

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN



## ĐỒ ÁN CUỐI KỲ CHỦ ĐỀ: POKER

Môn: Cơ sở lập trình  
Lớp: 24C03  
Nhóm: 8  
Tên nhóm: Triple Star

Thành phố Hồ Chí Minh, 12-2024

# Contents

<b>I.CHƯƠNG TRÌNH POKER</b>	<b>6</b>
<b>1. Luật chơi:</b>	<b>6</b>
a. Luật chơi Poker mode PVP: . . . . .	6
b. Luật chơi Poker mode PVE: . . . . .	6
c. Luật chơi Poker mode Single Card Duel: . . . . .	6
d. Luật chơi Poker mode 3 Card Poker (Bài cào): . . . . .	7
<b>2. Các tính năng của các thư viện chương trình chương trình:</b>	<b>7</b>
a. Các thư viện chuẩn: . . . . .	7
b. Các thư viện tự khởi tạo: . . . . .	8
<b>3. Các hàm trong thư viện tự khởi tạo:</b>	<b>8</b>
a. Thư viện "52_cards.h": . . . . .	8
b. Thư viện "game_database.h": . . . . .	11
c. Thư viện "poker_desk.h": . . . . .	12
d. Thư viện "gameMode.h": . . . . .	19
e. Thư viện "file_manipulation.h": . . . . .	24
<b>4. Cách hoạt động của chương trình</b>	<b>25</b>
a. Màn hình chính: . . . . .	25
<b>II. THAM KHẢO</b>	<b>35</b>
<b>1. Các nguồn trên mạng:</b>	<b>35</b>
a. Code: . . . . .	35
b. Latex: . . . . .	35
c. Luật chơi Poker: . . . . .	35
<b>2. Các công cụ AI hỗ trợ:</b>	<b>35</b>
<b>3. Con người:</b>	<b>36</b>

## TÓM TẮT ĐỒ ÁN

Tài liệu này cung cấp chi tiết cho chương trình Poker viết bằng ngôn ngữ C++ gồm luật chơi mỗi chế độ, các tính năng của chương trình và cách hoạt động của chương trình.

## THÀNH VIÊN NHÓM

Họ Và Tên	MSSV	Vai trò
Hoàng Cao Phong	24127486	Trưởng nhóm
Nguyễn Trung Kiên	24127068	Thành viên
Võ Thúc Trí	24127570	Thành viên

## ĐÓNG GÓP CÁC THÀNH VIÊN

	rubric	24127486	24127068	24127570
Standard feature	-			
Initialization	20	20	0	0
Gameplay	20	10	0	10
player's information	20	20	0	0
Advance features	-			
PVE game	30	10	10	10
Color/ Sound effects	30	0	0	30
Leaderboards	30	15	5	10
Creatioins	30/each			
Single Card Duel	30	15	5	10
Three Card Poker	30	10	10	10
Report	-			
	100	10	70	20
latex	10	0	10	0
	Total:	110	110	100

# I.CHƯƠNG TRÌNH POKER

## 1. Luật chơi:

### a. Luật chơi Poker mode PVP:

- Chương trình tự động phát 5 lá bài ngẫu nhiên cho từng người.
- Bài sẽ được xét theo hạng  $2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < J < Q < K < A$ .Chương trình sẽ xét bài của người chơi theo trật tự từ cao xuống thấp sau:

Thứ tự	Tên trường hợp	Mô tả
1	Sảnh đồng chất (Straight Flush)	5 lá bài liên tiếp, cùng chất.
2	Tứ quý (Four of a Kind)	4 lá bài cùng số.
3	Cù lũ (Full House)	3 lá bài cùng số và 1 đôi.
4	Thùng (Flush)	5 lá bài cùng chất, không cần liên tiếp.
5	Sảnh (Straight)	5 lá bài liên tiếp, không cần cùng chất.
6	Bộ ba (Three of a Kind)	3 lá bài cùng số.
7	Hai đôi (Two Pairs)	Hai cặp bài khác nhau.
8	Một đôi (One Pair)	Một cặp bài giống nhau.
9	Bài cao (High Card)	Lá bài cao nhất.

- Chất lá bài là **không quan trọng** (chỉ xét đến hạng của lá bài).
- Bộ bài của những người nào có bộ bài mang giá trị cao nhất sẽ chiến thắng

### b. Luật chơi Poker mode PVE:

- Người chơi có quyền được chọn chế độ đấu với máy (Normal, Challenge, Legendary).
- Chương trình tự động phát 5 lá bài ngẫu nhiên cho người chơi và máy.
- Chương trình sẽ xét bài của người chơi tương tự với mode PVP
- Người chơi chiến thắng sẽ cộng 1 điểm, nếu không thì máy sẽ cộng 1 điểm.

### c. Luật chơi Poker mode Single Card Duel:

- Chương trình phát duy nhất 1 lá bài duy nhất cho từng người chơi.

- Bài sẽ được xét theo hạng  $2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < J < Q < K$ . Chất bài Bích  $>$  Cơ  $>$  rô  $>$  Chuồn.
- So sánh hạng bài của người chơi, nếu bằng nhau so hạng bài. Bài người nào có chất mạnh hơn là thắng.

#### **d. Luật chơi Poker mode 3 Card Poker (Bài cào):**

- Chương trình phát 3 lá bài cho từng người chơi.
- Điểm dựa trên lá bài (ví dụ lá 6 sẽ là sáu điểm), nếu là con át sẽ là 1 điểm, nếu lá bài đó là 10, J, Q, K thì được tính là 0 điểm.
- Bài sẽ được xét theo tổng điểm 3 lá bài và lấy số hàng đơn vị của lá bài đó (Ví dụ: Bài người chơi có 3 con 6 rô, 9 cơ, K Bích thì điểm là  $6+9+0 = 15$ . Quy ra được 5 điểm)
- So sánh điểm của từng người chơi, nếu người chơi chiến thắng nếu có số điểm cao nhất.

## **2. Các tính năng của các thư viện chương trình**

### **chương trình:**

#### **a. Các thư viện chuẩn:**

- `<iostream>`: Thư viện nhập xuất chuẩn của chương trình.
- `<vector>`: Thư viện sử dụng mảng động.
- `<fstream>`: Thư viện xuất nhập file.
- `<string>/<cstring>`: 2 thư viện dùng để liên quan đến chuỗi.
- `<sstream>`: Thư viện xử lý chuỗi
- `<map>`: Thư viện xử lý map
- `<algorithm>`: Thư viện chứa các thuật toán có sẵn cần thiết cho chương trình.
- `<iomanip>`: Thư viện hỗ trợ nhập xuất chuẩn cho chương trình.

