QUESTÃO 1 de 2 – Lista Encadeada

Enunciado: Com a finalidade de melhorar o atendimento e priorizar os casos mais urgentes, a direção de um hospital criou um sistema de triagem em que um profissional da saúde classifica a ordem de atendimento com base numa avaliação prévia do paciente, entregando-lhe um cartão numerado verde (V) ou amarelo (A), que define o menor ou maior grau de urgência da ocorrência, respectivamente. Para informatizar esse processo, a direção do hospital contratou você para desenvolver uma fila de chamada seguindo as seguintes regras:

* Pacientes com cartão numerado amarelo (A) são chamados antes dos pacientes com cartão numerado verde (V)
* Entre os pacientes com cartão numerado amarelo (A), os que tem numeração menor são atendidos antes.
* Entre os pacientes com cartão numerado verde (V), os que tem numeração menor são atendidos antes.
* As numerações dos cartões amarelos (A) iniciam em 201.
* As numerações dos cartões verdes (V) inicial em 1.

Elabore um programa em Python que:

1. Deve-se implementar uma **Lista Encadeada Simples** em que**:** [EXIGÊNCIA DE CÓDIGO 1 de 7];
   1. O Nodo representa um cartão numerado contendo: **número**, **cor** e um ponteiro para o **próximo**;
   2. A lista contém um ponteiro para a cabeça da lista (***head***);
2. Deve-se implementar a função **inserirSemPrioridade(nodo)** em que**:** [EXIGÊNCIA DE CÓDIGO 2 de 7];
   1. Deve-se andar pela lista a partir da cabeça (***head***) e inserir o nodo no **final da lista**.
3. Deve-se implementar a função **inserirComPrioridade(nodo)** em que**:** [EXIGÊNCIA DE CÓDIGO 3 de 7];
   1. Deve-se andar pela lista a partir da cabeça (***head***) e inserir o nodo **após todos os nodos com cor “A” que estão na lista**.
   2. O nodo inserido deve **sempre** estar antes de todos os nodos com cor “V”.
4. Deve-se implementar a função **inserir()** em que**:** [EXIGÊNCIA DE CÓDIGO 4 de 7];
   1. Deve-se solicitar ao usuário a cor (“A” ou “V”) e o número (inteiro).
   2. Deve-se criar um nodo com a cor e o número fornecidos pelo usuário.
   3. Se a lista estiver vazia, a cabeça (***head***) da lista deve apontar para o nodo criado.
   4. Senão, se a cor do nodo for “V”, deve-se chamar a função **inserirSemPrioridade(nodo).**
   5. Senão, se a cor do nodo for “A”, deve-se chamar a função **inserirComPriordade(nodo).**
5. Deve-se implementar a função **imprimirListaEspera()** em que: [EXIGÊNCIA DE CÓDIGO 5 de 7];
   1. Deve-se imprimir todos os cartões e seus respectivos números a partir do primeiro até o último da lista.
6. Deve-se implementar a função **atenderPaciente()** em que: [EXIGÊNCIA DE CÓDIGO 6 de 7];
   1. Deve-se remover o primeiro paciente da fila e imprimir uma mensagem chamando o paciente para atendimento informando o número do seu cartão.
7. Deve-se implementar um menu para utilização do sistema em que: [EXIGÊNCIA DE CÓDIGO 7 de 7];
   1. Deve-se apresentar as opções (1 – adicionar paciente a fila, 2 – mostrar pacientes na fila, 3 – chamar paciente, 4 – sair)
   2. Se escolhida a opção 1, chamar a função **inserir().**
   3. Se escolhida a opção 2, chamar a função **imprimirListaEspera().**
   4. Se escolhida a opção 3, chamar a função **atenderPaciente().**
   5. Se escolhida a opção 4, encerrar o programa.
   6. Se escolhida uma opção diferente as opções disponíveis, voltar ao item G.a.

