Jogo de Cartas

Laboratório de Algoritmia 1

Laboratórios de Informática 2

Ano Letivo 2023/24

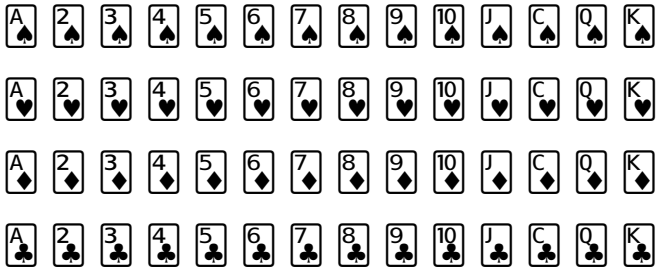
Sumário

1 Introdução

O objetivo deste projeto é implementar um jogo de cartas relativamente simples em C no sistema operativo Linux.

1.1 Baralho

O baralho contém as seguintes 56 cartas:



A ordem dos valores e dos naipes é dada acima. Assim:

- A ordem dos naipes é: ♠ ♥ ♦ ♣
- A ordem **ascendente** dos valores é: A 2 3 4 5 6 7 8 9 10 J C Q K

1.2 Objetivo

O objetivo do jogo é livrarmo-nos de todas as cartas. Cada jogador recebe 14 cartas. Na sua vez, o jogador pode jogar uma combinação de cartas permitida ou passar a vez. Não é obrigatório jogar mesmo que o possa fazer.

Se for o primeiro a jogar, o jogador pode escolher qualquer combinação válida. Se jogarmos após outro jogador, temos que jogar uma combinação igual e com o mesmo número de cartas. A resposta tem de ser uma combinação cuja carta mais alta seja superior à carta mais alta da combinação jogada ou passar. Há algumas excepções a esta regra quando a última combinação jogada só contém reis. Caso todos os outros jogadores passarem, a ronda acabou e somos nós a começar a jogar novamente e por isso podemos jogar qualquer combinação.

Quando somos os primeiros a livramo-nos de todas as cartas, ganhamos pontos por cada carta que ficou nas mãos dos nossos adversários.

1.3 Combinações válidas

Existem várias combinações possíveis:

- **Conjunto** de uma até quatro cartas todas com o mesmo valor mas de naipes diferentes;
- **Sequência** de três ou mais cartas de valores consecutivos (não necessariamente todas do mesmo naipe);
- **Dupla sequência** de três ou mais pares de valores consecutivos.

Cabe ao jogador escolher a combinação que quer jogar. Uma combinação só pode ser seguida de uma combinação do mesmo tipo e com o mesmo número de cartas. Para comparar duas combinações, comparamos a carta mais alta de cada combinação. Uma carta é mais alta se tiver o valor mais alto ou, sendo os valores iguais, se tiver um naipe mais valioso.

Existem algumas excepções à regra de que uma combinação tenha que ser seguida de outra com o mesmo tipo e número de cartas:

- Um rei (K) pode ser seguido de um conjunto de quatro cartas iguais;
- Um rei (K) pode ser seguido de uma sequência de três pares;
- Um par de reis pode ser seguido de uma sequência de quatro pares;
- Um trio de reis pode ser seguido por uma sequência de cinco pares.

A seguir a uma destas combinações tem que se seguir outra do mesmo tipo mas mais valiosa.

1.4 Exemplo de um jogo

Jogador 1	Jogador 2	Jogador 3	Jogador 4
6 ♠	6 ♣	8 ♦	10 ♦
C ♥	K ♥	3 ♠ 3 ♥ 3 ♦ 3 ♣	7 ♠ 7 ♥ 7 ♦ 7 ♣
PASSO	PASSO	PASSO	4 ♥ 4 ♦ 5 ♠ 5 ♥ 6 ♥ 6 ♣
PASSO	PASSO	PASSO	10 ♠ J ♥ C ♦

Tabela 1: Exemplo de um jogo

O primeiro jogador jogou o 6 ♠. Seguiram-se várias cartas isoladas até que o segundo jogador jogou o K ♥. O terceiro jogador jogou a combinação de excepção do conjunto de quatro cartas iguais. A jogador a seguir jogou quatro setes. Todos os outros jogadores escolheram passar (ou porque não podiam jogar ou porque não o quiseram fazer).

O quarto jogador decidiu jogar uma dupla sequência de três cartas e novamente todos os outros passaram. Finalmente, ele jogou as suas últimas três cartas numa sequência de três cartas.