## b. Các thư viện tự khởi tạo:

- "52\_cards.h": Thư viện này bao gồm các hàm liên quan đến việc khởi tạo và sử dụng bộ bài 52 lá.
- "game\_database.h": Thư viện này bao gồm các hàm liên quan đến việc xây dựng Database cho trò chơi.
- "poker\_desk.h": Thư viện này bao gồm các hàm liên quan đến việc nhập xuất và hiển thị thông tin.
- "gameMode.h": Thư viện này bao gồm các hàm liên quan đến việc khởi tạo và thực thi các chế độ game.
- "file\_manipulation.h": Thư viện này bao gồm các hàm liên quan đến việc nhập xuất file
- "t3\_game\_library/t3\_library.h": Thư viện tổng hợp toàn bộ thư viện khác.

## 3. Các hàm trong thư viện tự khởi tạo:

### a. Thư viện "52\_cards.h":

Thư viện này bao gồm các các hàm:

- **vector <string> return\_pack ():**

**Hàm 'return\_pack()'** Hàm này tạo ra một bộ bài 52 lá và trả về nó dưới dạng một vector các chuỗi.

1. Khởi tạo: - Khai báo một vector tên là 'pack' để lưu trữ các chuỗi đại diện cho các lá bài.
2. Vòng lặp tạo lá bài: - Một vòng lặp 'for' lặp qua 52 lần, đại diện cho mỗi lá bài trong bộ bài. - Trong mỗi lần lặp: - Xác định cấp số của lá bài hiện tại bằng cách sử dụng toán tử modulo ('- Nếu số dư là 10, 11, 12, 0 hoặc 1, lá bài là "10", "J", "Q", "K" hoặc "A" tương ứng. - Nếu không, số dư được chuyển đổi thành một ký tự đại diện cho cấp số số (2-9). - Thêm chất của lá bài vào chuỗi lá bài dựa trên lần lặp hiện tại: - Nếu lần lặp nằm trong khoảng từ 1 đến 13, chất là "spade". - Nếu lần lặp nằm trong khoảng từ 14 đến 26, chất là "club". - Nếu lần lặp nằm trong khoảng từ 27 đến 39, chất là "diamond". - Nếu lần lặp nằm trong khoảng từ 40 đến 52, chất là "heart". - Chuỗi lá bài hoàn chỉnh được thêm vào vector 'pack'.
3. Trả về: - Hàm trả về vector 'pack' chứa 52 chuỗi đại diện cho các lá bài.



- **void rank\_player\_cards(vector<vector<string>>&player\_cards, int number\_player, int ranks[ ], int suits[ ]):** Hàm sắp xếp lá bài của các người chơi, quá trình bao gồm:
  - Xác định quân và chất của mỗi lá bài:
    - \* Duyệt qua từng người chơi.
    - \* Duyệt qua 5 lá bài của mỗi người chơi.
    - \* Sử dụng hàm ‘get\_rank’ và ‘get\_suit’ để xác định quân số và chất của mỗi lá bài.
    - \* Lưu trữ thông tin quân số và chất vào các mảng ‘ranks’ và ‘suits’ tương ứng.
  - Sắp xếp các lá bài:
    - \* Duyệt qua từng người chơi.
    - \* Sử dụng thuật toán sắp xếp nổi bọt (bubble sort) để sắp xếp các lá bài của mỗi người chơi theo thứ tự tăng dần của quân số.
    - \* So sánh cặp lá bài liên tiếp. Nếu quân số của lá bài thứ nhất lớn hơn quân số của lá bài thứ hai, thì hoán đổi vị trí của hai lá bài đó trong cả mảng ‘player\_cards’ và mảng ‘ranks’.
- **std::vector<std::vector<std::string>>deal\_cards(...):** Chia bài
  - Khởi tạo mảng: Tạo một vector 2 chiều để lưu trữ bài của từng người chơi.
  - Chia bài:
    - \* Duyệt qua từng lá bài trong bộ bài đã xáo trộn.
    - \* Sử dụng phép toán modulo để xác định người chơi tiếp theo nhận bài.
    - \* Thêm lá bài hiện tại vào vector bài của người chơi đó.
- **vector <string> return \_shuffled\_pack (vector <string> shuffled\_pack):** Hàm chia bài cho n người chơi
  1. Khởi tạo bộ sinh số ngẫu nhiên:
    - ‘srand(time(0))’: Hàm ‘srand’ được gọi với đối số là ‘time(0)’. Điều này sẽ giúp đảm bảo rằng mỗi lần chạy chương trình, ta sẽ nhận được một dãy số ngẫu nhiên khác nhau.
  2. Xáo trộn bộ bài:
    - ‘std::random\_shuffle(shuffled\_pack.begin(), shuffled\_pack.end())’: Hàm này sẽ sắp xếp ngẫu nhiên các phần tử trong vector ‘shuffled\_pack’ dựa trên bộ sinh số ngẫu nhiên đã được khởi tạo ở bước trước.
  3. Trả về kết quả:
    - Hàm trả về vector ‘shuffled\_pack’ sau khi đã được xáo trộn.

- **vector<vector<string>>deal\_cards\_Single\_Card\_Duel(const vector<string>& shuffled\_pack, int number\_player):** Hàm chia bài cho n người chơi chế độ Single Card Duel.

1. Khởi tạo:

- Tạo một vector hai chiều player\_cards để lưu trữ các lá bài của mỗi người chơi. Kích thước của vector này bằng số lượng người chơi.

2. Chia bài:

- Duyệt qua từng người chơi.
- Lấy lá bài đầu tiên từ bộ bài đã xáo trộn.
- Thêm lá bài này vào vector của người chơi hiện tại.

3. Trả về kết quả:

- Trả về vector player\_cards chứa các lá bài đã được chia cho mỗi người chơi.

- **int get\_rank(const std::string& card):** Hàm lấy giá trị thứ hạng từ chuỗi.

1. Tách thứ hạng của lá bài:

- Hàm substr được sử dụng để trích xuất phần biểu thị thứ hạng của lá bài từ chuỗi card. Phần này nằm trước khoảng trắng đầu tiên.
- Kết quả được gán vào biến rank\_str.

2. Tạo một map ánh xạ thứ hạng:

- Một std::map được tạo để ánh xạ giữa biểu diễn chuỗi của thứ hạng (ví dụ: "2", "J", "A") và giá trị số tương ứng.

