FACULDADE SENAC-POA / Escola de Tecnologia

Algoritmos e Programação III
Atividade Fixação Teórica e Prática
Representação Física e Manipulação dos Dados
Conceitos de Estruturas de Dados
Tipos Abstratos de Dados
Listas

Regras e Dicas no final do documento.

Questões de Fixação – Teóricas:

Questão 1: Assinale a alternativa que completa corretamente o texto a seguir:

"Quando um programa é compilado, o binário executável fica armazenado em **(A)**. Quando o usuário executa uma chamada para esse executável, o binário é transferido para **(B)**. Durante sua execução, existem três espaços reservados: **(C)**."

Os itens A, B e C correspondem a:

a. memória RAM / memória cache / espaço de código, espaço estático e espaço Dinâmico (Pilha e *Heap*).
b. monitor; lixeira; estático, dinâmico e virtual.
c. disco rígido; memória RAM; espaço de código, espaço estático e espaço Dinâmico (Pilha e *Heap*).
d. memória virtual; memória local; espaço virtual, espaço estático e espaço dinâmico.

1.1) Gabarito: __
1.2) Justifique as alternativas incorretas (com as suas palavras) em um parágrafo único:

_____
_____
_____
_____

Questão 2: Diferencie Pilha e *Heap* nos seguintes itens: ***(i)*** escopo de variáveis; ***(ii)*** alocação estática e dinâmica; ***(iii)*** gerenciamento de memória.

_____
_____
_____
_____

Questão 3: Explique com suas palavras as seguintes definições:

a. Tipos de dados primitivos
b. Tipos de dados estruturados
c. Tipos de dados abstratos

_____
_____
_____
_____

Questão 4: Tipos de dados abstratos são fundamentais para o desenvolvimento de sistemas complexos, pois eles permitem uma camada de abstração a mais para que o programador possa desenvolver seus códigos com maior facilidade. **Justifique essa afirmativa**.

_____
_____
_____
_____

Questão 5: Na maioria das linguagens de programação, existem duas formas de representar coleções de dados: usando vetores/matrizes ou listas encadeadas.

a. Vetores em (de preferência em C ou Java) são sempre estáticos? Justifique sua resposta.
b. Qual a diferença entre vetores e listas encadeadas?
c. Quando se trabalha com alocação dinâmica, um problema muito comum é a fragmentação de memória durante a execução do programa. O que é fragmentação de memória? Como as listas encadeadas amenizam esse problema quando comparadas a vetores?

_____
_____
_____
_____

Questão 6: Crie estruturas abstratas que representem cada uma das seguintes definições:

Atenção estas estruturas devem ser definidas apenas no documento, não é necessária a implementação, sigo o exemplo exposto abaixo.

a. [EXEMPLO] "**Data** é o modo pelo qual se define um certo momento no tempo. Normalmente dá-se esse nome a uma forma de designar o número ou o nome de um Dia, Mês ou Ano, muitas vezes de forma conjunta."

a) Data	`class DateStruct:`
	`  def __init__(self,day, month, year):`
	`  self.day = day`
	`  self.month = month`
	`  self.year = year`

b. "No nosso banco, cada **cliente**, é associado por um nome, um CPF, um telefone, um endereço, um CEP e uma conta. Cada **conta** possui a agência e o número da conta."

c. "Criamos a rede social ***MyBook***, onde cada **usuário** possui um identificador único, um e-mail, uma senha, uma lista de seguidores e um mural com mensagens de texto. Cada mensagem de texto pode ter no máximo 128 caracteres."

Questão 7: Sobre Tipo Abstrato de Dados (TADs), considerando a implementação de **Lista de Ingredientes:**

Atenção estas estruturas devem ser definidas apenas no documento, não é necessária a implementação, sigo o exemplo exposto abaixo.

7.1) Crie um registro/*struct* em que representa o TAD **Ingrediente**. Esse registro deve conter três campos: nome, quantidade e medida.
Lembrando a forma de como criar TADs (acesse material de apoio) e o exemplo abaixo o qual apresenta a implementação do TAD Data.

