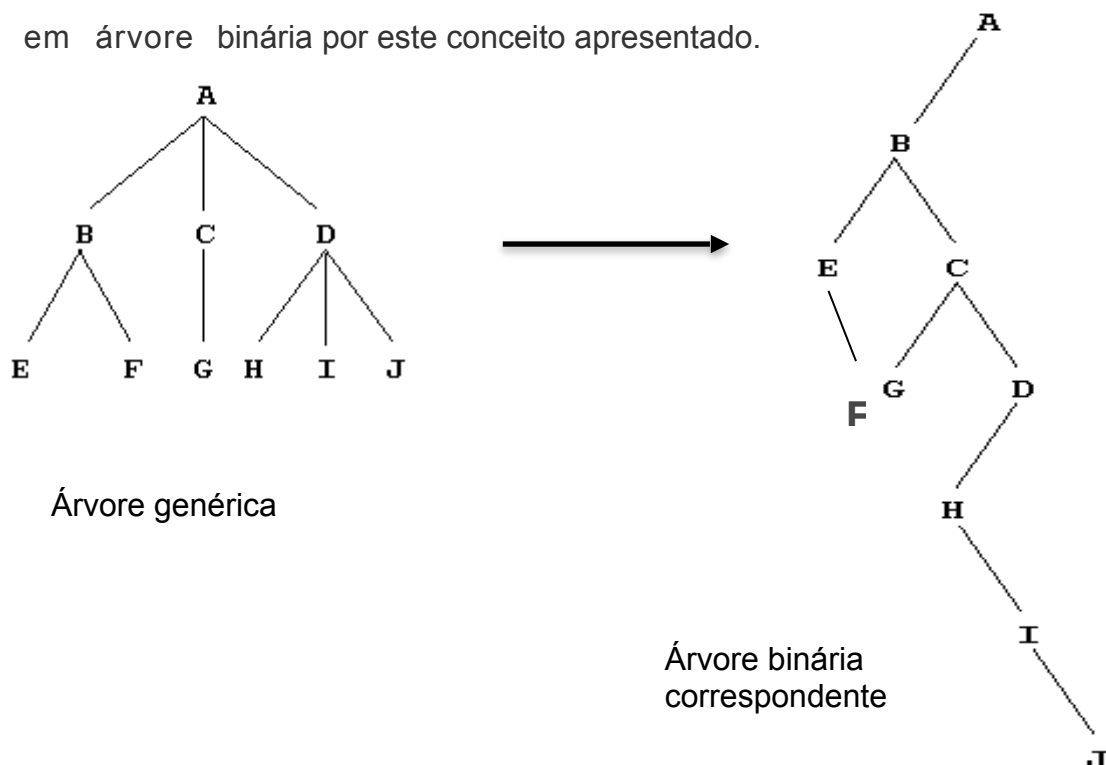


Algoritmos e Estruturas de Dados I – Tarefa da Semana 10 -

PESO 2.5

Atenção: inclua cabeçalho com seu nome completo, RA e Turma, comentários e INDENTAÇÃO no programa.

Uma árvore com grau maior que 2 (árvore genérica) pode ser transformada em uma árvore binária para facilitar a implementação. Para fazer isso, em uma árvore binária, à esquerda da árvore binária estará o filho primogênito da árvore genérica e à direita da árvore binária estará seu irmão correspondente na árvore genérica. Assim, a raiz principal da árvore binária que representa a árvore genérica nunca terá filho à direita, já que o nó principal não terá irmãos. O desenho abaixo ilustra como ficará a árvore genérica após a transformação em árvore binária por este conceito apresentado.