3. Trả về giá trị thứ hạng:

- Giá trị chất của lá bài được lấy từ suit\_map dựa trên khóa suit\_str và được trả về.

- **int get\_suit(const std::string& card):** Hàm lấy giá trị chất của chuỗi.

1. Tách chất của lá bài:

- Hàm substr được sử dụng để trích xuất phần biểu thị chất của lá bài từ chuỗi card. Phần này nằm sau khoảng trắng đầu tiên.
- Kết quả được gán vào biến suit\_str.

2. Tạo một map ánh xạ chất:

- Một std::map được tạo để ánh xạ giữa biểu diễn chuỗi của chất (ví dụ: "spade", "heart", "diamond", "club") và giá trị số tương ứng.

3. Trả về giá trị chất:

- Giá trị chất của lá bài được lấy từ suit\_map dựa trên khóa suit\_str và được trả về.

- **int get\_card\_value(const std::string& card):** Lấy giá trị điểm của lá bài (chỉ tính chữ số hàng đơn vị) trong Bài cào:

1. Lấy thứ hạng:

- Gọi hàm `get_rank` để lấy thứ hạng của lá bài. Thứ hạng này là một số nguyên từ 1 đến 13 tương ứng với các lá bài từ 2 đến Át.

2. Tính giá trị điểm:

- Các lá bài từ 2 đến 10: Giá trị điểm bằng chính thứ hạng của lá bài.
- Các lá bài J, Q, K: Giá trị điểm đều là 10.
- Toán tử `% 10`: Dùng để lấy phần dư khi chia cho 10. Điều này đảm bảo rằng giá trị điểm luôn nằm trong khoảng từ 0 đến 9.

## b. Thư viện "game\_database.h":

Thư viện này xây dựng Database cho người chơi bao gồm:

- **enum Hand\_Rank**: Phân loại bài.

```
//Phân loại các mức bài
enum Hand_Rank {
    HIGH_CARD = 0,
    PAIR,
    TWO_PAIR,
    THREE_OF_A_KIND,
    STRAIGHT,
    FLUSH,
    FULL_HOUSE,
    FOUR_OF_A_KIND,
    STRAIGHT_FLUSH
};
```

- **struct Player**: Cấu trúc dữ liệu người chơi bao gồm số thứ tự người chơi, tổng số màn chơi, số lần thắng, tỉ lệ thắng, dạng thắng nhiều nhất.

```
struct Player {
    int number; // Player's number (1-based index)
    int total_games = 0; // Total games played
    int wins = 0; // Total wins
    float win_rate = 0.0; // Win rate in percentage
    std::map<std::string, int> win_situations; // Track win types (card value & suit)
};
```

- **enum Suit**:

```
// Enums for suits
enum Suit { CLUBS, DIAMONDS, HEARTS, SPADES };
```

- **struct Card:**

```
// Struct for representing a card
struct Card
{
    int value; // 2-14 (where 11 = J, 12 = Q, 13 = K, 14 = A)
    Suit suit; // Suit enumeration
};
```

### c. Thư viện "poker\_desk.h":

Thư viện bao gồm các hàm liên quan đến giao diện game như việc nhập xuất và hiển thị thông tin. Thư viện này bao gồm các hàm:

Các hàm nhập, kiểm tra thông tin:

- **int input\_number\_of\_players():** Chức năng nhập số lượng người chơi
  1. Khai báo biến:
    - Khai báo biến `number_player` để lưu trữ số lượng người chơi nhập vào.
  2. Yêu cầu nhập liệu:
    - In ra màn hình một thông báo yêu cầu người dùng nhập số lượng người chơi.
  3. Kiểm tra đầu vào:
    - Dùng vòng lặp `while` để kiểm tra tính hợp lệ của đầu vào.
    - Nếu người dùng không nhập một số nguyên hoặc số lượng người chơi không nằm trong khoảng từ 2 đến 9, thì in ra thông báo lỗi và yêu cầu nhập lại.
    - Dùng `std::cin.clear()` để xóa cờ lỗi của luồng nhập và `std::cin.ignore()` để bỏ qua các ký tự không hợp lệ.
  4. In hình ASCII:
    - Nếu đầu vào hợp lệ, in ra một hình ASCII.
  5. Trả về số lượng người chơi:

- Trả về giá trị `number_player` đã được nhập và kiểm tra hợp lệ.
- **`bool check_Yes_No_Input()`**: Yêu cầu người dùng nhập Có hoặc Không
  1. Khai báo biến:
    - Khai báo biến `input` kiểu `char` để lưu trữ ký tự nhập vào bởi người dùng.
  2. Vòng lặp kiểm tra:
    - In ra màn hình một thông báo yêu cầu người dùng nhập số lượng người chơi.
    - Sử dụng một vòng lặp `while(true)` để tạo một vòng lặp vô hạn.
    - Yêu cầu người dùng nhập một ký tự.
    - Kiểm tra xem ký tự nhập vào có phải là 'Y', 'y', 'N' hay 'n'.
    - Nếu là 'Y' hoặc 'y', trả về `true`.
    - Nếu là 'N' hoặc 'n', trả về `false`.
    - Nếu không phải là một trong các ký tự trên, in ra thông báo lỗi và yêu cầu nhập lại.
- **`bool ask_to_choose_mode_again()`**: Hỏi người chơi xem họ có muốn chọn lại chế độ trò chơi không
  1. Khai báo biến:
    - Khai báo biến `user_input` kiểu `std::string` để lưu trữ câu trả lời của người dùng.
  2. Yêu cầu nhập liệu:
    - In ra màn hình một thông báo yêu cầu người dùng nhập "yes" hoặc "no".
  3. Kiểm tra đầu vào:
    - In ra màn hình một thông báo yêu cầu người dùng nhập "yes" hoặc "no".
    - Sử dụng một vòng lặp `while(true)` để tạo một vòng lặp vô hạn.
    - Yêu cầu người dùng nhập một ký tự.
    - Kiểm tra xem ký tự nhập vào có phải là 'Y', 'y', 'N' hay 'n'.
    - Nếu là 'Y' hoặc 'y', trả về `true`.
    - Nếu là 'N' hoặc 'n', trả về `false`.
    - Nếu không phải là một trong các ký tự trên, in ra thông báo lỗi và yêu cầu nhập lại.
- **`bool ask_to_continue()`**: Chức năng hỏi người chơi có muốn tiếp tục chơi không
  1. Khai báo biến:

