sqtpm
[ggm]

ajuda
voltar

Trabalho: 24-reconstrucao

Data de abertura: 27/05/2025 10:00:00
Data limite para envio: 03/06/2025 23:59:00 (aberto)

Número de arquivos a enviar: entre 1 e 3

Número máximo de envios: 30
Número de envios feitos por ggm: 0

Casos-de-teste abertos: casos-de-teste.tgz

Recursos: cpu 1 s, memória 32768 kb, stack 8192 kb

Enviar:

Linguagem: C

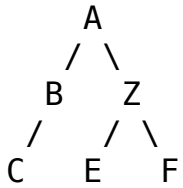
Arquivos: Browse... No files selected.

Enviar

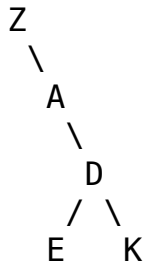
Reconstrução de árvores binárias

É sempre possível reconstruir uma árvore enraizada binária com filhos ordenados a partir das seqüências de nós visitados em pré-ordem e em-ordem.

Por exemplo, dadas as cadeias ABCZEF e CBAEZF, resultantes das impressões das chaves em pré-ordem e em-ordem respectivamente, podemos reconstruir a árvore abaixo.



Como outro exemplo, dadas as cadeias ZADEK e ZAEDK, resultantes das impressões das chaves em pré-ordem e em-ordem respectivamente, podemos reconstruir a árvore abaixo.



Escreva um programa para reconstruir uma árvore binária na memória a partir das seqüências pré-ordem e em-ordem. Depois

sqtpm

[ggm]

ajuda
voltar

imprima as seqüências de nós na ordem de percursos em pós-ordem e em largura.

Entrada

A entrada para o programa consiste de vários casos-de-teste. Cada caso-de-teste é formado por duas cadeias que são os símbolos nos nós da árvore binária impressos em pré-ordem e em-ordem, respectivamente.

As cadeias são formadas por símbolos ASCII entre 33 e 126 inclusive, distintos.

Saída

A saída deve ter uma linha com os percursos pós-ordem e largura para cada caso-de-teste separados por um espaço.

Exemplos

Entrada	Saída
ABCZEF CBAEZF ZADEK ZAEDK A A ab ba Yxz xYz fxy yxf abc bac weq qew abg bga acd adc abt bta poeq eopq qwte wtqe cdef fedc ACBD ABDC ABCDEF CBAEDF	CBEFZA ABZCEF EKDAZ ZADEK A A ba ab xzY Yxz yxf fxy bca abc qew weq gba abg dca acd tba abt eoqp poqe tweq qwet fedc cdef DBCA ACBD CBEFDA ABDCEF

Requisitos adicionais

- Seu programa deve reconstruir a árvore em representação explícita.
- Toda memória alocada dinamicamente deve ser liberada pelo seu programa (usando free()).
- Não pode haver qualquer variável global. Uma variável é global se estiver declarada fora de alguma função (variáveis declaradas dentro da main não são globais, são locais à função main).

sqtpm
[ggm]

ajuda
voltar

Sobre organização do código e comentários

- Faça um programa organizado, bem indentado e que seja fácil de ler.
- Adicione comentários que vão ser úteis para entender o programa se você for relê-lo daqui a alguns anos: comentar cada linha vai ser redundante; documentar blocos de código e a estratégia usada na solução vai ser muito útil.
