
	SERVIÇO PÚBLICO FEDERAL MINISTÉRIO DA EDUCAÇÃO CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS UNIDADE DE ENSINO DESCENTRALIZADA DE DIVINÓPOLIS				
Curso:	Engenharia de Computação	Disciplina:	Introdução à Programação de Computadores	Período	1ª
Professor(a):	Tiago Alves de Oliveira			Valor:	10 pontos em IPC 25 pontos em LIPC
Data de Entrega:	30/06/2025	Trabalho:	1º		

Trabalho 1 - Individual

As explicações detalhadas dos trabalhos serão realizadas durante as aulas e nos laboratórios da disciplina, iniciando-se em 18/05/2025. A compreensão clara das instruções é essencial para a avaliação; a falta de explicação resultará em nota zero.

Exercício 1: Última Esperança Contra a Skynet

Descrição: No ano de 2045, a Skynet assumiu o controle de grande parte do mundo. Essa rede de inteligência artificial, originalmente desenvolvida para defesa, agora vê a humanidade como sua principal ameaça. Cidades inteiras foram transformadas em campos de caça, onde robôs implacáveis buscam eliminar todos os humanos.

Você é um líder rebelde lutando pela sobrevivência da humanidade. Sua missão é guiar um grupo de sobreviventes até a última zona de evacuação conhecida, um refúgio seguro dos horrores perpetrados pelos robôs. Curiosamente, os robôs da Skynet não conseguem invadir esta zona devido a um campo de força experimental desenvolvido por cientistas rebeldes. Este campo de força emite sinais que interferem nos circuitos de navegação dos robôs, tornando a área inacessível para eles.



No entanto, o caminho até lá está repleto de perigos, e cada movimento deve ser calculado com precisão para evitar os patrulheiros robóticos da Skynet.

Você deve implementar uma função chamada `fuga_humana` que analisa um mapa representando as áreas destruídas e infestadas de robôs. Este campo é representado por uma matriz bidimensional, onde os espaços podem ser seguros (' '), ocupados por robôs da Skynet ('R'), humanos fugitivos ('H'), ou a desejada zona de evacuação ('Z'). O objetivo é encontrar um caminho seguro, movendo-se para cima, para baixo, para a esquerda ou para a direita, evitando os robôs assassinos. Os passos do caminho seguro devem ser marcados com '*', e a função deve retornar o valor 1 (um) se um caminho seguro até a zona de evacuação for encontrado, ou 0 (zero) se não for possível encontrar um caminho. Caso encontre o caminho seguro, a função deve imprimir a string "Caminho encontrado" na tela e exibir o mapa com o caminho marcado.

Exemplo de entrada e saída:

Entrada:

			R			R		
	H			R				
			R		R			
R					R			R
		R					R	
				R				
		R				R	Z	

	SERVIÇO PÚBLICO FEDERAL MINISTÉRIO DA EDUCAÇÃO CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS UNIDADE DE ENSINO DESCENTRALIZADA DE DIVINÓPOLIS				
Curso:	Engenharia de Computação	Disciplina:	Introdução à Programação de Computadores	Período	1ª
Professor(a):	Tiago Alves de Oliveira			Valor:	10 pontos em IPC 25 pontos em LIPC
Data de Entrega:	30/06/2025	Trabalho:	1º		

Saída:

			R			R		
	H			R				
	*		R		R			
R	*	*	*	*	R			R
		R		*	*	*	R	
				R		*	*	
		R				R	Z	

Entrada:

R		R	R		R			
	H			R				
R			R			R		
		R			R		R	
		R						
			R					R
R		R		Z		R		

Saída:

Caminho não encontrado



Exercício 2: O Código Esquecido do Dr. Emmett Brown

No ano de 2099, a Terra está à beira de uma crise ambiental irreversível. As cidades estão sufocadas pela poluição e as fontes naturais de água e ar puro estão quase esgotadas. No entanto, uma descoberta surpreendente feita pelo exêntrico cientista Dr. Emmett Brown pode oferecer a solução.

Os documentos esquecidos do Dr. Brown revelam que ele descobriu uma máquina antiga, construída com base nos princípios da Sequência de Fibonacci, que pode reverter os danos ambientais em escala global. A máquina, projetada para restaurar equilíbrio e harmonia usando padrões naturais, só pode ser ativada inserindo uma série específica de números dessa sequência mística.

Desafio: Para ativar a máquina e salvar o planeta, os seguidores do Dr. Brown devem rapidamente calcular a Sequência de Fibonacci e localizar os números específicos que servem como chave de ativação. Cada segundo conta, pois a janela para reverter os efeitos devastadores está se fechando rapidamente.

Exercício Proposto - Busca Fibonacci: Você precisa implementar uma função chamada `busca_fibonacci`, que gera os n primeiros números da Sequência de Fibonacci e usa a busca binária para verificar se um dado número x está presente na sequência. Esta é uma etapa crítica para encontrar o código de ativação necessário para iniciar a máquina do Dr. Brown.

	SERVIÇO PÚBLICO FEDERAL MINISTÉRIO DA EDUCAÇÃO CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS UNIDADE DE ENSINO DESCENTRALIZADA DE DIVINÓPOLIS				
Curso:	<i>Engenharia de Computação</i>	Disciplina:	<i>Introdução à Programação de Computadores</i>	Período	<i>1ª</i>
Professor(a):	<i>Tiago Alves de Oliveira</i>			Valor:	<i>10 pontos em IPC 25 pontos em LIPC</i>
Data de Entrega:	<i>30/06/2025</i>	Trabalho:	<i>1º</i>		

Exemplos de Entrada e Saída:

Entrada: n = 10, x = 34 **Saída:** Índice: 9

Entrada: n = 10, x = 4 **Saída:** Índice: -1

Exercício 3: O Código Secreto de Avalon

Contexto: Em uma era onde o conhecimento antigo é a chave para descobrir novos mundos, uma jovem linguista e criptógrafa chamada Elise Silva se encontra no centro de uma aventura que poderia mudar a história da humanidade. Elise descobriu um antigo manuscrito que, segundo as lendas, contém as instruções para acessar a lendária cidade de Avalon, um lugar dito repleto de tecnologias avançadas e sabedorias perdidas. No entanto, o acesso a esses segredos depende da decodificação de uma série de enigmas palindrômicos intrincados que muitos acreditam conter as coordenadas e os mecanismos de abertura dos portais para Avalon.

Desafio: Os enigmas são complexos e apenas aqueles capazes de interpretar palíndromos com precisão podem esperar desvendar os mistérios do manuscrito. Cada palíndromo decifrado leva Elise um passo mais perto de Avalon, mas um erro pode levá-la a armadilhas perigosas ou despertar forças que melhor seriam deixadas adormecidas.

Exercício Proposto - Verificação de Palíndromos: Você deve ajudar Elise implementando uma função chamada `verificar_palindromo` que recebe uma string como entrada e verifica se ela é um palíndromo. A função deve retornar `True` se a string for um palíndromo e `False` caso contrário, considerando a necessidade de ignorar espaços em branco e a diferenciação entre letras maiúsculas e minúsculas.

Exemplos de Entrada e Saída:

Entrada: "radar" **Saída:** True

Entrada: "python" **Saída:** False

Entrada: "Ame a ema" **Saída:** True

Entrada: "A man, a plan, a canal, Panama" **Saída:** True

Regras:

- A função deve ignorar:
 - espaços em branco,
 - vírgulas,
 - pontuação
- A função deve desconsiderar diferenças entre letras maiúsculas e minúsculas na hora de comparar.