- Khai báo biến `choice` kiểu `char` để lưu trữ ký tự nhập vào bởi người dùng.
2. Yêu cầu nhập liệu:
    - In ra màn hình một thông báo yêu cầu người dùng nhập "Y" hoặc "N".
  3. Kiểm tra đầu vào:
    - In ra màn hình một thông báo yêu cầu người dùng nhập "yes" hoặc "no".
    - Sử dụng một vòng lặp `while(true)` để tạo một vòng lặp vô hạn.
    - Yêu cầu người dùng nhập một ký tự.
    - Kiểm tra xem ký tự nhập vào có phải là 'Y', 'y', 'N' hay 'n'.
    - Nếu là 'Y' hoặc 'y', trả về `true`.
    - Nếu là 'N' hoặc 'n', trả về `false`.
    - Nếu không phải là một trong các ký tự trên, in ra thông báo lỗi và yêu cầu nhập lại.
- **`int selectGameLevel()`**: Chức năng chọn mức độ khó của trò chơi.
1. Khai báo biến:
    - Khai báo biến `level` để lưu trữ mức độ khó được chọn.
    - Khai báo biến `isValid` để kiểm tra tính hợp lệ của đầu vào.
  2. Vòng lặp kiểm tra:
    - Sử dụng vòng lặp `while` để lặp lại cho đến khi người dùng nhập một đầu vào hợp lệ.
    - Hiển thị menu với các mức độ khó.
    - Dùng `std::cin » level` để đọc đầu vào từ người dùng.
  3. Kiểm tra đầu vào:
    - Kiểm tra kiểu dữ liệu: Nếu người dùng nhập một giá trị không phải số, `std::cin » level` sẽ trả về `false`.
    - Kiểm tra giá trị: Nếu đầu vào là số, kiểm tra xem nó nằm trong khoảng từ 1 đến 3.
    - Nếu hợp lệ, gán `isValid` là `true` để thoát vòng lặp và in thông báo xác nhận.
    - Nếu không hợp lệ, in ra thông báo lỗi và yêu cầu nhập lại.
  4. Xử lý lỗi đầu vào:
    - Nếu người dùng nhập một giá trị không hợp lệ, sử dụng `std::cin.clear()` để xóa cờ lỗi và `std::cin.ignore()` để bỏ qua các ký tự không hợp lệ còn lại trong bộ đệm đầu vào.
  5. In hình ASCII:

- Sau khi nhận được đầu vào hợp lệ, hàm `printASCIIArt()` được gọi để in ra một hình ASCII.
6. Trả về mức độ khó:
- Trả về giá trị của level đã được chọn.

Các hàm xuất thông tin:

- **`void print_player_cards(std::vector<std::string> player_cards);`**: Chức năng in thẻ của người chơi
  1. Duyệt qua các lá bài:
    - Sử dụng một vòng lặp for-each để duyệt qua từng lá bài trong vector `player_cards`.
  2. In ra màn hình:
    - Đối với mỗi lá bài, thay đổi màu chữ thành xanh dương (mã màu 4) và in ra lá bài đó trên một dòng mới.
    - Sau khi in xong tất cả các lá bài, in ra một dòng trống để phân cách với các thông tin khác.
- **`void print_all_player_cards(const std::vector<std::vector<std::string>& player_cards, int number_player)`**: Chức năng in tất cả các thẻ của người chơi
  1. Duyệt qua các người chơi:
    - Sử dụng một vòng lặp for để duyệt qua từng người chơi.
  2. In thông tin người chơi:
    - In ra màn hình thông tin về người chơi hiện tại, bao gồm số thứ tự của người chơi.
  3. In các lá bài của người chơi:
    - Sử dụng một vòng lặp for-each để duyệt qua từng lá bài của người chơi hiện tại.
    - Đối với mỗi lá bài, thay đổi màu chữ thành xanh dương và in ra lá bài đó trên một dòng mới.
- **`void print_player_info(const Player& player);`**Chức năng in thông tin người chơi
  1. In thông tin cơ bản:
    - In ra số thứ tự, tên người dùng, tổng số trận đấu đã chơi và tổng số trận thắng của người chơi.
  2. In tỉ lệ thắng:
    - Tính toán và in ra tỉ lệ thắng của người chơi, làm tròn đến 2 chữ số thập phân.

3. In các tình huống thắng:
    - Nếu người chơi chưa có bất kỳ tình huống thắng nào, in ra thông báo tương ứng.
    - Nếu có, duyệt qua danh sách các tình huống thắng và in ra từng tình huống cùng với số lần xảy ra.
- **void print\_pve\_info(const Player& player, bool check\_bot, int level\_bot);:**  
 Chức năng in thông tin chế độ PvE
    1. Kiểm tra chế độ chơi:
      - Nếu check\_bot là true, tức là đang chơi với bot, thì in thông tin của bot.
      - Nếu check\_bot là false, tức là chơi với người thật, thì in thông tin của người chơi.
    2. In thông tin cơ bản:
      - In ra thông tin cơ bản của người chơi hoặc bot, bao gồm tên (hoặc mức độ khó của bot), tổng số trận đấu đã chơi và tổng số trận thắng.
    3. In tỉ lệ thắng:
      - Tính toán và in ra tỉ lệ thắng, làm tròn đến 2 chữ số thập phân.
    4. In các tình huống thắng:
      - Nếu người chơi hoặc bot chưa có bất kỳ tình huống thắng nào, in ra thông báo tương ứng.
      - Nếu có, duyệt qua danh sách các tình huống thắng và in ra từng tình huống cùng với số lần xảy ra.
  - **void leaderboard\_pve(const std::vector<Player>& players, int level\_bot);:**  
 Chức năng in bảng xếp hạng (PvE) xếp hạng người chơi dựa trên tỉ lệ thắng
    1. Sao chép danh sách người chơi:
      - Tạo một bản sao của danh sách người chơi để tránh thay đổi danh sách gốc.
    2. Sắp xếp danh sách:
      - Sắp xếp danh sách người chơi theo tỉ lệ thắng giảm dần.
    3. In tiêu đề bảng xếp hạng:
      - In tiêu đề bảng xếp hạng bao gồm các cột: Rank, Participant, Games Played, Wins, Win Rate.
    4. In thông tin từng người chơi:
      - Duyệt qua danh sách người chơi đã được sắp xếp.
      - In thông tin của từng người chơi, bao gồm thứ hạng, tên (hoặc "Bot"), số trận đấu đã chơi, số trận thắng, và tỉ lệ thắng.



