

Exercícios

Árvores Binárias

1. Implemente uma função que retorne a quantidade de folhas de uma árvore binária. Essa função deve obedecer ao protótipo:

```
int folhas (Arvore a) {  
  
}
```

2. Implemente uma função que compare se duas árvores binárias são iguais. Essa função deve obedecer ao protótipo:

```
Arvore igual (Arvore a, Arvore b) {  
  
}
```

Considere uma *árvore binária de busca* que armazena valores inteiros. Nesta estrutura, pode ocorrer repetições de um mesmo valor. Assim, os valores associados aos nós da sub-árvore à esquerda são *menores* que o valor associado à raiz e os valores dos nós da sub-árvore à direita são *maiores ou iguais*. O tipo que representa um nó da árvore é dado por:

```
class BintNo {  
    int valor;  
    BintNo esq, dir;  
  
    BintNo (int novoValor){  
        valor =  
        novoValor; }  
}
```

(a) Escreva uma função que retorne o número de ocorrências de um dado valor x na árvore. A função deve tirar proveito da ordenação da árvore e obedecer ao seguinte protótipo:

```
Int ocorrencias_x (Arvore a, int x) { }
```

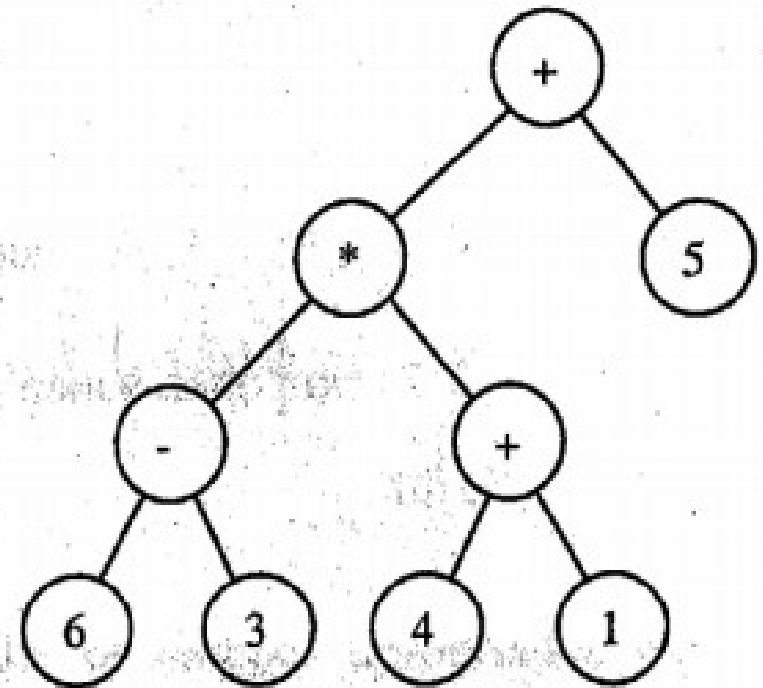
(b) Escreva uma função que imprima os valores associados às folhas da árvore em *ordem não crescente* (isto é, do maior para o menor, podendo haver repetições). A função deve obedecer ao seguinte protótipo:

```
Int imprime_folhas (Arvore a) { }
```

Considere uma árvore binária que representa expressões. Por exemplo, a expressão $(6-3) * (4+1) + 5$ é representada pela árvore binária ilustrada na figura.

As folhas da árvore armazenam operando e os nós internos operadores. Se avaliada, esta expressão resulta no valor 20.

Considere a existência do tipo abaixo usado para representar árvores binárias de expressões.



```
struct arv {  
    char op; /* operador: '+', '-', '*', ou '/' */  
    float valor; /* valor do operando */  
    struct arv *esq, *dir;  
};  
typedef struct arv Arv;
```

onde o campo value é usado apenas pelas folhas e o campo op pelos nós internos. Escreva funções que, dada a raiz de uma árvore binária de expressões (pode-se considerar que a árvore nunca será vazia):

(a) Imprima a expressão em notação pós-fixada. Para exemplo acima, imprimiria:
6 3 - 4 1 + * 5 +. O protótipo da função deve ser:

```
void imprime (Arv* a);
```

(b) Retorne o valor correspondente à avaliação da expressão (para o exemplo acima, retornaria 20). O protótipo da função deve ser:

```
float avalia (Arv* a);
```

Árvores Genéricas

4. Implemente uma função que retorne a quantidade de folhas de uma árvore com número variável de filhos. Essa função deve obedecer ao protótipo:

```
int folhas (Arvore a) {  
  
}
```

5. Implemente uma função que compare se duas árvores genéricas são iguais. Essa função deve obedecer ao protótipo:

```
Arvore igual (Arvore a, Arvore b) {  
  
}
```