



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
Rua Dom Manoel de Medeiros, s/n – Dois Irmãos 52171-900 Recife-PE
Fone: 81 3302 1000 www.dc.ufrpe.br

DISCIPLINA: Algoritmos e Estruturas de Dados	CÓDIGO: 06214
DEPARTAMENTO: Computação	ÁREA: Informática
CURSO: Licenciatura em Computação	
PROFESSOR RESPONSÁVEL: Luciano Demétrio Santos Pacífico	
DATA MÁXIMA DE ENTREGA: 31-10-2020	

Regras do Projeto 03 – Avaliação Final

1. **Não é permitido o uso de Estruturas de Dados prontas de Linguagens de Programação.** O aluno deve implementar suas próprias Estruturas de Dados. Na Linguagem de Programação **C**, deve-se usar **structs**. Nas demais Linguagens de Programação permitidas (vide **Regras da Disciplina**), deve-se usar **classes**.
2. **Não é permitido o uso de Algoritmos e comandos otimizados prontos de Linguagens de Programação.** Todos os algoritmos solicitados devem ser implementados pelos alunos como **procedimentos** (funções, métodos, etc.).
3. As questões que solicitam **escrita de código** devem ser resolvidas **apenas através dos recursos oferecidos pela pseudolingagem definida para a disciplina, e dos recursos equivalentes em Linguagens de Programação reais**, sendo eles: variáveis, constantes e tipos primitivos, expressões, estruturas condicionais, estruturas de repetição, sub-rotinas, estruturas de dados homogêneas (**Arrays**) e estruturas de dados heterogêneas (registros – classes e structs).
4. Para o Projeto 02, os **Arrays** devem ser implementados através de **Alocação Estática**. Não será permitido o uso de **Estruturas de Dados dinâmicas** implementadas em Linguagens de Programação, como os **Vectors** e **Lists** da Linguagem de Programação **Java**, por exemplo. Deve-se usar **arrays**.
5. Como a Linguagem de Programação **Python** não suporta **arrays estáticos**, o aluno que optar por usar esta Linguagem deverá usar **Lists**, **única e exclusivamente para simular o comportamento de arrays estáticos**, de acordo com os seguintes critérios:
 - a. Deve-se criar uma **List** com posições vazias através do método **append** dessa estrutura. O uso do **append** será permitido **apenas na alocação de memória para a variável que representará o array estático**;
 - b. O limite máximo de **M** objetos deve ser controlado **através de código**;
 - c. A Estrutura deve ser manipulada como se fosse um **array estático**, com os **procedimentos escritos pelo aluno, não sendo permitido** o uso de métodos, funções ou otimizações oferecidos pela Linguagem **Python**.
6. **Regra de Ouro:** Todos os alunos envolvidos em **cópias** terão suas notas **ANULADAS** nas referidas questões.
7. O arquivo de resposta com o código do projeto deve ser nomeado na forma “P03.c”, “P03.java”, etc.
8. A resposta do Projeto 03 deve ser submetida **unicamente através da tarefa criada no Google Classroom para este propósito**.

Projeto 03 – Avaliação Final

Um jogo de cartas é composto por **52 (cinquenta e duas) cartas**, distribuídas em **4 (quatro) naipes: ouros, paus, copas e espadas**. Cada naipe contém exatamente **13 (treze) cartas**, às quais são atribuídos valores

que podem ser ordenados entre si. As cartas em cada naipe são: A (corresponde ao valor 1 – um), 2, 3, 4, 5, 6, 7, 8, 9, 10, J (11), Q (12), K (13). Implemente, em uma **Linguagem de Programação**, as **Estruturas de Dados** e **Procedimentos** necessários para representar esse jogo, de acordo com o que é especificado abaixo.

PRELIMINARES

O jogo suporta entre **2 (dois)** e **4 (quatro)** jogadores.

No início de uma partida, as 52 cartas devem ser **embaralhadas** (organizadas aleatoriamente), e, em seguida, colocadas em uma **pilha única (pilha de compra)**, com as faces viradas para baixo.

Os jogadores devem se organizar em uma **Lista Circular**.

No início do jogo, o primeiro jogador receberá a carta do topo da pilha de compra, e em seguida, o segundo jogador receberá a nova carta do topo da pilha de compra, e assim por diante, até que **cada jogador** tenha comprado um total de **9 (nove) cartas**, compondo a **mão do jogador**.

As cartas restantes da pilha de compra devem ser mantidas nessa estrutura, com suas faces viradas para baixo.

Durante a execução do jogo, uma nova pilha será formada: a **pilha de descarte**, explicada em detalhes na sequência.

FUNCIONAMENTO

O jogo será um **jogo baseado em turnos**, no qual cada jogador executa suas ações, e em seguida, os demais jogadores executarão suas ações na sequência, em seus respectivos turnos.