- **void final \_leaderboard\_pve(const std::vector<Player>& players, int level\_bot);**: Chức năng in bảng xếp hạng cuối cùng (PvE) xếp hạng người chơi dựa trên tỷ lệ thắng tương tự với hàm `leaderboard_pve()`:
  1. Sao chép danh sách người chơi:
    - Tạo một bản sao của danh sách người chơi để tránh thay đổi danh sách gốc.
  2. Sắp xếp danh sách:
    - Sắp xếp danh sách người chơi theo tỉ lệ thắng giảm dần.
  3. In tiêu đề bảng xếp hạng:
    - In tiêu đề bảng xếp hạng bao gồm các cột: Rank, Participant, Games Played, Wins, Win Rate.
  4. In thông tin từng người chơi:
    - Duyệt qua danh sách người chơi đã được sắp xếp.
    - In thông tin của từng người chơi, bao gồm thứ hạng, tên (hoặc "Bot"), số trận đấu đã chơi, số trận thắng, và tỉ lệ thắng.
- **void leaderboard(const std::vector<Player>& players);**: Chức năng in bảng xếp hạng người chơi
  1. Sao chép danh sách người chơi:
    - Tạo một bản sao của danh sách người chơi để tránh thay đổi danh sách gốc.
  2. Sắp xếp danh sách:
    - Sắp xếp danh sách người chơi theo tỉ lệ thắng giảm dần.
  3. In tiêu đề bảng xếp hạng:
    - In tiêu đề bảng xếp hạng bao gồm các cột: Rank, Participant, Games Played, Wins, Win Rate.
  4. In thông tin từng người chơi:
    - Duyệt qua danh sách người chơi đã được sắp xếp.
    - In thông tin của từng người chơi, bao gồm thứ hạng, số thứ tự người chơi, tên người chơi, số trận đấu đã chơi, số trận thắng, và tỉ lệ thắng.
- **void final \_leaderboard(const std::vector<Player>& players);**: Chức năng in bảng xếp hạng cuối cùng của người chơi
  1. Sao chép danh sách người chơi:
    - Tạo một bản sao của danh sách người chơi để tránh thay đổi danh sách gốc.
  2. Sắp xếp danh sách:
    - Sắp xếp danh sách người chơi theo tỉ lệ thắng giảm dần.

3. In tiêu đề bảng xếp hạng:
    - In tiêu đề bảng xếp hạng bao gồm các cột: Rank, Participant, Games Played, Wins, Win Rate.
  4. In thông tin từng người chơi:
    - Duyệt qua danh sách người chơi đã được sắp xếp.
    - In thông tin của từng người chơi, bao gồm thứ hạng, số thứ tự người chơi, tên người chơi, số trận đấu đã chơi, số trận thắng, và tỉ lệ thắng.
- **void printPvPGuide():** Chức năng in hướng dẫn chế độ PvP
  - **void printPvEGuide():** Chức năng in hướng dẫn chế độ PvE
  - **void printASCIIPoker():** In ASCII Art Poker
  - **void printASCIIArt():** In ảnh ASCII
  - **void printSingleCardDuelGuide():** In hướng dẫn chế độ Đấu bài đơn
  - **void printThreeCardPokerGuide():** In hướng dẫn chế độ chơi Bài cào

#### d. Thư viện "gameMode.h":

- `void select_game_mode(std::vector<Player>& players)`: Hàm chọn chế độ chơi.

```
// Hàm chọn chế độ chơi
void select_game_mode(std::vector<Player>& players);
```

##### 1. Khai báo biến:

- `mode_choice`: Lưu trữ lựa chọn chế độ chơi của người chơi (kiểu `int`).
- `valid_choice`: Kiểm tra xem lựa chọn của người chơi có hợp lệ không (kiểu `bool`).

##### 2. Hiển thị menu:

- In ra danh sách các chế độ chơi có sẵn.

##### 3. Nhập lựa chọn:

- Yêu cầu người chơi nhập lựa chọn chế độ chơi.

##### 4. Kiểm tra tính hợp lệ của đầu vào:

- Sử dụng vòng lặp `do-while` để lặp lại cho đến khi người chơi nhập lựa chọn hợp lệ.
- Kiểm tra xem đầu vào có phải là số nguyên nằm trong phạm vi 1-4 hay không.
- Xóa trạng thái lỗi của `cin` và bỏ qua đầu vào không hợp lệ nếu người dùng nhập sai.

##### 5. Xử lý lựa chọn hợp lệ:

- Sử dụng câu lệnh `switch` để thực thi hàm tương ứng với lựa chọn của người chơi:
- case 1: Chạy chế độ chơi "Player vs Player" (chưa được cung cấp hàm `run_poker_pvp_mode`).
- case 2: Chạy chế độ chơi "Player vs Bot" (chưa được cung cấp hàm `run_poker_pve_mode`).
- case 3: Chạy chế độ chơi "Single Card Duel" (chưa được cung cấp hàm `run_Single_Card_Duel`).
- case 4: Chạy chế độ chơi "Three Card Poker" (chưa được cung cấp hàm `run_Three_Card_Poker`).

### \*Chế độ PvP\_Poker:

- **void run\_poker\_pvp\_mode():** Khởi tạo và thực thi chế độ chơi PVP.
  1. Khởi tạo:
    - Khởi tạo số lượng người chơi.
    - Khởi tạo thông tin cho mỗi người chơi.
  2. Vòng lặp chơi:
    - Tạo bộ bài, xáo bài và chia bài cho các người chơi.
    - Xác định loại bài và tính điểm cho mỗi người chơi.
    - Xác định người chiến thắng.
    - Cập nhật thông tin người chơi (số trận đấu, số trận thắng, tỉ lệ thắng, tình huống thắng).
    - In thông tin người chơi và bảng xếp hạng.
    - Hỏi người chơi có muốn tiếp tục hay không.
  3. Kết thúc trò chơi:
    - In bảng xếp hạng cuối cùng.
    - Lưu dữ liệu trò chơi vào file.
- **void calculate\_player\_points(...):** Tính điểm cho từng người chơi. Hàm đầy đủ ở dưới đây:

```
void calculate_player_points(int number_player, const std::vector<std::vector<std::string>>& player_cards, int ranks[], int suits[], std::vector<int> &player_value_card_points, std::vector<int> &player_rank_points, bool check_print, bool pve_mode);
```

1. Xác định tổ hợp bài:
  - Hàm kiểm tra từng tay bài để xác định tổ hợp bài mạnh nhất. Các tổ hợp bài được kiểm tra bao gồm:
    - \* Thùng phá sảnh (Straight Flush)
    - \* Tứ quý (Four of a Kind)
    - \* Cù lũ (Full House)
    - \* Thùng (Flush)
    - \* Sảnh (Straight)
    - \* Ba lá giống nhau (Three of a Kind)
    - \* Hai đôi (Two Pair)
    - \* Một đôi (One Pair)
    - \* Mậu thầu (High Card)
2. Gán điểm:

