[i].gr1.ad1,pustoi)||!strcmp (c[i].gr1.name,pustoi)){}

else

{

printf ("ФИО: %s %s %s\n", &c[i].gr1.fam, &c[i].gr1.name, &c[i].gr1.ot);

printf ("Адрес: %s %s\n", &c[i].gr1.ad, &c[i].gr1.ad1);

printf ("Номер телефона: %s\n", &c[i].gr1.nb);

printf ("Дата рождения: %i %i %i\n", c[i].gr1.dd, c[i].gr1.mm, c[i].gr1.gg);

printf ("\n", &i);

}

}

}

}

}

}

}

break;

}

case 8:

{

printf ("Поиск месяцу рождения:\n");

gets(nnb);

for (int i=0; i<w; ++i)

{

printf( "Введите месяц рождения для поиска: " );

scanf("%d", &mmm);

for (int i=0; i<50; ++i)

{

if (mmm<1||mmm>12)

{

printf("Данного месяца рождения не существует.\n");break;

}

else

{

if (c[i].gr1.mm==mmm)

{

if (mmm==c[i].gr1.mm){if(c[i].gr1.dd>31||c[i].gr1.dd<1||c[i].gr1.gg<1||c[i].gr1.gg>2019)break;

else

{

if(!strcmp (c[i].gr1.ad,pustoi)||!strcmp (c[i].gr1.nb,pustoi)||!strcmp (c[i].gr1.ad1,pustoi)||!strcmp (c[i].gr1.name,pustoi)){}

else

{

printf ("ФИО: %s %s %s\n", &c[i].gr1.fam, &c[i].gr1.name, &c[i].gr1.ot);

printf ("Адрес: %s %s\n", &c[i].gr1.ad, &c[i].gr1.ad1);

printf ("Номер телефона: %s\n", &c[i].gr1.nb);

printf ("Дата рождения: %i %i %i\n", c[i].gr1.dd, c[i].gr1.mm, c[i].gr1.gg);

printf ("\n", &i);

}

}

}

}

}

}

}

break;

}

case 9:

{

printf ("Поиск году рождения:\n");

gets(nnb);

for (int i=0; i<w; ++i)

{

printf( "Введите год рождения для поиска: " );

scanf("%d", &ggg);

for (int i=0; i<50; ++i)

{

if (ggg<1||ggg>2019)

{

printf("Данного года рождения в базе данных не существует.\n");break;

}

else

{

if (c[i].gr1.gg==ggg)

{

if (ggg==c[i].gr1.gg){if(c[i].gr1.dd>31||c[i].gr1.dd<1||c[i].gr1.mm>12||c[i].gr1.mm<1)break;

else

{

if(!strcmp (c[i].gr1.ad,pustoi)||!strcmp (c[i].gr1.nb,pustoi)||!strcmp (c[i].gr1.ad1,pustoi)||!strcmp (c[i].gr1.name,pustoi)){}

else

{

printf ("ФИО: %s %s %s\n", &c[i].gr1.fam, &c[i].gr1.name, &c[i].gr1.ot);

printf ("Адрес: %s %s\n", &c[i].gr1.ad, &c[i].gr1.ad1);

printf ("Номер телефона: %s\n", &c[i].gr1.nb);

printf ("Дата рождения: %i %i %i\n", c[i].gr1.dd, c[i].gr1.mm, c[i].gr1.gg);

printf ("\n", &i);

}

}

}

}

}

}

}

break;

}

default :printf ("Такого номера списка нет.\n");break;

}

}

}

\_getch();

}

void dobavlenie()

{

printf("Введите количество людей, которых вы хотите внести в записную книгу, если не хотите введите 0: ");

scanf ("%u", &n);

if(n>0)

{

printf( "Введите имя файла, в который будет записываться новая информация: " );

scanf( "%s", finame1 );

FILE \*f1 = fopen( finame1, "w" );

for (int i=11; i<(11+n); ++i)

