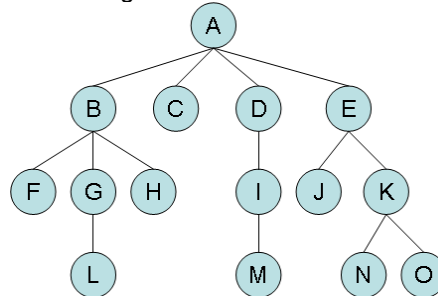


## Exercícios Árvores Binárias

1 - Converter a árvore apresentada a seguir em árvore binária.



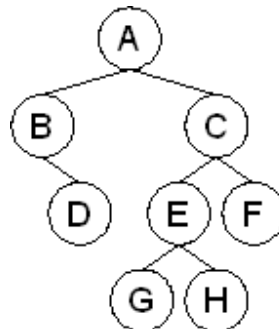
2 - Considerando a árvore binária gerada no exercício 1, execute as seguintes operações:

- Inserir o nodo X como filho a direita de A
- Inserir os nodos Y e Z como filhos de X
- Inserir os nodos P e Q como filhos de Z
- Excluir o nodo O e o nodo M
- Inserir os nodos R e S como filhos de L
- Inserir V como filho a esquerda de C

3 - Converter a árvore gerada no exercício para uma árvore n-ária equivalente.

4 - Para a figura apresentada a seguir, qual a ordem de exibição dos nodos, considerando :

- Caminhamento central à esquerda
- Caminhamento central à direita
- Caminhamento pré-fixado à esquerda
- Caminhamento pré-fixado à direita
- Caminhamento pós-fixado à esquerda
- Caminhamento pós-fixado à direita



5 – Considerando a estrutura de dados apresentada abaixo, verifique o que faz os trechos de código apresentados a seguir:

<pre> <b>Type</b> PNodo = ^nodo; nodo = <b>record</b>     info: Telem;     esq, dir: PNodo; <b>End;</b>  Type tree: PNodo;         </pre>
---

<p><b>A)</b>  <b>Procedure</b> Ordem1(N:tree);          Begin            If N &lt;&gt; nil Then            Begin              Imprime(N);              Ordem1(N^.esq);              Ordem1(N^.dir);            End;          End;</p>	<p><b>B)</b>  <b>Procedure</b> Ordem2(N:tree);          Begin            If N &lt;&gt; nil Then            Begin              Ordem2(N^.esq);              Imprime(N);              Ordem2(N^.dir);            End;          End;</p>	<p><b>C)</b>  <b>Procedure</b> Ordem3(N:tree)          Begin            If N &lt;&gt; nil Then            Begin              Ordem3(N^.esq);              Ordem3(N^.dir);              Imprime(N);            End;          End;</p>
<p><b>D)</b>  <b>Function</b> OQueSera (t: tree; item: Telem): integer;            Var n: integer;                achou:boolean;</p> <p><b>Procedure</b> Travessia(ptr: tree; VAR n: integer; item:Telem; VAR achou:boolean);          Begin            If ptr&lt;&gt;nil Then            Begin              n:=n+1;              If Igual(ptr^.info, item) Then                achou:=TRUE;              Else              Begin                Travessia(ptr^.esq, n, item, achou);                If (NOT achou) then                  Travessia(ptr^.dir, n, item, achou);                If (NOT achou) Then                  n:=n-1;              End;            End;          End;</p> <p>Begin {OQueSera}            achou:= false;            n:= 0;            Travessia(t,n,item, achou);            OQueSera :=n;          End;</p>		

**E)**

```
Function OQueSera2(t: tree; item:Telem):Telem;
  Var achou: boolean;
      it : Telem;

Procedure Travessia(t: tree; item: Telem; var it: Telem; var
achou:Booleam);
  Begin
    If not Vazia(t) Then
      Begin
        If t^.esq<>nil Then
          If Igual(item, t^.esq^.info) Then
            Begin
              achou:=true;
              it:=t^.info;
            End;
          If not achou Then
            If t^.dir<>nil Then
              If Igual(item, t^.dir^.info) Then
                Begin
                  achou:=true;
                  it:=t^.info;
                End;
              If not achou Then
                Travessia(t^.esq, item, it, achou);
              If not achou Then
                Travessia(t^.dir, item, it, achou);
            End;
          End; {Travessia}

  Begin { OQueSera2}
    If not Vazia(t) Then
      Begin
        achou:=false;
        If Igual(item, t^.info) Then
          OQueSera2:=t^.info;
        Else
          Begin
            Travessia(t,item,it,achou);
            OQueSera2:=it;
          End;
        End;
      End;
  End;
End;
```