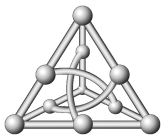

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
Faculdade de Computação

Poker

Algoritmos e Programação II - Trabalho 1

1 Descrição

O jogo de poker (ou pôquer) é um jogo de cartas jogado por duas ou mais pessoas muito comum em casinos. É considerado o jogo de cartas mais popular do mundo, e o mais popular de uma classe de jogos nos quais os jogadores com as cartas total ou parcialmente escondidas fazem apostas para um monte central, após o que o resultante das apostas é atribuído ao(s) jogador(es) que possuir(em) o melhor conjunto de cartas dentre os que permaneceram na mão, ou ao jogador restante caso os outros tenham desistido. Os valores das várias combinações de cartas, chamadas **mãos**, decidem quem é o ganhador.

Sabe-se que as capivaras da UFMS apreciam muito este jogo, motivo pelo qual a elite da sociedade capivarense criou um grupo denominado "The Capybara Club", com reuniões aos finais de semana. O grupo é destinado à prática de uma versão simplificada do poker.

Figura 1: A elite da sociedade capivarense

Nessa versão simplificada, há menos possibilidades de mãos, e mãos iguais resultam em empate. Considere que as cartas J, Q, K e A valem 11, 12, 13 e 14, respectivamente (o A não vale 1). Temos as seguintes mãos, do maior valor para o menor valor:

- Royal Flush: 5 cartas do mesmo naipe exatamente na sequência 10, J, Q, K e A
- Straight Flush: 5 cartas do mesmo naipe com valores em sequência (p. ex. 4, 5, 6, 7 e 8)
- Quadra: 4 cartas de mesmo valor
- Full House: 3 cartas de mesmo valor e outras 2 cartas de mesmo valor (p. ex. 5, 5, 5, 3 e 3)
- Flush: 5 cartas de quaisquer valores com o mesmo naipe
- Sequência: 5 cartas de quaisquer naipes com valores em sequência
- Trinca: 3 cartas de mesmo valor
- 2 pares: 2 cartas de mesmo valor e outras 2 cartas de mesmo valor
- Par: 2 cartas com o mesmo valor
- Carta mais alta: se nenhuma das mãos acima ocorre, é considerada a carta de maior valor

Considere que empates independem do valor das cartas e naipes, exceto para "Carta mais alta", que independe apenas dos naipes. Portanto, um par de 9's empata com um par de 5's, uma carta mais alta 9 ganha de uma carta mais alta 5, e há empate caso a mão de ambos jogadores seja carta mais alta com valor 7, por exemplo. Você deve escrever um programa que, dentre duas jogadoras com 5 cartas, ajude as capivaras a decidir qual possui a mão ganhadora ou se há empate.

2 Entrada e saída

Como **entrada**, seu programa deve receber uma linha com um inteiro $k > 0$ que representa a quantidade de casos de teste. Cada caso de teste consiste em duas linhas, contendo a primeira linha as 5 cartas da primeira capivara, e a segunda linha as 5 cartas da segunda. Cada carta é apresentada no formato `valor naipe`. O campo `valor` pode ser um número entre 2 e 10 ou uma letra entre J, Q, K e A. O campo `naipe` é uma letra entre `P` (Paus ♣), `O` (Ouros ♢), `C` (Copas ♡) e `E` (Espadas ♠). Considere que as cartas não necessariamente serão dadas em ordem crescente de valor, e que podem haver espaços extras entre os dados de entrada.

Para cada um dos casos de teste, seu programa deve gerar uma **saída** em uma linha escrevendo `1` caso a primeira capivara vença, `2` caso a segunda vença, e `E` caso haja empate. Adicionalmente, caso não haja empate você deve escrever após `1` ou `2` as cartas da capivara ganhadora, ordenadas de acordo com os seguintes critérios:

1º Menor valor;

2º Naipe, na ordem paus (`P` ♣) (menor), ouros (`O` ♢), copas (`C` ♡) e espadas (`E` ♠) (maior).

Portanto, um 8 de ouros deve vir antes de (é menor que) um 8 de espadas na saída da mão vencedora. Por fim, ao escrever a mão deve haver um espaço entre cada valor ou naipe, mas não após o último naipe.

3 Exemplo de entrada

```
5
6 P   7 O   3 P   9 C   J E
10 C  5 E   4 E   3 E   2 E
K C   2 P   5 O   7 P   10 C
6 P   9 O   K E   3 P   4 C
8 C   9 C   10 O  J P   Q E
5 E   4 E   3 E   7 E   6 E
5 C   6 O   2 O   3 O   4 P
K C   10 C  J C   A P   Q C
A C   7 E   5 E   7 O   7 P
4 E   A O   3 E   3 P   4 C
```

4 Exemplo de saída

```
1 3 P 6 P 7 O 9 C J E
E
2 3 E 4 E 5 E 6 E 7 E
E
1 5 E 7 P 7 O 7 E A C
```

5 Exigências

Você DEVE utilizar as seguintes estruturas de dados para armazenar, ordenar ou manipular as cartas lidas:

```
typedef enum {
   COPAS = 'C',
   ESPADAS = 'E',
   OUROS = 'O',
   PAUS = 'P'
} t_naipe;

typedef struct {
   short valor;
   t_naipe naipe;
} t_carta;
```

Além disso, você DEVE implementar algum método visto em sala para ordenar as cartas dadas como entrada, a fim de escrever na saída as cartas do ganhador ordenadas de acordo com os critérios apresentados anteriormente. Observe que ordenar as cartas pode facilitar a verificação das mãos das capivaras para decidir a mão ganhadora.