{

printf ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

printf("Введите имя человека: ");

scanf ("%s", &c[i].gr1.name);

fprintf(f1, "%s\n", &c[i].gr1.name);

printf("Введите фамилию человека: ");

scanf ("%s", &c[i].gr1.fam);

fprintf(f1, "%s\n", &c[i].gr1.fam);

printf("Введите отчество человека: ");

scanf ("%s", &c[i].gr1.ot);

fprintf(f1, "%s\n", &c[i].gr1.ot);

printf("Введите улицу человека: ");

scanf ("%s", &c[i].gr1.ad);

fprintf(f1, "%s\n", &c[i].gr1.ad);

printf("Введите номер дома человека: ");

scanf ("%s", &c[i].gr1.ad1);

fprintf(f1, "%s\n", &c[i].gr1.ad1);

printf("Введите номер телефона если он есть человека, если телефона нет введите 0: ");

scanf("%s", &c[i].gr1.nb);

fprintf(f1, "%s\n", &c[i].gr1.nb);

printf("Введите день рождения человека: ");

scanf("%i", &c[i].gr1.dd);

fprintf(f1, "%i\n", c[i].gr1.dd);

printf("Введите месяц рождения человека: ");

scanf("%i", &c[i].gr1.mm);

fprintf(f1, "%i\n", c[i].gr1.mm);

printf("Введите год рождения человека: ");

scanf("%i", &c[i].gr1.gg);

fprintf(f1, "%i\n", c[i].gr1.gg);

printf ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

}

fclose(f1);

}

}

void menu1()

{

int n;

int key = 0;//начальное положение ключа

system("cls");

do

{

system("cls");

key = (key + 5) % 5;

XY1(47, 3);

printf("БАЗА ДАННЫХ");

if (key == 0)

{

XY1(46, 9);

printf("=>Добавление<=");

}

else

{

XY1(48, 9);

printf("Добавление");

}

if (key == 1)

{

XY1(47, 11);

printf("=>Удаление<=");

}

else

{

XY1(49, 11);

printf("Удаление");

}

if (key == 2)

{

XY1(48, 13);

printf("=>Вывод<=");

}

else

{

XY1(50, 13);

printf("Вывод");

}

if (key == 3)

{

XY1(48, 15);

printf("=>Поиск<=");

}

else

{

XY1(50, 15);

printf("Поиск");

}

if (key == 4)

{

XY1(48, 17);

printf("=>Назад<=");

}

else

{

XY1(50, 17);

printf("Назад");

}

{

n = \_getch();

if (n == 80) key++;

if (n == 72) key--;

}

}

while (n != 13);

switch (key)

{

case 0:

{

system("cls");

dobavlenie();

\_getch();

menu1();

break;

}

case 1:

{

system("cls");

delete\_();

menu1();

break;

}

case 2:

{

system("cls");

print\_();

menu1();

break;

}

case 3:

{

system("cls");

search\_();

\_getch();

menu1();

break;

}

{

}

case 4:

{

system("cls");

return;

menu1();

break;

}

}

key = 0;

}

int Baza\_Dannih()

{

setlocale(LC\_ALL, "RUS");

int j=1;

printf( "Введите имя файла, с которого будет считываться информация: " );

scanf( "%s", finame );

FILE \*f = fopen( finame, "r" );

for (int i=1; i<11; ++i)

{

fscanf(f, "%s", &z[i].gr.name);

fscanf(f, "%s", &z[i].gr.fam);

fscanf(f, "%s", &z[i].gr.ot);

fscanf(f, "%s", &z[i].gr.ad);

fscanf(f, "%s", &z[i].gr.ad1);

fscanf(f, "%s", &z[i].gr.nb);

fscanf(f, "%i", &z[i].gr.dd);

fscanf(f, "%i", &z[i].gr.mm);

fscanf(f, "%i", &z[i].gr.gg);

}

fclose(f);

for(int i=1;i<31;i++)