2 Guiões

1. Dado um conjunto de cartas, imprimir qual é o tipo de combinação que este conjunto de cartas representa e qual é a carta mais alta dessa combinação;

2. Dadas várias combinações, dizer se são compatíveis e ordená-las;
3. Dada a informação apresentada ao jogador para o torneio e a jogada do jogador, imprimir a nova mão desse jogador. Deverá imprimir-se a mesma mão caso a jogada seja inválida;
4. Dada a última combinação jogada e as cartas da mão, apresentar todas as combinações possíveis que se pode jogar nessa posição ou escrever PASSO.

3 Torneio

3.1 Elegibilidade

São elegíveis para o torneio os grupos que:

- Tenham concluído com sucesso todos os guiões;
- Entreguem o jogador no concurso correto do servidor;
- Tenham **Accepted** a todos os testes,

3.2 Rondas

- Inicialmente, os jogadores são sorteados em grupos de quatro jogadores;
- Na ronda inicial existem dois jogadores aleatórios em cada grupo;
- Cada grupo joga 16 vezes começando cada jogador à vez;
- Os dois jogadores com maior pontuação passam à ronda seguinte;
- Se um grupo não tiver vencedores, os jogadores serão todos eliminados;
- As rondas sucedem-se até que exista um só jogo a decorrer.

3.3 Jogo

- Um jogo decorre com quatro jogadores, recebendo cada um o mesmo número de cartas;
- Há várias rodadas até que um dos jogadores fique sem cartas;
- Cada jogador tem a hipótese de jogar uma combinação válida ou passar;
- Se um jogador sugerir uma combinação inválida, isso equivale a passar;
- O jogo acaba quando um jogador ficar sem cartas ou se todos os jogadores passarem;
- Quando um jogador ficar sem cartas, ele recebe tantos pontos como o número de cartas que ficou na mão dos outros jogadores;
- Um jogo é ignorado caso terminar sem ninguém ficar sem cartas.

3.4 Implementação

Para participar no torneio, é necessário escrever um programa que leia do *standard input*:

- Uma linha com as suas cartas;
- As combinações jogadas até agora por cada jogador, ou a palavra PASSO se esse jogador passou (uma por cada linha)

E que imprima qual é a combinação que joga ou a palavra PASSO.

3.4.1 Exemplo de input

Usando o exemplo do jogo acima, o seu programa poderia receber o seguinte input:

```
10 ♠ J ♥ C ♦
6 ♠
6 ♣
8 ♦
10 ♦
C ♥
K ♥
3 ♠ 3 ♥ 3 ♦ 3 ♣
7 ♠ 7 ♥ 7 ♦ 7 ♣
PASSO
PASSO
PASSO
4 ♥ 4 ♦ 5 ♠ 5 ♥ 6 ♥ 6 ♣
PASSO
PASSO
PASSO
```

3.4.2 Exemplo de output

Nesta situação, o seu programa poderia imprimir uma das seguintes linhas:

1. `PASSO`
2. `10 ♠`
3. `J ♥`
4. `C ♦`
5. `10 ♠ J ♥ C ♦`

Mas, como é óbvio, neste caso só a última destas combinações assegura a vitória.

4 Avaliação

Eis a informação pertinente sobre a avaliação do projeto:

Etapa	Data de Entrega	Pontuação
Guião 1	8 de Abril	2
Guião 2	15 de Abril	2
Guião 3	22 de Abril	2
Guião 4	29 de Abril	2
Entrega Final	18 de Maio	2
Entrega para Torneio	18 de Maio	2

Tabela 2: Avaliação do projeto

4.1 Grupos

- Os grupos são de quatro elementos;
- Existe **avaliação de pares** em que cada elemento do grupo avalia os restantes elementos;
- A avaliação de pares é **pública**;
- A avaliação determina a percentagem da componente de grupo para cada um dos elementos.

4.2 Avaliação dos Guiões

- Implica a submissão para o MOOshak;
- Sujeita a defesa durante as aulas.

4.3 Defesa

- A defesa será entre 27 e 29 de Maio;
- Serão feitas perguntas sobre o código.

4.4 Anulação da componente de grupo

- Caso um aluno não compareça na defesa;
- Caso um aluno não seja capaz de responder a qualquer pergunta sobre o código;
- Se nenhum elemento do grupo conseguir explicar uma parte do código.

4.5 Pontuação do torneio

A pontuação é distribuída da seguinte maneira:

- **Vencedor**: 2 pontos
- **Final**: 1.5 pontos

- **Semi-final**: 1 ponto
- **Quartos de final**: 0.75 pontos
- **Oitavos de final**: 0.5 pontos