Caso as exigências acima não sejam cumpridas, a nota de seu trabalho é ZERO. Adicionalmente, considere como **sugestão** utilizar também as seguintes estruturas para armazenar e manipular mãos:

```
typedef enum {
   PAR = 15,
   DOISPARES,
   TRINCA,
   SEQUENCIA,
   FLUSH,
   FULL,
   QUADRA,
   SFLUSH,
   RFLUSH
} t_valor_m;

typedef struct {
   t_carta cartas[5];
   t_valor_m valor;
} t_mao;
```

6 Entrega

Instruções para entrega do seu trabalho:

1. **Cabeçalho**

   Seu trabalho deve ter um cabeçalho com o seguinte formato:

```
/*************************************************
 *
 * Nome do(a) estudante
 * Trabalho 1
 * Professor(a): Nome do(a) professor(a)
 *
 */
```

2. **Compilador**

   Os(as) professores(as) usam o compilador da linguagem C da coleção de compiladores GNU `gcc`, com as opções de compilação `-Wall -std=c99 -pedantic` para corrigir os programas. Se você usar algum outro compilador para desenvolver seu programa, antes de entregá-lo verifique se o seu programa tem extensão `.c`, compila sem mensagens de alerta e executa corretamente.

3. **Forma de entrega**

   A entrega será realizada diretamente na página da disciplina no AVA/UFMS. Um fórum de discussão deste trabalho já se encontra aberto. Após abrir uma sessão digitando seu *login* e sua senha, vá até o tópico "Trabalhos", e escolha "T1 - Entrega". Você pode fazer o upload de vários rascunhos, mas **o envio definitivo deve ser feito até a data indicada no AVA**. Apenas com o envio definitivo seu trabalho será corrigido. Encerrado o prazo, não serão mais aceitos trabalhos.

4. **Atrasos**

   Trabalhos atrasados não serão aceitos. Não deixe para entregar seu trabalho na última hora. Para prevenir imprevistos como queda de energia, problemas com o sistema, falha de conexão com a internet, sugerimos que a entrega do trabalho seja feita pelo menos um dia antes do prazo determinado.

5. **Erros**

   Trabalhos com erros de compilação receberão nota ZERO. Faça todos os testes necessários para garantir que seu programa está livre de erros de compilação.

6. **O que entregar?**

   Você deve entregar um único arquivo contendo APENAS o seu programa fonte com o mesmo nome de seu login no passaporte UFMS, como por exemplo, `fulano.silva.c`. NÃO entregue qualquer outro arquivo, tal como o programa executável, já compilado.

7. **Verificação dos dados de entrada**

   Não se preocupe com a verificação dos dados de entrada do seu programa. Seu programa não precisa fazer consistência dos dados de entrada. Isto significa que se, por exemplo, o seu programa pede um número entre 1 e 10 e o usuário digita um número negativo, uma letra, um cifrão, etc, o seu programa pode fazer qualquer coisa, como travar o computador ou encerrar a sua execução abruptamente com respostas erradas.

8. **Arquivo com o programa fonte**

   Seu arquivo contendo o programa fonte na linguagem C deve estar bem organizado. Um programa na linguagem C tem de ser muito bem compreendido por uma pessoa. Verifique se seu programa tem a indentação adequada, se não tem linhas muito longas, se tem variáveis com nomes significativos, entre outros. Não esqueça que um programa bem descrito e bem organizado é a chave de seu sucesso. Não esqueça da documentação de seu programa e de suas funções.

   Dê o nome do seu usuário do Passaporte UFMS para seu programa e adicione a extensão `.c` a este arquivo. Por exemplo, `fulano.silva.c` é um nome válido.

9. **Conduta Ética**

   O trabalho deve ser feito INDIVIDUALMENTE. Cada estudante tem responsabilidade sobre cópias de seu trabalho, mesmo que parciais. Não faça o trabalho em grupo e não compartilhe seu programa ou trechos de seu programa. Você pode consultar seus colegas para esclarecer dúvidas e discutir idéias sobre o trabalho, ao vivo ou no fórum de discussão da disciplina, mas NÃO copie o programa!

   Trabalhos envolvidos em plágio, mesmo que parcial, terão nota ZERO.

   Além disso, NÃO utilize **assistentes ou geradores de IA**. Trabalhos com indícios de uso dessas ferramentas terão nota ZERO. O aluno poderá ser chamado para uma entrevista referente ao seu código.