- Mỗi tổ hợp bài được gán một điểm số tương ứng. Tổ hợp bài càng mạnh, điểm số càng cao.
  - Nếu hai người chơi có cùng một tổ hợp bài, người chơi có lá bài cao nhất trong tổ hợp đó sẽ thắng.
3. Lưu trữ điểm số:
- `player_value_card_points`: Lưu trữ tổng điểm của tay bài.
  - `player_rank_points`: Lưu trữ điểm của lá bài cao nhất trong tay bài.
4. In thông tin (tùy chọn):
- Nếu `check_print` được đặt là `true`, hàm sẽ in ra thông tin về tay bài của mỗi người chơi, bao gồm loại tổ hợp bài và lá bài cao nhất.
- **`void determine_winner(...)`**: Xác định người chiến thắng. Khai báo hàm đầy đủ dưới đây.

```
// Xác định người chiến thắng
void determine_winner(int number_player,
std::vector<int>& player_value_card_points,
std::vector<int>& player_rank_points,
std::vector<Player>& players);
```

1. Tìm người chơi có điểm số cao nhất:
  - Duyệt qua danh sách người chơi để tìm người có điểm số cao nhất.
  - Nếu có nhiều người chơi có cùng điểm số cao nhất, thêm tất cả vào danh sách người thắng.
2. Xác định người thắng:
  - Nếu chỉ có một người thắng, in ra thông tin người thắng và cập nhật thông tin của người đó
  - Nếu có nhiều người thắng (hòa), in ra thông tin của tất cả người thắng và cập nhật thông tin của từng người.
  - Xác định người chiến thắng.
3. Cập nhật thông tin người chơi:
  - Cập nhật số trận đấu, số trận thắng, tỉ lệ thắng và tình huống thắng cho cả người thắng và người thua.

#### \*Chế độ PvE\_Poker:

- **`int random_number()`**: Tạo ngẫu nhiên các số (1 đến 10). Nhằm tạo random của hàm xác định người chiến thắng.

- **void run\_poker\_pve\_mode()**: Khởi tạo và thực thi chế độ chơi PVE

1. Khởi tạo:

- Khởi tạo hai người chơi: người chơi và bot.
- Chọn mức độ khó của bot.

2. Vòng lặp chơi:

- Tạo bộ bài 52 lá và xáo bài.
- Chia bài cho người chơi và bot.
- Xác định người chiến thắng.
- Xác định loại bài và tính điểm cho cả người chơi và bot.
- Logic đặc biệt cho chế độ PvE: Sử dụng số ngẫu nhiên để quyết định xem bot có nên giữ lá bài mạnh nhất hay không, dựa trên mức độ khó.
- Hiển thị các lá bài của người chơi và bot.
- Xác định người thắng và cập nhật thông tin người chơi.
- Hiển thị thông tin người chơi và bảng xếp hạng.
- Hỏi người chơi có muốn tiếp tục hay không.

3. Kết thúc trò chơi:

- In bảng xếp hạng cuối cùng.
- Lưu dữ liệu trò chơi vào file.

- **void calculate\_player\_points(...)**: Tính điểm cho từng người chơi tương tự như PVP. Hàm đầy đủ ở dưới đây:

```
void calculate_player_points(int number_player,
const std::vector<std::vector<std::string>>& player_cards,
int ranks[], int suits[], std::vector<int> &player_value_card_points,
std::vector<int> &player_rank_points, bool check_print, bool pve_mode);
```

- **void determine\_winner\_PVE(...)**: Xác định người chiến thắng tương tự như PVP chỉ khác là ở người là bot. Khai báo hàm đầy đủ dưới đây:

```
void determine_winner_PvE(int number_player,
...
std::vector<int>& player_value_card_points,
std::vector<int>& player_rank_points,
std::vector<Player>& players);
```

### \*Chế độ Single Dual Card:

- **void run\_Single\_Card\_Duel():** Hàm khởi tạo và thực thi chế độ chơi Single Card Duel
  1. Khởi tạo:
    - Khởi tạo số lượng người chơi.
    - Khởi tạo thông tin cho mỗi người chơi.
  2. Vòng lặp chơi:
    - Tạo bộ bài, xáo bài và chia bài cho các người chơi.
    - Xác định giá trị và chất của lá bài mỗi người chơi.
    - Xác định người chiến thắng.
    - Cập nhật thông tin người chơi (số trận đấu, số trận thắng, tỉ lệ thắng, tình huống thắng).
    - In thông tin người chơi và bảng xếp hạng.
    - Hỏi người chơi có muốn tiếp tục hay không.
  3. Kết thúc trò chơi:
    - In bảng xếp hạng cuối cùng.
    - Lưu dữ liệu trò chơi vào file.
- **bool is\_stronger\_between\_2\_cards(const Card& a, const Card& b):** Hàm kiểm tra giữa 2 lá bài lá nào mạnh hơn
- **void calculate\_player\_points\_Single\_Card\_Duel(...):** Tính điểm cho từng người chơi trong chế độ Single Card Duel

```
void calculate_player_points(int number_player, const std::vector<std::vector<std::string>>
& player_cards, int ranks[], int suits[], std::vector<int> &player_value_card_points,
std::vector<int> &player_rank_points, bool check_print, bool pve_mode);
```

- **void determine\_winner\_Single\_Card\_Duel(...):** Xác định người chiến thắng. Khai báo hàm đầy đủ dưới đây.

```
// Hàm xác định người chiến thắng trong chế độ Three Card Poker
std::vector<int> determine_winners_Three_Card_Poker(
    std::vector<Player>& players,
    std::vector<std::vector<std::string>>& player_cards);
```

### \*Chế độ Three Card Poker (Bài cào):

- **void run\_Three\_Card\_Poker();**: Hàm khởi tạo và thực thi chế độ chơi Single Card Duel
- **bool is\_stronger\_between\_2\_cards(const Card& a, const Card& b);**: Hàm kiểm tra giữa 2 lá bài lá nào mạnh hơn
- **void calculate\_player\_points\_Three\_Card\_Poker(...)**: Tính điểm cho từng người chơi trong chế độ Single Card Duel

```
// Hàm tính điểm cho chế độ Three Card Poker
int calculate_player_score_Three_Card_Poker(std::vector<std::string>& player_cards);
```