Data	**Ingrediente**

`class DateStruct:`
`  def __init__(self,day, month, year):`
`  self.day = day`
`  self.month = month`
`  self.year = year`	

7.2) Após, defina, pelo menos, 4 métodos de acordo com o domínio do problema: **Lista de Ingredientes**. A definições devem seguir o modelo abaixo (nome do método, entrada(s), saída(s), objetivo) :

```
Método Exemplo acrescentarDias
Entradas: D (data), Dias (inteiro)
Saída: (Data)
Objetivo: método que recebe soma um determinado número de Dias a uma data
recebida como parâmetro e retorna o resultado. Caso não seja possível realizar a
operação, o método retorna uma data cujo Dia seja -1
```

Questões de Fixação – Práticas:

Questão 8: Acesse a implementação de Listas Encadeadas (Pasta da Aula 6 – Arquivo: CodigosListaEncadeada.zip).

8.1) Debug o código, realize testes de modo a entender a implementação e responder as perguntas abaixo:

a. Qual o objetivo do código?

b. Quantas classes são implementadas?

c. Qual o relacionamento entre elas?

d. Como os dados ficam armazenados na estrutura proposta?

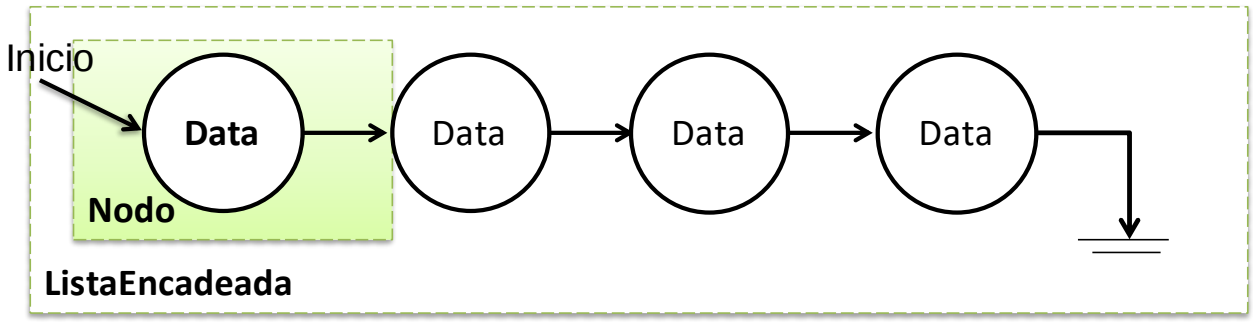
8.2) Escreva com suas palavras como funciona a implementação apresentada. Utilize diagramas (caixinhas), fluxogramas, mapa mental (entre outras alternativas) de modo que a explicação fique mais clara / didática possível.

Questão 9: Sobre o código disponibilizado:

a. Aponte possíveis melhorias no código Exemplo e implemente uma nova versão do código exposto de modo a melhorar os conceitos de: Orientação Objetos, Estrutura de Dados e Linguagem Python.

b. Faça as outras operações necessárias para manipular listas, especialmente o método de REMOVER.

Questão 10: **INDO ALÉM:** Implemente uma Lista de Datas (Tad definido nas aulas anteriores) utilizando a definição de listas encadeadas vista em aula. Utilize o código exemplo ou aprimorada para esta implementação.



Regras e Dicas

Informações de envio:

1. **Envio:**
   - **a. Onde?**
     A entrega deve ser feita via *Blackboard* (no link desta especificação).
   - **b. Quando?**
     Até o dia **21/04/2020** até às **19:00**.
   - c. **O que eu devo enviar?**
     Compactar o trabalho teórico e prático em pasta zipada.
   - d. **Onde eu realizo este envio?** via Atividades Avaliativas de modo a ficar oficializado o seu envio.