Para testar o software, execute os seguintes passos e apresente a saída do console conforme exemplo de saída de console (próxima página):

1. Deve-se testar o sistema inserindo três (3) pacientes com cartão de cor “V”, dois (2) pacientes com cartão de cor “A”, dois (2) pacientes com cartão “V” e três (3) pacientes com cartão de cor “A”, **nessa respectiva ordem**. [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 3];
2. Deve-se apresentar na saída de console a impressão da lista de espera (opção 2 do menu principal). [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 3];
3. Deve-se apresentar na saída de console o atendimento de dois (2) pacientes (opção 3 do menu principal) e em seguida mostrar a lista de espera (opção 2 do menu principal). [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 3];

EXEMPLO DE SAÍDA DE CONSOLE:

A imagem apresenta a seguinte saída do console:
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>1
Informe a cor do cartão (A/V): V
Informe o número do cartão: 1
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>1
Informe a cor do cartão (A/V): V
Informe o número do cartão: 2
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>1
Informe a cor do cartão (A/V): V
Informe o número do cartão: 3
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>1
Informe a cor do cartão (A/V): A
Informe o número do cartão: 201
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair A imagem apresenta a seguinte saída do console:
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>1
Informe a cor do cartão (A/V): A
Informe o número do cartão: 202
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>1
Informe a cor do cartão (A/V): V
Informe o número do cartão: 4
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>1
Informe a cor do cartão (A/V): V
Informe o número do cartão: 5
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>1
Informe a cor do cartão (A/V): A
Informe o número do cartão: 203
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair A imagem apresenta a seguinte saída do console:
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>1
Informe a cor do cartão (A/V): A
Informe o número do cartão: 204
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>1
Informe a cor do cartão (A/V): A
Informe o número do cartão: 205
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair

**Figura 1: Exemplo de saída de console que o aluno deve fazer. Em que se insere 10 pacientes (5 com cartão verde e 5 com cartão amarelo) conforme** [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 3];

A imagem apresenta a seguinte saída do console:
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>2
Lista -> [A,201] [A,202] [A,203] [A,204] [A,205] [V,1] [V,2] [V,3] [V,4] [V,5] 
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair

**Figura 2: Exemplo de saída de console que o aluno deve fazer. Em que mostra a lista de pacientes, conforme** [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 3];

A imagem apresenta a seguinte saída do console:
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>3
Atendendo o paciente cartão cor A e número 201
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>3
Atendendo o paciente cartão cor A e número 202
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>2
Lista -> [A,203] [A,204] [A,205] [V,1] [V,2] [V,3] [V,4] [V,5] 
1 - Adicionar paciente a fila
2 - Mostrar pacientes na fila
3 - Chamar paciente
4 - sair
>>4

**Figura 3: Exemplo de saída de console que o aluno deve fazer. Em que ele chama dois pacientes para atendimento e em seguida mostra a lista de pacientes, conforme** [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 3];

Apresentação de **Código da Questão 1**:

SUBSTITUIR ESSE TEXTO QUE ESTÁ EM VERMELHO PELO SEU CÓDIGO DO EXERCÍCIO 1.

NÃO ESQUECER DE CUMPRIR AS EXIGÊNCIAS DE CÓDIGO!!

O CÓDIGO DEVE ESTAR IDENTADO!!

**SERÃO ACEITOS SOMENTE CÓDIGOS NO FORMATO TEXTO (NADA DE IMAGEM NEM PRINT, ZERA A QUESTÃO!).**

class Nodo:

    def \_\_init\_\_(self, numero, cor):

        self.numero = numero

        self.cor = cor

        self.proximo = None

class ListaEncadeada:

    def \_\_init\_\_(self):

        self.head = None

    def inserirSemPrioridade(self, nodo):

        if self.head is None:

            self.head = nodo

        else:

            atual = self.head

            while atual.proximo is not None:

                atual = atual.proximo

            atual.proximo = nodo

    def inserirComPrioridade(self, nodo):

        if self.head is None or self.head.cor == 'V':

            nodo.proximo = self.head

            self.head = nodo

        else:

            atual = self.head

            while atual.proximo is not None and atual.proximo.cor == 'A':

                atual = atual.proximo

            nodo.proximo = atual.proximo

            atual.proximo = nodo

    def inserir(self):

        cor = input("Digite a cor do cartão (A ou V): ").strip().upper()

        numero = int(input("Digite o número do cartão: "))

        nodo = Nodo(numero, cor)

        if self.head is None:

            self.head = nodo

        else:

            if cor == 'V':

                self.inserirSemPrioridade(nodo)

            else:

                self.inserirComPrioridade(nodo)

    def imprimirListaEspera(self):

        atual = self.head

        while atual is not None:

            print(f"Cartão {atual.numero} - Cor {atual.cor}")

            atual = atual.proximo

    def atenderPaciente(self):

        if self.head is None:

            print("Nenhum paciente na fila.")

        else:

            paciente = self.head

            self.head = self.head.proximo

            print(f"Chamando paciente com cartão {paciente.numero} para atendimento.")

def menu():

    lista = ListaEncadeada()

    while True:

        print("###################################")

        print("##             Menu              ##")

        print("## 1 - Adicionar paciente à fila ##")

        print("## 2 - Mostrar pacientes na fila ##")

        print("## 3 - Chamar paciente           ##")

        print("## 4 - Sair                      ##")

        print("###################################")

        opcao = int(input("Escolha uma opção: "))

        if opcao == 1:

            lista.inserir()

        elif opcao == 2:

            lista.imprimirListaEspera()

        elif opcao == 3:

            lista.atenderPaciente()

        elif opcao == 4:

            break

        else:

            print("Opção inválida. Tente novamente.")