- **void determine\_winner\_Three\_Card\_Poker(...)**: Xác định người chiến thắng. Khai báo hàm đầy đủ dưới đây.

```
// Hàm xác định người chiến thắng trong chế độ Three Card Poker
std::vector<int> determine_winners_Three_Card_Poker(
    std::vector<Player>& players,
    std::vector<std::vector<std::string>>& player_cards);
```

### e. Thư viện "file\_manipulation.h":

Thư viện này bao gồm các hàm liên quan đến việc nhập xuất file và gồm các hàm:

- **void clear\_file();**: Hàm xóa nội dung file
- **void save\_game\_data\_to\_file(...)**: Hàm lưu dữ liệu trò chơi vào file
- **void save\_game\_data\_to\_file\_poker\_PvE\_mode(...)**: Hàm lưu dữ liệu trò chơi vào file cho chế độ Poker PvE

Hàm được khai báo đầy đủ như sau:

```
// Hàm lưu dữ liệu trò chơi vào file
void save_game_data_to_file(const std::string& mode, const std::vector<Player>& players);

// Hàm lưu dữ liệu trò chơi vào file cho chế độ Poker PvE
void save_game_data_to_file_poker_PvE_mode(const std::string& mode, const std::vector<Player>& players);
```





## Bước 2: Chọn chế độ game

Có 4 chế độ theo như trên hình, chọn từ 1 đến 4 vào chế độ game mong muốn

```
Do you want to read the guide to playing the PvP mode game?
Please enter Y (Yes) or N (No): y
--- PvP (Player vs. Player) Poker Mode ---

Description:
+ The PvP mode allows two players to compete directly against each other. Each player will take turns during their play.

Instructions:
+ When entering PvP mode, input the number of players participating (from 2 to 10).
+ Enter the usernames of the players.
+ The computer will randomly deal cards to each player.
+ The computer will automatically evaluate and determine the strongest hand for each player and display it on the screen.
+ The computer will automatically determine the winner.
+ Players who have or share the strongest hand will win and receive 1 point.
+ The computer will automatically update player information.
+ The computer will display the current round leaderboard, sorted by win rate.
+ The game does not limit the number of rounds.
+ The game ends when the participants decide to stop playing.
+ The computer will display the final leaderboard on the screen.
+ The computer will save the match data, including player information and the final Leaderboard, into the file Data_Game.txt.

Rules for Determining Hands:
1. The program will randomly deal 5 cards to each participant.

2. The card rankings are as follows:
   2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < J < Q < K < A.

3. The program evaluates the participants' hands in the following order, from highest to lowest:

   | Rank | Hand Type       | Description                                     |
   |-----|-----|-----|
   | 1 | Straight Flush | 5 consecutive cards of the same suit.          |
   | 2 | Four of a Kind | 4 cards of the same rank.                      |
   | 3 | Full House     | 3 cards of the same rank and a pair.           |
   | 4 | Flush          | 5 cards of the same suit but not consecutive.  |
   | 5 | Straight       | 5 consecutive cards, not necessarily suited.    |
   | 6 | Three of a Kind | 3 cards of the same rank.                      |
   | 7 | Two Pairs      | Two pairs of cards with different ranks.        |
   | 8 | One Pair       | A single pair of cards with the same rank.      |
   | 9 | High Card      | The highest-value card in the hand.            |

4. Suits are not considered (only the rank of the cards matters).

5. The player with the highest-ranked hand wins the match.
```

3. Khi vào chế độ PvP, chương trình sẽ yêu cầu nhập số lượng người chơi sẽ tham gia (từ 2 đến 10)

```
Enter number of players(between 2 and 10): █
```

Giả sử nhập 3 người

-> Màn hình sẽ hiện lên hình ảnh người chia bài

[illegible]

Giả sử lần lượt nhập An, Binh, Vui

5 1 5 1 4

5. Chương trình sẽ lần lượt chia bài ngẫu nhiên cho mỗi người và hiện ra màn hình

```
Cards of player 1:
2 spade
6 club
8 club
10 spade
J diamond

Cards of player 2:
6 heart
8 heart
10 diamond
Q heart
K spade

Cards of player 3:
2 diamond
4 heart
5 club
9 spade
Q diamond
```

6. Máy máy tính sẽ tự động tính toán và tìm ra thể bài mạnh nhất của mỗi người và in ra màn hình

- + Máy tính sẽ tự động xác định người chiến thắng
- + Những người chơi có hoặc cùng có thể bài mạnh nhất sẽ giành sẽ chiến thắng về nhận 1 điểm
- + Máy tính sẽ tự động cập nhật thông tin của người chơi
- + Máy tính sẽ in ra màn hình Bảng Leaderboard vòng hiện tại sắp xếp theo tỉ lệ thắng
- + Trò chơi không giới hạn số lượng vòng chơi
- + Trò chơi kết thúc khi người tham gia quyết định dừng chơi
- + Máy tính sẽ in ra màn hình Bảng Leaderboard cuối cùng

+ Máy tính sẽ in ra màn hình Bảng Leaderboard cuối cùng

+ Máy tính sẽ lưu dữ liệu trận đấu gồm thông tin người chơi và Bảng Leaderboard cuối cùng vào file Data\_Game.txt

#### Quy tắc xác định thế bài

1, Chương trình sẽ chia ngẫu nhiên 5 lá bài cho mỗi người tham gia.

2, Thứ tự xếp hạng của các lá bài như sau:

$2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < J < Q < K < A$ .

3, Chương trình sẽ đánh giá bài của người tham gia theo thứ tự sau, từ cao nhất đến thấp nhất:

	Thứ tự	Loại bài	Mô tả
	1	Straight Flush	5 lá bài liên tiếp cùng chất.
	2	Four of a Kind	4 lá bài cùng bậc.
	3	Full House	3 lá bài cùng bậc và 1 đôi khác bậc.
	4	Flush	5 lá bài cùng chất nhưng không liên tiếp.
	5	Straight	5 lá bài liên tiếp, không nhất thiết cùng chất.
	6	Three of a Kind	3 lá bài cùng bậc.

## Single Card Duel:

Mô tả

+ Chế độ Single Card Duel cho phép nhiều người chơi cạnh tranh trực tiếp

+ Mỗi người sẽ được nhận 1 lá bài

+ Một chế độ chơi đơn giản và nhanh gọn

Mô tả:

+ Chế độ Three\_Card\_Poker cho phép nhiều người chơi cạnh tranh trực tiếp

+ Mỗi người sẽ được nhận 3 lá bài

+ Những ai có thể bài mạnh nhất sẽ giành chiến thắng

Hướng dẫn chơi

+ Nhập số lượng người chơi:

+ Khi bắt đầu chế độ, người chơi nhập số lượng người tham gia (từ 2 đến 10).