{

c[o].gr1.dd=z[i].gr.dd;

c[o].gr1.gg=z[i].gr.gg;

c[o].gr1.mm=z[i].gr.mm;

strcpy(c[o].gr1.name,z[i].gr.name);

strcpy(c[o].gr1.ad,z[i].gr.ad);

strcpy(c[o].gr1.ad1,z[i].gr.ad1);

strcpy(c[o].gr1.nb,z[i].gr.nb);

strcpy(c[o].gr1.ot,z[i].gr.ot);

strcpy(c[o].gr1.fam,z[i].gr.fam);

o++;

}

o=1;

menu1();

save();

return 0;

}

#endif // BAZA\_DANNIH\_H\_INCLUDED

**CApp.h:**

#ifndef CAPP\_H\_INCLUDED

#define CAPP\_H\_INCLUDED

#include <SDL2/SDL\_mixer.h>

#include <SDL2/SDL\_ttf.h>

#include "CEvent.h"

#include "CSurface.h"

#include "Defines.h"

#include <math.h>

#include <stdlib.h>

#include <string>

#include <time.h>

#include <iostream>

#include <windows.h>

using namespace std;

class CApp:public CEvent

{

public:

CApp();

virtual ~CApp();

int onExecute();

virtual void onEvent(SDL\_Event\* e);//Обработка событий

virtual void onExit();

virtual void onLButtonDown(int xm, int ym);

private:

bool running;

SDL\_Window \*window;

SDL\_Surface\* screen;

SDL\_Surface\* background;

//--------------CARDS--------------------

SDL\_Surface\* cart\_2\_spades;

SDL\_Surface\* cart\_2\_clubs;

SDL\_Surface\* cart\_2\_hearts;

SDL\_Surface\* cart\_2\_diamonds;

SDL\_Surface\* cart\_3\_spades;

SDL\_Surface\* cart\_3\_clubs;

SDL\_Surface\* cart\_3\_hearts;

SDL\_Surface\* cart\_3\_diamonds;

SDL\_Surface\* cart\_4\_spades;

SDL\_Surface\* cart\_4\_clubs;

SDL\_Surface\* cart\_4\_hearts;

SDL\_Surface\* cart\_4\_diamonds;

SDL\_Surface\* cart\_5\_spades;

SDL\_Surface\* cart\_5\_clubs;

SDL\_Surface\* cart\_5\_hearts;

SDL\_Surface\* cart\_5\_diamonds;

SDL\_Surface\* cart\_6\_spades;

SDL\_Surface\* cart\_6\_clubs;

SDL\_Surface\* cart\_6\_hearts;

SDL\_Surface\* cart\_6\_diamonds;

SDL\_Surface\* cart\_7\_spades;

SDL\_Surface\* cart\_7\_clubs;

SDL\_Surface\* cart\_7\_hearts;

SDL\_Surface\* cart\_7\_diamonds;

SDL\_Surface\* cart\_8\_spades;

SDL\_Surface\* cart\_8\_clubs;

SDL\_Surface\* cart\_8\_hearts;

SDL\_Surface\* cart\_8\_diamonds;

SDL\_Surface\* cart\_9\_spades;

SDL\_Surface\* cart\_9\_clubs;

SDL\_Surface\* cart\_9\_hearts;

SDL\_Surface\* cart\_9\_diamonds;

SDL\_Surface\* cart\_10\_spades;

SDL\_Surface\* cart\_10\_clubs;

SDL\_Surface\* cart\_10\_hearts;

SDL\_Surface\* cart\_10\_diamonds;

SDL\_Surface\* cart\_J\_spades;

SDL\_Surface\* cart\_J\_clubs;

SDL\_Surface\* cart\_J\_hearts;

SDL\_Surface\* cart\_J\_diamonds;

SDL\_Surface\* cart\_Q\_spades;

SDL\_Surface\* cart\_Q\_clubs;

SDL\_Surface\* cart\_Q\_hearts;

