

Lista 1

Questão 1

Crie uma função que receba uma string e retorne o ponteiro para uma nova string invertida.

Questão 2

Crie uma estrutura **Car** com os membros: **brand**, **model**, **value**, **km_l** e **km_h**. Crie uma função que retorne o ponteiro para um vetor de **N** elementos tipo **structCar** alocado dinamicamente. Caso **N** seja negativo ou igual a zero, um ponteiro nulo deverá ser retornado. Use a função para criar um vetor de tamanho 5, preencha suas informações e imprima o vetor.

Questão 3

Escreva uma função que receba por parâmetro os valores inteiros e positivos **M** e **N** e retorne um ponteiro para uma matriz **M** × **N** alocada dinamicamente. Essa matriz deverá ter todas as posições inicializadas com zero.

Questão 4

Escreva uma função chamada **map** que receba como parâmetro um vetor de números reais **A** contendo **N** elementos, um número real **K** e um ponteiro para função (`float (*operation)(float a, float b)`). A função deve retornar o ponteiro para um novo vetor **B** de **N** elementos alocado dinamicamente, em que $B[i] = operation(A[i], k)$.

Questão 5

Considere a seguinte declaração: `int a, *b, **c, ***d;`. Escreva um programa que leia a variável **a** e calcule e exiba o dobro, o triplo e o quádruplo desse valor utilizando apenas os ponteiros **b**, **c** e **d**. O ponteiro **b** deve ser usado para calcular o dobro, **c**, o triplo, e **d**, o quádruplo.

Lista 2

Questão 1

Crie um arquivo de definição "Ponto2d.h" (definição de TAD para um "ponto" no plano bidimensional) com funções para: criação, destruição e para o cálculo da distância entre dois pontos.

Questão 2

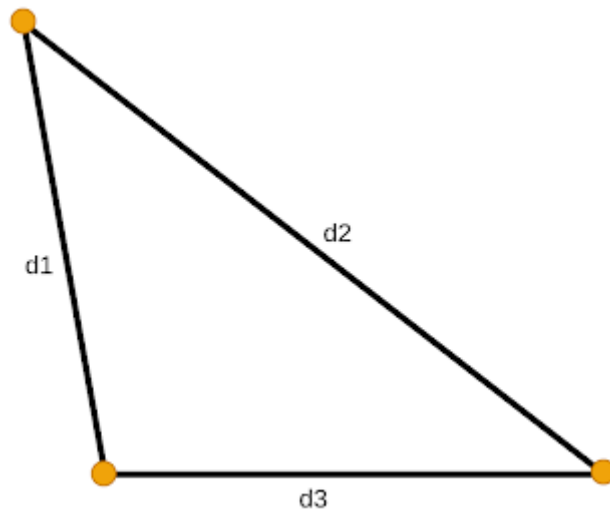
Crie um arquivo de implementação "Ponto2d.c" com as implementações das funções de criação, destruição e do cálculo da distância entre dois pontos no arquivo "Ponto2d.h".

Use o arquivo "Ponto2d_main.c" para testar sua implementação, para baixar esse arquivo use o link abaixo:

<https://drive.google.com/file/d/1z1nwCKYi2if8XfG4H8tts-9jPjaWEfiz/view?usp=sharing>

Questão 3

Crie um programa que recebe do usuário as coordenadas (x,y) dos três vértices de um triângulo e mostra o perímetro do triângulo. Use um `INCLUDE` para "Ponto2d.h" e use as funções que você já implementou. A imagem abaixo tem um exemplo de triângulo, seu perímetro pode ser obtido pela soma das distâncias **d1**, **d2** e **d3**.



Questão 4

Continue a implementação do arquivo **StackInt.c**. Links: (Stackint.c) https://drive.google.com/file/d/1LQuzqlVhTDXXq5CWEHc6D0ws_VVDdaIX/view?usp=sharing (StackInt.h) <https://drive.google.com/file/d/1JRxCMtCsxiwOeEIPUtClVof-1AVilHfM/view?usp=sharing>

Lista 3

Questão 1

Use como referência o arquivo **Queue.h** (ele pode ser encontrado na pasta Material de Apoio cujo link está no sistema) e implemente suas funções em um arquivo **Queue.c** para criar uma versão genérica de fila.

Questão 2

O programa **palindromeChain.c** (pasta Material de Apoio) não reconhece "**Amor a Roma**" como uma cadeia palíndroma. Use a função `toupper()`, declarada em **ctype.h**. Para resolver esse problema (essa função converte uma letra minúscula em maiúscula).