+ Chia bài ngẫu nhiên: Máy tính sẽ tự động chia ngẫu nhiên lá bài cho từng người chơi.

+ Sau khi bài được chia, máy tính sẽ xác định điểm tối đa của mỗi người và hiển thị kết quả.

+ Người chơi có điểm cao nhất sẽ giành chiến thắng vòng đầu và nhận 1 điểm.

- + Nếu có nhiều người cùng bằng điểm cao nhất, tất cả họ sẽ nhận 1 điểm.

- + Máy tính tự động cập nhật điểm số của người chơi sau mỗi vòng.

- + Hiện thị Bảng Leaderboard tạm thời, sắp xếp theo tỷ lệ thắng, để mọi người theo dõi tiến độ.

- + Trò chơi không giới hạn vòng

- + Khi kết thúc trò chơi, máy tính sẽ in ra Bảng Leaderboard cuối cùng, hiển thị kết quả toàn bộ trận đấu

Quy tắc xác định thể bài

- +Mô tả:

- + Chế độ Three\_Card\_Poker cho phép nhiều người chơi cạnh tranh trực tiếp

- + Mỗi người sẽ được nhận 3 lá bài

- + Những ai có thể bài mạnh nhất sẽ giành chiến thắng

Hướng dẫn chơi

- + Nhập số lượng người chơi:

- + Khi bắt đầu chế độ, người chơi nhập số lượng người tham gia (từ 2 đến 10).

- + Chia bài ngẫu nhiên: Máy tính sẽ tự động chia ngẫu nhiên lá bài cho từng người chơi.

+ Sau khi bài được chia, máy tính sẽ xác định điểm tối đa của mỗi người và hiển thị kết quả.

+ Người chơi có điểm cao nhất sẽ giành chiến thắng vòng đầu và nhận 1 điểm.

+ Nếu có nhiều người cùng bằng điểm cao nhất, tất cả họ sẽ nhận 1 điểm.

+ Máy tính tự động cập nhật điểm số của người chơi sau mỗi vòng.

+ Hiển thị Bảng Leaderboard tạm thời, sắp xếp theo tỷ lệ thắng, để mọi người theo dõi tiến độ.

+ Trò chơi không giới hạn vòng

+ Khi kết thúc trò chơi, máy tính sẽ in ra Bảng Leaderboard cuối cùng, hiển thị kết quả toàn bộ trận đấu

Quy tắc xác định lá bài

1. Hạng (rank) được sắp xếp từ 2 (thấp nhất) đến A (cao nhất):

$2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < J < Q < K < A$ .

2. Chất (suit) được sắp xếp như sau:

Cơ > rô > Chuồn/Tép > Bích

3. Cách xác định người chiến thắng như sau:

4. Người chơi có lá bài có bộ số cao nhất sẽ chiến thắng.



5. Nếu hai hoặc nhiều người chơi có cùng hạng lá bài, sẽ xem xét đến chất để phân thắng thua:

Chất có giá trị lớn hơn sẽ chiến thắng.

Three Card Poker:

Mô tả:

+ Chế độ Three\_Card\_Poker cho phép nhiều người chơi cạnh tranh trực tiếp

+ Mỗi người sẽ được nhận 3 lá bài

+ Những ai có thể bài mạnh nhất sẽ giành chiến thắng

Hướng dẫn chơi

+ Nhập số lượng người chơi:

+ Khi bắt đầu chế độ, người chơi nhập số lượng người tham gia (từ 2 đến 10).

+ Chia bài ngẫu nhiên: Máy tính sẽ tự động chia ngẫu nhiên lá bài cho từng người chơi.

+ Sau khi bài được chia, máy tính sẽ xác định điểm tối đa của mỗi người và hiển thị kết quả.

+ Người chơi có điểm cao nhất sẽ giành chiến thắng vòng đầu và nhận 1 điểm.

+ Nếu có nhiều người cùng bằng điểm cao nhất, tất cả họ sẽ nhận 1 điểm.

+ Máy tính tự động cập nhật điểm số của người chơi sau mỗi vòng.

+ Hiển thị Bảng Leaderboard tạm thời, sắp xếp theo tỷ lệ thắng, để mọi người theo dõi tiến độ.

+ Trò chơi không giới hạn vòng

+ Khi kết thúc trò chơi, máy tính sẽ in ra Bảng Leaderboard cuối cùng, hiển thị kết quả toàn bộ trận đấu

Quy tắc xác định thể bài

+ Bài tây chuẩn (52 lá)

+ Giá trị lá bài:

A (Át) = 1 điểm.

Các lá từ 2 đến 9 giữ nguyên điểm tương ứng với số trên lá bài.

10, J, Q, K bằng 10 điểm

+ Cộng điểm của 3 lá bài.

+ Chỉ lấy chữ số hàng đơn vị.

Ví dụ: Tổng điểm là 18, điểm thực tế là 8.

+ Điểm tối đa: 9 điểm

## II. THAM KHẢO

### 1. Các nguồn trên mạng:

#### a. Code:

- Tạo random number: [Tạo số ngẫu nhiên](#)
- Thuật toán xáo bài: [C++ Algorithm\\_random\\_shuffle](#)
- Đổi màu chữ trong terminal: [Colored output in C](#)

#### b. Latex:

- Latex tutorial in Vietnamese: [Latex tutorial in Vietnamese](#)
- Cách làm một báo cáo Latex (Đại học Bách Khoa Hà Nội): [Link danh sách phát](#)
- Tạo bảng tự động: [Excel to Latex](#)
- Gắn link: [Latex hyperref URL hyperlinks](#)
- Tài liệu Latex môn Nhập môn Công nghệ thông tin.

#### c. Luật chơi Poker:

- Upswing Poker: [:link Upswing Poker](#)

### 2. Các công cụ AI hỗ trợ:

Một số đoạn code tại em không sử dụng nguồn code có sẵn mà nhờ sự hỗ trợ và các đoạn code được tạo hoặc sửa chữa bởi các loại AI , một số đoạn code dùng AI để chỉnh sửa. Dưới đây là các loại AI tại em đã dùng:

- Chat GPT
- Gemini
- Copilot

### 3. Con người:

- **Mai Vinh Hiền:** Cựu chủ nhiệm GDGoC club
- **Lê Quốc Vĩ:** học Khoa học máy tính HCMUT
- **Phan Ngọc Mỹ Duyên:** học Khoa học máy tính HCMUT