SDL\_Surface\* cart\_Q\_diamonds;

SDL\_Surface\* cart\_K\_spades;

SDL\_Surface\* cart\_K\_clubs;

SDL\_Surface\* cart\_K\_hearts;

SDL\_Surface\* cart\_K\_diamonds;

SDL\_Surface\* cart\_A\_spades;

SDL\_Surface\* cart\_A\_clubs;

SDL\_Surface\* cart\_A\_hearts;

SDL\_Surface\* cart\_A\_diamonds;

SDL\_Surface\* Back\_Card;

//--------------CARDS--------------------

SDL\_Surface\* queue\_cards[52];

//--------------CARDS--------------------

Mix\_Chunk\* snd\_draw;

Mix\_Chunk\* snd\_lose;

Mix\_Chunk\* snd\_win;

Mix\_Chunk\* snd\_place;

TTF\_Font\* font;

//--------------CARDS--------------------

int temp\_card=0;

int money=2000;

int moneybox=0;

int card\_1\_player;

int card\_2\_player;

int card\_1\_computer1;

int card\_2\_computer1;

int card\_1\_computer2;

int card\_2\_computer2;

int card\_1\_computer3;

int card\_2\_computer3;

int card\_1\_computer4;

int card\_2\_computer4;

int card\_1\_computer5;

int card\_2\_computer5;

int card\_1\_all;

int card\_2\_all;

int card\_3\_all;

int card\_4\_all;

int card\_5\_all;

int card\_1\_temp;

int card\_2\_temp;

int card\_3\_temp;

int counts\_combinations=0;

int check\_temp\_card\_straight=0;

int check\_temp\_high\_card=0;

int round\_card=0;

//--------------player--------------------

int check\_flush\_royalP=0;

int check\_straight\_flushP=0;

int check\_kareP=0;

int check\_tripleP=0;

int check\_triple2P=0;

int check\_doubleP=0;

int check\_double2P=0;

int check\_double3P=0;

int check\_straighP=0;

int check\_flushP=0;

int check\_full\_houseP=0;

int check\_highP=0;

int straight\_tempP=11;

int proverkaP=0;

int check\_fh\_double\_P=0;

int check\_fh\_triple\_P=0;

int check\_tr\_triple\_P=0;

int check\_db\_double\_P=0;

int check\_2db\_double\_P=0;

int temp\_lose=0;

int temp\_win=0;

int win\_cardP=0;

//--------------player(END)--------------

//--------------computer-1---------------

int check\_flush\_royalC1=0;

int check\_straight\_flushC1=0;

int check\_kareC1=0;

int check\_tripleC1=0;

int check\_triple2C1=0;

int check\_doubleC1=0;

int check\_double2C1=0;

int check\_double3C1=0;

int check\_straighC1=0;

int check\_flushC1=0;

int check\_full\_houseC1=0;

int check\_highC1=0;

int straight\_tempC1=11;

int check\_fh\_double\_C1=0;

int check\_fh\_triple\_C1=0;

int check\_tr\_triple\_C1=0;

int check\_db\_double\_C1=0;

int check\_2db\_double\_C1=0;

int win\_cardC1=0;

int proverkaT=0;

//--------------computer-1(END)----------

//--------------computer-2---------------

int check\_flush\_royalC2=0;

int check\_straight\_flushC2=0;

int check\_kareC2=0;

int check\_tripleC2=0;

int check\_triple2C2=0;

int check\_doubleC2=0;

int check\_double2C2=0;

int check\_double3C2=0;

int check\_straighC2=0;

int check\_flushC2=0;

int check\_full\_houseC2=0;

int check\_highC2=0;

int straight\_tempC2=11;

int check\_fh\_double\_C2=0;

int check\_fh\_triple\_C2=0;

int check\_tr\_triple\_C2=0;

int check\_db\_double\_C2=0;

int check\_2db\_double\_C2=0;

int win\_cardC2=0;