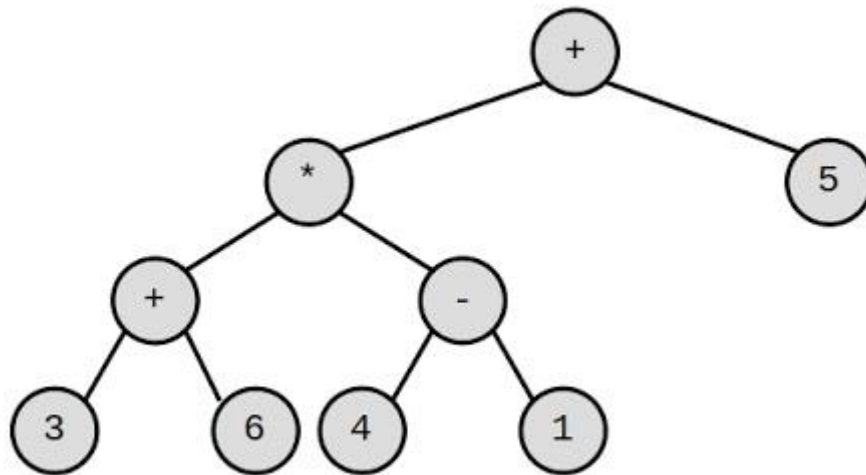
Questão 3

Usando coloração limitada por borda, crie uma versão da função `coloringArea()` no arquivo **coloring.c** (pasta Material de Apoio) e faça um programa para testá-la.

Lista 4

Questão 1

Implemente um TAD árvore binária de expressões aritméticas. Por exemplo, a expressão $(6-3)*(4+1)+5$ é representada pela árvore binária ilustrada na figura abaixo. As folhas da árvore armazenam operandos e os nós internos operadores. Teste sua implementação com a expressão desta árvore que resulta no valor 20.



Questão 2

Escrever uma função que calcule a altura de uma árvore binária dada. A altura de uma árvore é igual ao máximo nível de seus nós.

Questão 3

Escrever o algoritmo de visita em Pré-Ordem utilizando sem utilizar procedimentos recursivos.

Questão 4

Fazer função para retornar o pai de um dado nó de uma árvore: (a) dado um item, procura se item existe na árvore (usando algum algoritmo de travessia) (b) caso positivo retorna o conteúdo do pai do nó.

Lista 5

Questão 1

Escreva uma função não-recursiva que recebe uma árvore binária de busca BST como parâmetro e retorna o ponteiro para o nó cuja chave possui o valor mínimo ou NULL caso a árvore esteja vazia: `BST BST_min(BST bst);`

Questão 2

Escreva uma função não-recursiva que verifica a existência de algum elemento com chave negativa na árvore: `int BST_negativeKey(BST bst);`

Questão 3

Considere que você dado um vetor ordenado você precisa construir uma árvore de busca que contenha os mesmos elementos. Como você faria a construção da árvore para evitar que esta ficasse desbalanceada?: `BST vecToTree(int vec[], int length);`

Questão 4

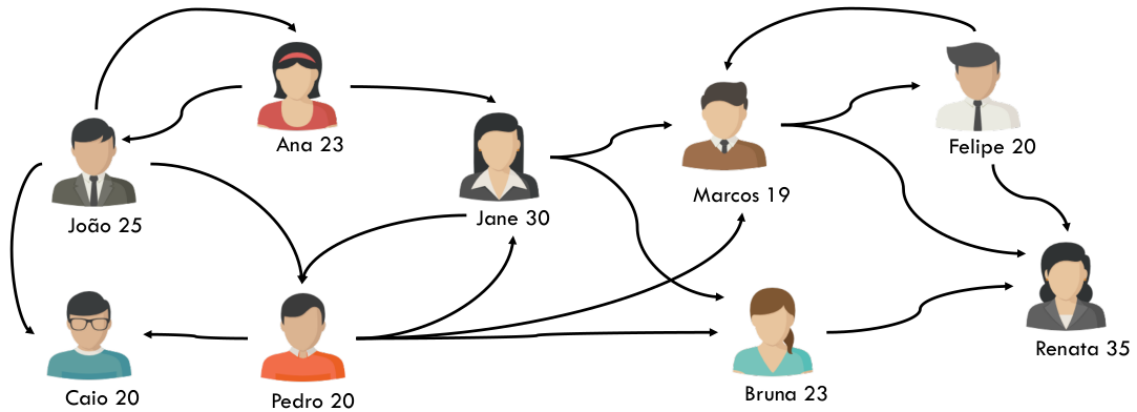
Crie funções para (a) contar o número de nós em uma árvore binária (b) contar o número de folhas (c) contar o número de filhos direitos (d) para encontrar a altura da árvore (e) para deletar todas as folhas de uma árvore binária.

Lista 6

Observação

Considere o grafo na figura seguir, que representa seguidores no Instagram. Cada pessoa tem nome e idade (nome é o id do vértice). Uma aresta (v1 , v2) significa que v1 segue v2 no Instagram.

Grafo de seguidores



Questão 1

Implemente a função: `int contar_seguidos(Grafo *g, char *nome);` que retorna quantas pessoas uma determinada pessoa segue.

Questão 2

Implemente a função: `int contar_seguidores(Grafo *vertice, char *nome, int imprime);` que retorna quantos são os seguidores de uma determinada pessoa. A função retorna quantidade de seguidores e, caso a flag `imprime` contenha um valor booleano verdadeiro (0 = false, 1 = true), também deve imprimir os nomes dos seguidores.

Questão 3

Implemente a função: `Vertice *mais_popular(Grafo *g);` que retorna o vértice que contém a pessoa com maior número de seguidores.

Questão 4

Implemente uma função: que retorna o número de pessoas que só seguem pessoas mais velhas do que ela própria? A função retorna quantidade de pessoas e, caso a flag `imprime` contenha um valor booleano verdadeiro (0 = false, 1 = true), também deve imprimir os nomes das pessoas.