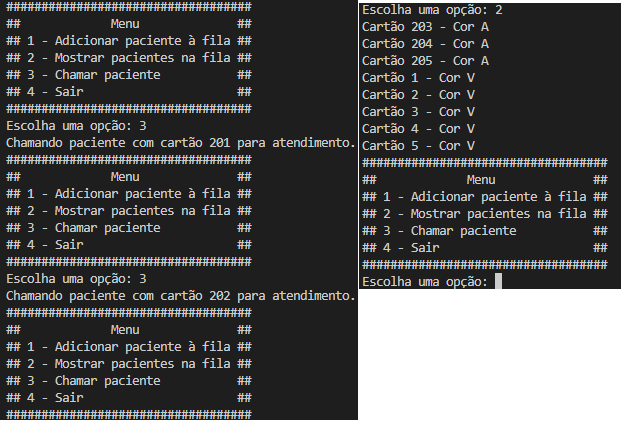
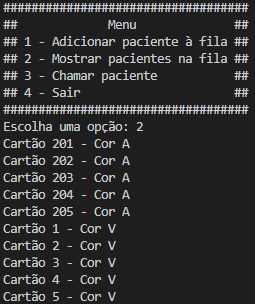
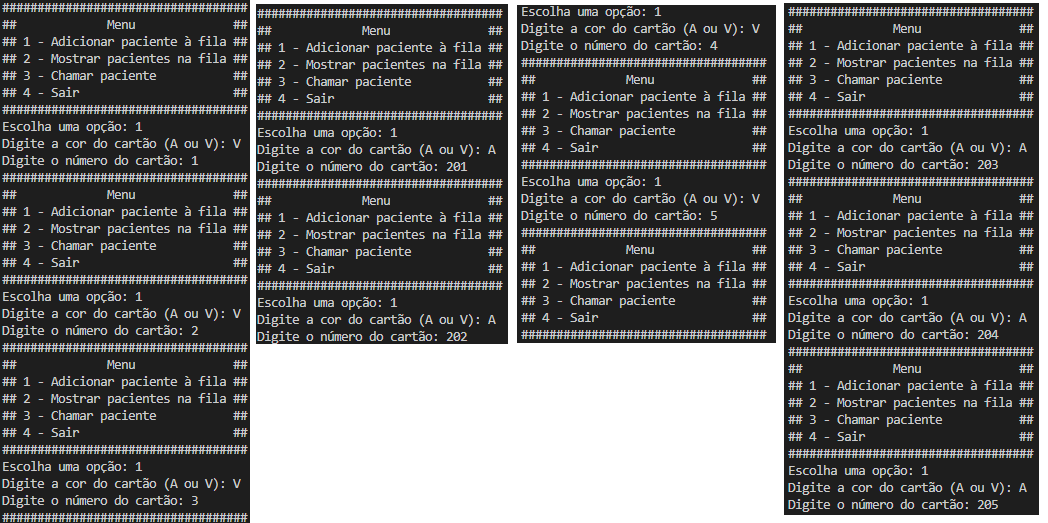
menu()

Apresentação de **Saída do Console da Questão 1**:

SUBSTITUIR ESSE TEXTO QUE ESTÁ EM LARANJA PELA A SAÍDA DO CONSOLE DO EXERCÍCIO 1

NÃO ESQUECER DE CUMPRIR AS EXIGÊNCIAS

**SERÁ ACEITO SOMETE SAÍDAS DO CONSOLE NO FORMATO IMAGEM (NADA DE TEXTO AQUI! ZERA ESSA PARTE DA QUESTÃO!)**



QUESTÃO 2 de 2 – Tabela Hash

Enunciado: Com o objetivo de criar um sistema novo de emplacamento de veículos, deputados em do Distrito Federal – DF, decidiram que o último número da placa dos veículos, irá representar o estado de registro dele. Para isso, sua equipe de desenvolvedores foi encarregada de desenvolver uma **Tabela Hash com endereçamento em cadeia de** **10 posições** (cada posição do vetor deve ser uma lista encadeada), representando os números de 0 a 9 que irão representar os 26 estados e o Distrito Federal (total 27).