No turno de um jogador, existem **duas ações possíveis e obrigatórias**:

1. O jogador inicia seu turno pela **compra de uma carta**, podendo esta compra ser realizada pela compra de uma carta da **pilha de compra** (cartas com **face para baixo**) ou da **pilha de descarte** (cartas com faces **viradas para cima, se houver uma pilha de descarte**). Essa ação resultará em um total de **10 (dez) cartas** na mão do jogador.
2. Após a compra, o jogador deve escolher uma carta de sua mão e descartá-la, acrescentando-a ao topo da **pilha de descarte**, com sua face voltada para cima. Essa ação resultará no reestabelecimento de uma mão contendo **9 (nove) cartas** para o jogador.

Se após a compra de algum jogador a **pilha de compra esvaziar** (ou seja, não houver mais cartas nessa pilha), deve-se embaralhar (organizar aleatoriamente) as cartas da pilha de descarte, empilhando-as com suas faces viradas para baixo, de modo a **compor uma nova pilha de compra**, o que esvaziará a pilha de descarte.

O turno encerra imediatamente após a ação de descarte por parte do jogador, podendo nesse momento uma análise da **condição de vitória** ser realizada (vide seção **Sistema do Jogo**).

CONDIÇÃO DE VITÓRIA

Os **pontos no jogo** são obtidos quando **um conjunto de 3 (três) cartas é formado**, atendendo a um dos seguintes critérios:

- As três cartas são de **um mesmo naipe**, formando **uma sequência ordenada, em ordem crescente, de passo 1** (ou seja, as cartas se sucedem, considerando a sequência completa de possíveis valores). Ex.: as três cartas são do naipe de ouros, sendo elas “9”, “10” e “J”. As cartas ordenadas em ordem crescente vão de “A” até “K” (como indicado pelos valores na descrição do projeto).
- As três cartas são de **naipes diferentes**, porém possuem o mesmo valor. Ex.: “A” de copas, “A” de espadas e “A” de paus.
- As três cartas são de **um mesmo naipe**, e formam a sequência especial “Q”, “K” e “A”.

Um jogador é declarado **vitioso** se obtiver um total de **3 (três)** pontos no jogo em um mesmo turno (a pontuação não é cumulativa entre turnos diferentes), condição na qual necessariamente **todas as cartas em sua mão deverão estar sendo utilizadas para a formação de algum conjunto de três cartas**.

Ao final do turno de um jogador (imediatamente após a sua ação de descarte), o mesmo pode **declarar vitória** (se já obtiver três conjuntos válidos de três cartas) ou **passar o turno para o próximo jogador**.

SISTEMA DO JOGO

O Sistema de Controle do jogo deve oferecer um **menu interativo aos usuários**, que lhes permita executar as ações no decorrer da execução do jogo, de acordo com o especificado em cada fase.

No início de cada jogo, o sistema deve perguntar a quantidade de jogadores que participarão da partida.

Em seguida, o sistema deve distribuir as cartas entre os jogadores e formar a pilha de compras, de acordo com o processo descrito na seção de Preliminares.

No turno de um jogador, o sistema deve permitir ao usuário a execução das seguintes operações:

- Verificar qual a carta no topo da pilha de descarte;
- Verificar sua mão (impressão de todas as cartas na mão do jogador);
- Comprar uma carta da **pilha de compras** ou da **pilha de descarte** (se houver pilha de descarte).

Após a compra da carta por parte do jogador, o sistema deve **imprimir o valor e naipe da carta comprada**, e em seguida oferecer ao mesmo as seguintes opções:

- Verificar sua mão (impressão de todas as cartas na mão do jogador);
- Verificar qual a carta no topo da pilha de descarte;
- Descartar uma carta.

Após o descarte de uma carta, o sistema deve oferecer ao jogador apenas uma das seguintes opções:

- Passar o turno, que encerra o turno do jogador atual, iniciando o turno do jogador seguinte.
- Declarar vitória.

A operação de declaração de vitória será executada da seguinte forma:

- O sistema solicitará a indicação de uma carta da mão do jogador por vez, marcando essa carta como **usada**.
- O jogador deve indicar ao sistema as cartas que pretende usar para compor os conjuntos de três exatamente na ordem de uso. Uma mesma **carta não pode ser usada simultaneamente para compor mais de um conjunto**, para fins de pontuação.
- Após receber a indicação de **três cartas não marcadas de um jogador**, o sistema verifica se as mesmas correspondem a **um conjunto válido para a pontuação ou não**. Em caso **afirmativo**, o jogador marcará um ponto, e o sistema solicitará mais cartas, até que um conjunto total de três pontos seja obtido, situação na qual o sistema deve imprimir a **mensagem de vitória**. Em caso **negativo**, o sistema solicitará que o usuário recomece a inserção de cartas, por até **três tentativas**. Se as três tentativas resultarem em falha, o sistema deve automaticamente passar o turno para o **próximo jogador**.

A **mensagem de vitória** será composta pela declaração do jogador vencedor, assim como a impressão dos três conjuntos de cartas desse jogador que resultaram em sua vitória.