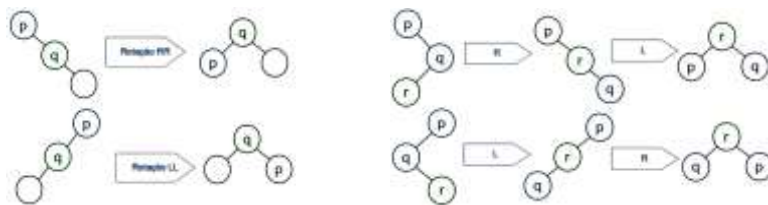


UDESC-CCT-DCC  
Prova online de EDA  
Resolva com um texto manuscrito, digitalize e  
envie sua solução pelo Moodle

Aluno(a): \_\_\_\_\_

Data: \_\_\_\_/\_\_\_\_/\_\_\_\_

Todas as funções utilizadas (não nativas do C) devem ser implementadas.  
Após o início não será permitida a saída da sala sem a entrega definitiva da prova.  
Para as questões da AVL considere FB= He - Hd



1) (2,5) ABB sem o requisito de balanceamento:

A) Para a ABB realize as seguintes inserções da esquerda para a direita:

4, 3, 7, 5, 8, 10, 1, 2, 6

B) Da árvore resultante da solução do quesito anterior, remova as chaves: 4 e 7. Se necessário utilize a substituição pelo SUCESSOR em ordem.

2) (2,5) ABB balanceada:

Para a ABB-AVL (justificando quaisquer rotações necessárias), execute as seguintes inserções da esquerda para a direita: 1,2,3,4,5,6,7,10,8

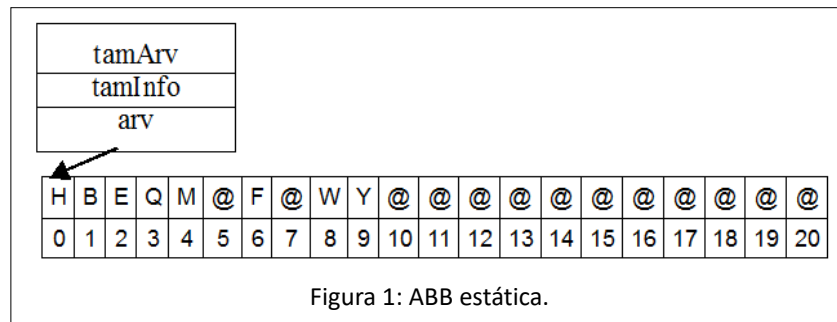
3) (2,5) Supondo uma B-tree de ordem **N=2** (cada página suporta no mínimo 2 e no máximo 4 chaves, exceto a raiz que pode guardar no mínimo 1 e no máximo 4 chaves), resolva justificando suas respostas:

A) Para a B-tree, insira as chaves 20,40,10,30,15;35,7,26,18,22 em sequência. Apresente as respectivas configurações da B-tree identificando as possíveis operações com a devida justificativa:

B) Para o resultado obtido na questão anterior, execute as seguintes remoções: 22, 35 e 20. Se for necessário, considere a substituição pelo ANTECESSOR em ordem.

4) (2,5) – Sabe-se que uma árvore binária pode ser representada em um vetor conforme as seguintes regras: a raiz está na posição zero, o filho esquerdo de uma posição  $i$  está em  $2i + 1$  e o filho direito de uma posição  $i$  está em  $2i + 2$ . Por sua vez, o pai de uma posição  $i$  está em  $(i-1) \div 2$ , onde o operador *div* retorna o quociente da divisão inteira, exemplos:

$1 \div 3 = 0$	$2 \div 3 = 0$	$3 \div 3 = 1$	$4 \div 3 = 1$	$5 \div 3 = 1$	$6 \div 3 = 2$	$7 \div 3 = 2$	$8 \div 3 = 2$	$9 \div 3 = 3$	$10 \div 3 = 3$
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	-----------------



Considerando árvore binária de caracteres representada na Figura 1, na qual o caractere “@” representa a inexistência de elemento naquela posição, qual será a sequência de caracteres exibida pela execução da função escrita no Quadro 1?

Justifique sua resposta com um teste de mesa sobre a árvore binária exibida na Figura 1, mostrando cada estado da pilha envolvida.

Quadro 1: Função que utiliza a ABB.

```

1. void funcaoY ( char * arv, int tamArv)
2. { Pilha *p; char c; int j = 0, i = 0;
3.   if ( tamArv == 0)
4.     return ;
5.   p=criaPilha (sizeof ( int ), tamArv); //cria uma pilha para valores inteiros
6.   if (arv [j] != '@')
7.     empilha(p, i); //empilha o valor atual da variável “i”
8.   while ( testaSePilhaVazia(p)==NAOVAZIA) //enquanto a pilha não estiver vazia
9.     { buscaNaPilha (p, &i); //busca o topo atual e o copia na variável “i”
10.      printf ("%c", arv[i]);
11.      desempilha(p); // remove o topo atual
12.      j = 2*i + 2;
13.      if (arv[j] != '@')
14.        empilha(p, j);
15.
16.      j = 2*i + 1;
17.      if (arv[j] != '@')
18.        empilha(p, j);
19.    }
20.   p=destroiPilha(p); //remove a pilha da memória
21. }
```