A função hash deve seguir as seguintes regras:

* A entrada da função hash deve ser uma string com 2 letras, representando a sigla do estado e/ou distrito federal.
* Caso a sigla seja DF (Distrito Federal), por questões de superstição, os deputados solicitaram que o retorno da função seja 7 sempre.
* Caso contrário, a função deve retornar a posição com base no valor ASCII das duas letras e seguindo a seguinte regra:

Onde e são os valores ASCII da primeira e segunda letra, respectivamente (Tabela ASCII no final do documento).

Elabore um programa em Python que:

1. Deve-se implementar a tabela Hash com 10 posições, onde inicialmente todas as posições possuem valor **None** [EXIGÊNCIA DE CÓDIGO 1 de 7];
2. Deve-se implementar as **Listas Encadeadas Simples** em que**:** [EXIGÊNCIA DE CÓDIGO 2 de 7];
   1. O Nodo representa um Estado contendo: **sigla**, **nomeEstado** e um ponteiro para o **próximo**;
   2. As 10 posições da tabela hash, representam a cabeça de cada lista (***head***).
3. Deve-se implementar a inserção no início da lista encadeada (cada elemento novo deve ser sempre **inserido no início da lista**) [EXIGÊNCIA DE CÓDIGO 3 de 7];
4. Deve-se implementar a impressão da tabela hash, onde devem ser impressas as **siglas** de todos os nodos que estão na tabela hash **separados por posição** [EXIGÊNCIA DE CÓDIGO 4 de 7];
5. Deve-se implementar a função hash, conforme enunciado. [EXIGÊNCIA DE CÓDIGO 5 de 7];
6. Deve-se implementar a inserção dos estados e distrito federal (**todos os 27 com nome e sigla**) na tabela hash utilizando a função hash (não precisa solicitar ao usuário, pode inserir no código mesmo) [EXIGÊNCIA DE CÓDIGO 6 de 7];
7. Deve-se inserir na Tabela, além dos estados e distrito federal, um estado fictício, sendo que esse estado tenha seu **nome completo** e como siglas, a primeira letra do seu nome e a primeira letra do seu último sobrenome. Exemplo: Bruno Kostiuk – BK. EXIGÊNCIA DE CÓDIGO 7 de 7];

Para testar o software, execute os seguintes passos e apresente a saída do console conforme exemplo de saída de console (próxima página):

1. Deve-se apresentar na saída de console, a impressão da tabela hash antes de inserir qualquer informação [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 3];
2. Deve-se apresentar na saída de console, a impressão da tabela hash após inserir os 26 estados e o Distrito Federal - DF [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 3];
3. Deve-se apresentar na saída de console, a impressão da tabela hash após inserir os 26 estados, Distrito Federal – DF e o estado fictício com seu nome completo. [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 3];

A imagem apresenta a tabela ASCII disponível também em:

https://www.matematica.pt/util/resumos/tabela-ascii.php

EXEMPLO DE SAÍDA DE CONSOLE:

A imagem apresenta a seguinte saída do console:

0: None
1: None
2: None
3: None
4: None
5: None
6: None
7: None
8: None
9: None

**Figura 1: Exemplo de saída de console que o aluno deve fazer. Impressão da tabela hash antes de inserir qualquer informação, conforme** [EXIGÊNCIA DE SAÍDA DE CONSOLE 1 de 3];

A imagem apresenta a seguinte saída do console:

0: SC->RN->MS->GO->None
1: RO->MT->BA->AL->None
2: SE->PR->MA->ES->AM->AC->None
3: TO->SP->PI->None
4: RR->None
5: RS->PA->AP->None
6: RJ->PB->CE->None
7: DF->None
8: MG->None
9: PE->None

**Figura 1: Exemplo de saída de console que o aluno deve fazer. Impressão da tabela hash após inserir os 26 estados e o Distrito Federal - DF, conforme** [EXIGÊNCIA DE SAÍDA DE CONSOLE 2 de 3];

A imagem apresenta a seguinte saída do console:

0: SC->RN->MS->GO->None
1: BK->RO->MT->BA->AL->None
2: SE->PR->MA->ES->AM->AC->None
3: TO->SP->PI->None
4: RR->None
5: RS->PA->AP->None
6: RJ->PB->CE->None
7: DF->None
8: MG->None
9: PE->None

**Figura 1: Exemplo de saída de console que o aluno deve fazer. Impressão da tabela hash após inserir os 26 estados, Distrito Federal – DF e o estado fictício com seu nome completo (No caso foi inserido BK na posição 1), conforme** [EXIGÊNCIA DE SAÍDA DE CONSOLE 3 de 3];

Apresentação de **Código da Questão 2**:

SUBSTITUIR ESSE TEXTO QUE ESTÁ EM VERMELHO PELO SEU CÓDIGO DO EXERCÍCIO 2.