//--------------computer-2(END)----------

//--------------computer-3---------------

int check\_flush\_royalC3=0;

int check\_straight\_flushC3=0;

int check\_kareC3=0;

int check\_tripleC3=0;

int check\_triple2C3=0;

int check\_doubleC3=0;

int check\_double2C3=0;

int check\_double3C3=0;

int check\_straighC3=0;

int check\_flushC3=0;

int check\_full\_houseC3=0;

int check\_highC3=0;

int straight\_tempC3=11;

int check\_fh\_double\_C3=0;

int check\_fh\_triple\_C3=0;

int check\_tr\_triple\_C3=0;

int check\_db\_double\_C3=0;

int check\_2db\_double\_C3=0;

int win\_cardC3=0;

//--------------computer-3(END)----------

//--------------computer-4---------------

int check\_flush\_royalC4=0;

int check\_straight\_flushC4=0;

int check\_kareC4=0;

int check\_tripleC4=0;

int check\_triple2C4=0;

int check\_doubleC4=0;

int check\_double2C4=0;

int check\_double3C4=0;

int check\_straighC4=0;

int check\_flushC4=0;

int check\_full\_houseC4=0;

int check\_highC4=0;

int straight\_tempC4=11;

int check\_fh\_double\_C4=0;

int check\_fh\_triple\_C4=0;

int check\_tr\_triple\_C4=0;

int check\_db\_double\_C4=0;

int check\_2db\_double\_C4=0;

int win\_cardC4=0;

//--------------computer-4(END)----------

//--------------computer-5---------------

int check\_flush\_royalC5=0;

int check\_straight\_flushC5=0;

int check\_kareC5=0;

int check\_tripleC5=0;

int check\_triple2C5=0;

int check\_doubleC5=0;

int check\_double2C5=0;

int check\_double3C5=0;

int check\_straighC5=0;

int check\_flushC5=0;

int check\_full\_houseC5=0;

int check\_highC5=0;

int straight\_tempC5=11;

int check\_fh\_double\_C5=0;

int check\_fh\_triple\_C5=0;

int check\_tr\_triple\_C5=0;

int check\_db\_double\_C5=0;

int check\_2db\_double\_C5=0;

int win\_cardC5=0;

//--------------computer-5(END)----------

//--------------CARDS--------------------

bool onInit();//Возвращает значение 1 или 0 в зависимости запустилась ли программа

void onLoop();//Обновление игрового процесса

void onRender();//Перерисовка

void onQuit();//Закрытие игры

void field\_init();

void field\_draw();

//--------------CARDS--------------------

void random\_card\_2\_player();

void random\_card\_1\_computer1();

void random\_card\_2\_computer1();

void random\_card\_1\_computer2();

void random\_card\_2\_computer2();

void random\_card\_1\_computer3();

void random\_card\_2\_computer3();

void random\_card\_1\_computer4();

void random\_card\_2\_computer4();

void random\_card\_1\_computer5();

void random\_card\_2\_computer5();

void random\_card\_1\_all();

void random\_card\_2\_all();

void random\_card\_3\_all();

void random\_card\_4\_all();

void random\_card\_5\_all();

void fold\_card();

void check\_card();

void bet\_card();

void combinations\_check(int card1, int card2, int \*check\_flush\_royal, int \*check\_straight\_flush, int \*check\_kare, int \*check\_triple, int \*check\_triple2, int \*check\_double, int \*check\_double2, int \*check\_double3, int \*check\_straigh, int \*check\_flush, int \*check\_full\_house, int \*check\_high, int \*straight\_temp);

void combinations\_double(int card1, int card2, int \*check\_double, int \*check\_double2, int \*check\_double3);

void combinations\_triple(int card1, int card2, int card3, int \*check\_triple, int \*check\_triple2);

void combinations\_kare(int card1, int card2, int card3, int card4, int \*check\_kare);