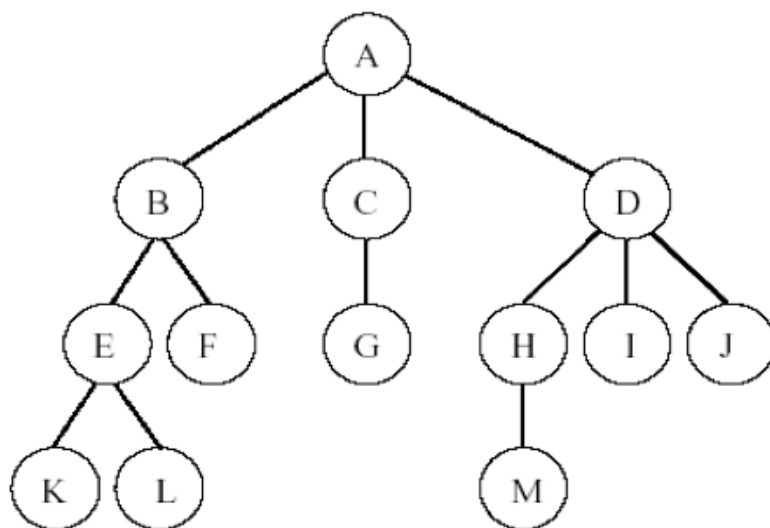
Veja que A não tem irmãos, portanto não tem filho à direita. À esquerda de qualquer nó sempre está seu irmão e à direita, seu primogênito. Por exemplo, B tem como irmãos C e D, portanto C fica à esquerda de B e D fica à esquerda de C. Assim, se quiser conhecer todos os irmãos de B, basta percorrer sempre à esquerda a partir de B e terá todos os seus irmãos. Para conhecer os filhos de B, basta pegar o primeiro descen-

dente (nó à esquerda, ou seja, nó E) e seus irmãos (no caso, nó à esquerda de E, que é o F). Para saber quem são os filhos de D, basta ir à direita de D e depois sempre à esquerda (nós H, I e J).

Faça um programa que converta uma árvore de qualquer grau maior ou igual a 1 em uma árvore binária. A árvore inicial será lida de um arquivo chamado arv.txt (não mude o nome do arquivo que tem que ser lido) em que cada linha do arquivo estará no seguinte formato:

✓ id num_filhos id_filho1 id_filho2 ... id_filhon

O campo *id* deverá ser do tipo char, pois irá armazenar uma letra que será usada para identificar o nó. Por exemplo, a árvore abaixo estará armazenada no arquivo da seguinte forma:



```

A 3 B C D
B 2 E F
E 2 K L
F 0
K 0
L 0
C 1 G
G 0
D 3 H I J
H 1 M
M 0
I 0
J 0
  
```

Imprima a árvore resultante em pré-ordem e na ordem infixa.

Faça um menu de opções quem que, após montada a árvore, o usuário poderá escolher entre as seguintes opções:

- 1) Imprimir a árvore resultante em pré-ordem;
- 2) Imprimir a árvore resultante em ordem infixa;
- 3) Imprimir os filhos de um dado nó (por exemplo, no desenho acima, se pedir para imprimir os filhos de D, deverá ser impresso H I J) (perguntar para o usuário qual o nó que ele quer conhecer os filhos);
- 4) Imprimir os irmão de um dado nó (por exemplo, no desenho acima, se quiser conhecer os irmãos do nó B, deverá ser impresso C e D);

5) Sair.

O programa só deverá ser finalizado quando o usuário escolher a opção 5. Não esqueça que esta opção significa ter de deslocar tudo o que for alocado.

Observações:

- 1) Será descontado ponto de programas com lógica confusa ou uso de breaks indevidos (por exemplo, para sair de laços ou em "if").
- 2) O exercício deverá ser implementado em linguagem C.
- 3) Erros de compilação: nota zero no exercício.
- 4) Tentativa de fraude: nota zero para todos os envolvidos.
- 5) Não uso de estrutura do tipo árvore binária: nota zero no exercício.
- 6) Serão avaliadas corretude e eficiência do código.
- 7) Faça as devidas verificações de erro no programa. A não observância a qualquer item incluído nestas especificações implicarão em perda de pontos ou até em pontuação zero no exercício.
- 8) **É imprescindível liberar tudo o que for alocado.**
- 9) Você deverá postar apenas o código do programa (.c) compactado no link disponível no Classroom. Exercícios enviados por qualquer outro meio ou postados **após a data ou horário limite** não serão corrigidos.