NÃO ESQUECER DE CUMPRIR AS EXIGÊNCIAS DE CÓDIGO!!

O CÓDIGO DEVE ESTAR IDENTADO!!

**SERÃO ACEITOS SOMENTE CÓDIGOS NO FORMATO TEXTO (NADA DE IMAGEM NEM PRINT, ZERA A QUESTÃO!).**

class Nodo:

    def \_\_init\_\_(self, sigla, nomeEstado):

        self.sigla = sigla

        self.nomeEstado = nomeEstado

        self.proximo = None

class TabelaHash:

    def \_\_init\_\_(self):

        self.tabela = [self.criarNodoNone() for \_ in range(10)]

    def criarNodoNone(self):

        return Nodo("None", "None")

    def hash(self, sigla):

        if sigla == "DF":

            return 7

        e1 = ord(sigla[0])

        e2 = ord(sigla[1])

        return (e1 + e2) % 10

    def inserir(self, sigla, nomeEstado):

        indice = self.hash(sigla)

        novo\_nodo = Nodo(sigla, nomeEstado)

        atual = self.tabela[indice]

        if atual.sigla == "None":

            self.tabela[indice] = novo\_nodo

            novo\_nodo.proximo = self.criarNodoNone()

        else:

            while atual.proximo and atual.proximo.sigla != "None":

                atual = atual.proximo

            novo\_nodo.proximo = atual.proximo

            atual.proximo = novo\_nodo

    def imprimirTabela(self):

        for i in range(len(self.tabela)):

            print(f"Posição {i}: ", end="")

            atual = self.tabela[i]

            while atual is not None:

                print(f"{atual.sigla} ", end="")

                atual = atual.proximo

            print()

def menu():

    tabela\_hash = TabelaHash()

    estados = {

        "AC": "Acre", "AL": "Alagoas", "AP": "Amapá", "AM": "Amazonas", "BA": "Bahia",

        "CE": "Ceará", "DF": "Distrito Federal", "ES": "Espírito Santo", "GO": "Goiás",

        "MA": "Maranhão", "MT": "Mato Grosso", "MS": "Mato Grosso do Sul", "MG": "Minas Gerais",

        "PA": "Pará", "PB": "Paraíba", "PR": "Paraná", "PE": "Pernambuco", "PI": "Piauí",

        "RJ": "Rio de Janeiro", "RN": "Rio Grande do Norte", "RS": "Rio Grande do Sul",

        "RO": "Rondônia", "RR": "Roraima", "SC": "Santa Catarina", "SP": "São Paulo",

        "SE": "Sergipe", "TO": "Tocantins", "MN": "Matheus Nakade"

    }

    while True:

        print("###################################")

        print("##             Menu              ##")

        print("##1 - Mostrar tabela hash        ##")

        print("##2 - Inserir estado             ##")

        print("##3 - Sair                       ##")

        print("###################################")

        opcao = int(input("Escolha uma opção: "))

        if opcao == 1:

            tabela\_hash.imprimirTabela()

        elif opcao == 2:

            sigla = input("Digite a sigla do estado (2 letras): ").upper()

            if sigla in estados:

                nomeEstado = estados[sigla]

                tabela\_hash.inserir(sigla, nomeEstado)

                print(f"Estado {nomeEstado} ({sigla}) inserido na tabela hash.")

            else:

                print("Sigla inválida. Tente novamente.")

        elif opcao == 3:

            break

        else:

            print("Opção inválida. Tente novamente.")

menu()

Apresentação de **Saída do Console da Questão 2**:

SUBSTITUIR ESSE TEXTO QUE ESTÁ EM LARANJA PELA A SAÍDA DO CONSOLE DO EXERCÍCIO 2

NÃO ESQUECER DE CUMPRIR AS EXIGÊNCIAS

**SERÁ ACEITO SOMETE SAÍDAS DO CONSOLE NO FORMATO IMAGEM (NADA DE TEXTO AQUI! ZERA ESSA PARTE DA QUESTÃO!)**