void combinations\_straight\_flash(int card1, int card2, int card3, int card4, int card5, int \*check\_flush\_royal, int \*check\_straight\_flush);

void combinations\_flash(int card1, int card2, int card3, int card4, int card5, int \*check\_flush);

void combinations\_straight(int card1, int card2, int card3, int card4, int card5, int NS, int NS1, int temp, int \*check\_straigh);

void combinations\_full\_house(int \*check\_full\_house, int \*check\_double, int \*check\_double2, int \*check\_double3, int \*check\_triple, int \*check\_triple2);

int high\_card(int card1);

void compare\_temp(int temp1,int temp2,int temp3,int temp4,int temp5,int temp6,int temp7, int \*check\_high);

void high\_card\_all(int card1,int card2, int \*check\_high);

void check\_card\_win();

void check\_card\_win(int \*check\_flush\_royalP,int \*check\_straight\_flushP,int \*check\_kareP,int \*check\_tripleP,int \*check\_triple2P,int \*check\_doubleP,int \*check\_double2P,

int \*check\_double3P,int \*check\_straighP,int \*check\_flushP,int \*check\_full\_houseP,int \*check\_highP,int \*straight\_tempP,int \*check\_fh\_double\_P,int \*check\_fh\_triple\_P,

int \*check\_tr\_triple\_P,int \*check\_db\_double\_P,int \*check\_2db\_double\_P,int \*win\_cardP,

int \*check\_flush\_royalC1,int \*check\_straight\_flushC1,int \*check\_kareC1,int \*check\_tripleC1,int \*check\_triple2C1,int \*check\_doubleC1,

int \*check\_double2C1,int \*check\_double3C1,int \*check\_straighC1,int \*check\_flushC1,int \*check\_full\_houseC1,int \*check\_highC1,int \*straight\_tempC1,int \*check\_fh\_double\_C1,

int \*check\_fh\_triple\_C1,int \*check\_tr\_triple\_C1,int \*check\_db\_double\_C1,int \*check\_2db\_double\_C1,int \*win\_cardC1,

int \*check\_flush\_royalC2,int \*check\_straight\_flushC2,int \*check\_kareC2,int \*check\_tripleC2,int \*check\_triple2C2,int \*check\_doubleC2,int \*check\_double2C2,int \*check\_double3C2,

int \*check\_straighC2,int \*check\_flushC2,int \*check\_full\_houseC2,int \*check\_highC2,int \*straight\_tempC2,int \*check\_fh\_double\_C2,int \*check\_fh\_triple\_C2,int \*check\_tr\_triple\_C2,

int \*check\_db\_double\_C2,int \*check\_2db\_double\_C2,int \*win\_cardC2,

int \*check\_flush\_royalC3,int \*check\_straight\_flushC3,int \*check\_kareC3,int \*check\_tripleC3,int \*check\_triple2C3,int \*check\_doubleC3,int \*check\_double2C3,

int \*check\_double3C3,int \*check\_straighC3,int \*check\_flushC3,int \*check\_full\_houseC3,int \*check\_highC3,int \*straight\_tempC3,int \*check\_fh\_double\_C3,int \*check\_fh\_triple\_C3,int \*check\_tr\_triple\_C3,

int \*check\_db\_double\_C3,int \*check\_2db\_double\_C3,int \*win\_cardC3,

int \*check\_flush\_royalC4,int \*check\_straight\_flushC4,int \*check\_kareC4,int \*check\_tripleC4,int \*check\_triple2C4,int \*check\_doubleC4,int \*check\_double2C4,int \*check\_double3C4,

int \*check\_straighC4,int \*check\_flushC4,int \*check\_full\_houseC4,int \*check\_highC4,int \*straight\_tempC4,int \*check\_fh\_double\_C4,int \*check\_fh\_triple\_C4,int \*check\_tr\_triple\_C4,

int \*check\_db\_double\_C4,int \*check\_2db\_double\_C4,int \*win\_cardC4,

int \*check\_flush\_royalC5,int \*check\_straight\_flushC5,int \*check\_kareC5,int \*check\_tripleC5,int \*check\_triple2C5,int \*check\_doubleC5,int \*check\_double2C5,int \*check\_double3C5,int \*check\_straighC5,

int \*check\_flushC5,int \*check\_full\_houseC5,int \*check\_highC5,int \*straight\_tempC5,int \*check\_fh\_double\_C5,int \*check\_fh\_triple\_C5,int \*check\_tr\_triple\_C5,

int \*check\_db\_double\_C5,int \*check\_2db\_double\_C5,int \*win\_cardC5);

//--------------CARDS--------------------

bool loadMedia();

};

#endif // CAPP\_H\_INCLUDED

**CEvent.h:**

#ifndef CEVENT\_H\_INCLUDED

#define CEVENT\_H\_INCLUDED

#include <SDL2/SDL.h>

class CEvent

{

public:

CEvent();

virtual ~CEvent();

virtual void onEvent(SDL\_Event\* e);

virtual void onExit();

virtual void onKeyDown(SDL\_Keycode sym, Uint16 mod, Uint16 scancode);

virtual void onKeyUp(SDL\_Keycode sym, Uint16 mod, Uint16 scancode);

virtual void onLButtonDown(int xm,int ym);

virtual void onRButtonDown(int xm,int ym);

virtual void onMButtonDown(int xm,int ym);

virtual void onMouseMove(int xm, int ym, int relx, int rely, bool left, bool right, bool middle);

virtual void onLButtonUp(int xm,int ym);

virtual void onRButtonUp(int xm,int ym);

virtual void onMButtonUp(int xm,int ym);

virtual void onWindowShown(int winId);

virtual void onWindowHidden(int winId);

virtual void onWindowResized(int winId, int data1, int data2);

virtual void onWindowSizeChanged(int winId, int data1, int data2);

virtual void onWindowExposed();

virtual void onWindowMove(int winId, int data1, int data2);

virtual void onWindowMinimize(int winId, int data1, int data2);

virtual void onWindowMaximize(int winId, int data1, int data2);

virtual void onTakeFocus();

virtual void onWindowRestored();

virtual void onInputFocus();

virtual void onInputFocusLost();

virtual void onMouseEnter();

virtual void onMouseLeave();

virtual void onUser(Uint8 type, int code, void\* data1, void\* data2);

private:

};

#endif // CEVENT\_H\_INCLUDED

**CEvent.h:**

#ifndef CSURFACE\_H\_INCLUDED

#define CSURFACE\_H\_INCLUDED

#include <SDL2/SDL.h>

#include <SDL2/SDL\_ttf.h>

class CSurface

{

public:

CSurface();

static SDL\_Surface\* loadSursface(char\* fname, SDL\_Surface\* screen);

static SDL\_Surface\* loadSursface(char\* fname, SDL\_Surface\* screen, int r, int g, int b);

static void drawSurface(SDL\_Surface\* src, SDL\_Surface\* dst, SDL\_Rect\* clip, int x, int y);

static void drawSurface(SDL\_Surface\* src, SDL\_Surface\* dst, int x, int y);

static void drawSurface(SDL\_Surface\* src, SDL\_Surface\* dst, SDL\_Rect\* clip, SDL\_Rect\* scale);

static void drawTextSurface(TTF\_Font\* font,SDL\_Surface\* screen, const char\* str, int x, int y, int r, int g, int b);

private:

};

#endif // CSURFACE\_H\_INCLUDED

**Defines.h:**

#define WINDOW\_WIDTH 640

#define WINDOW\_HEIGHT 480

#define WINDOW\_CAPTION "Poker"

#define \_WIN32\_WINNT 0x0500

//==================== Field constants ==

#define FIELD\_LEFT 208

#define FIELD\_TOP 32

#define CARD\_LEFT 451 //63

#define CARD\_TOP 449 //31