

Estes exercícios devem ser entregues no Google Classroom. O objetivo desta lista é implementar as principais operações em árvores com número variável de filhos na linguagem C.

Questões:

1. **Implementação da árvore.** Inicialmente crie os seguintes arquivos fonte:
 - **principal.c:** função main
 - **arvvar.h:** declarações de estruturas, tipo e funções da árvore
 - **arvvar.c:** implementação das funções da árvore
2. **Estruturas e tipos de dados.** Iremos representar um nó da árvore com número variável de filhos por meio de uma estrutura em C contendo a informação armazenada no nó e dois ponteiros para o primeiro filho e para o irmão do nó conforme o código apresentado abaixo. Neste exercício, cada nó da árvore armazena um dado do tipo caracter:

```
struct arvvar {  
    char info;  
    struct arvvar* prim;  
    struct arvvar* prox;  
};  
  
typedef struct arvvar ArvVar;
```

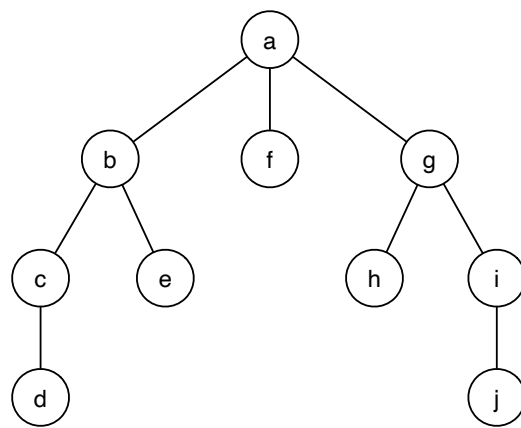
3. **Interface do tipo abstrato Árvore com Número Variável de Filhos.** A seguir temos os protótipos das funções a serem implementadas. Os protótipos devem estar declarados no arquivo arvvar.h, e as respectivas implementações devem estar no arquivo arvvar.c:

Função	Descrição
ArvVar* arvvar_cria(char c);	esta função cria um nó folha e retorna seu endereço de memória. Sendo um nó folha, seus ponteiros para prim e prox devem ser inicializadas como NULL
void arvvar_insere(ArvVar* a, ArvVar* sa);	insere uma sub-árvore sa como filha de um dado nó a
void arvvar_imprime(ArvVar* a);	deve imprimir a árvore em pré-ordem (primeiro o nó raiz, e depois cada uma das sub-árvores)
int arvvar_pertence(ArvVar* a, char c);	retorna 1 caso o caracter c estiver contido na árvore, e retorna 0 caso contrário
void arvvar_libera(ArvVar* a);	libera a memória alocada pela árvore
int arvvar_folhas(ArvVar* a);	retorna a quantidade de nós folha da árvore
int arvvar_altura(ArvVar* a);	retorna a altura da árvore (0 para árvore vazia)

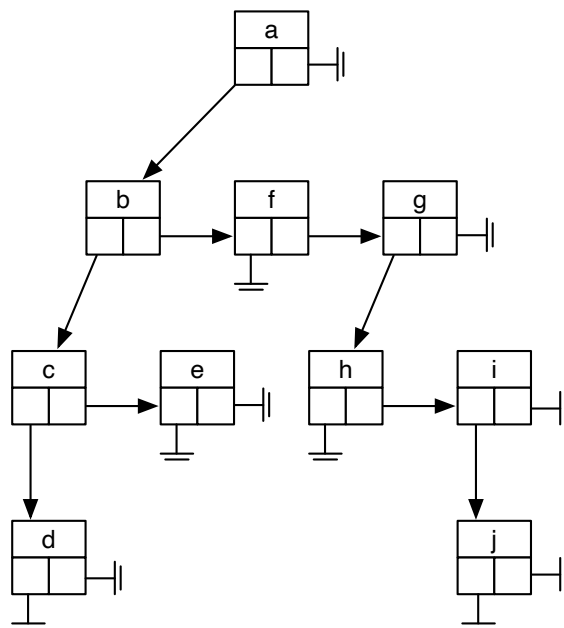
<code>int arv_v_igual(ArvVar* a1, ArvVar* a2);</code>	retorna 1 se as árvores a1 e a2 são iguais. Retorna 0 caso contrário. As duas árvores são iguais se seus nós armazenarem informações iguais nas mesmas posições
<code>ArvVar* arv_v_copia(ArvVar* a);</code>	retorna uma cópia da árvore a. A nova árvore deve ter novos nós instanciados

4. **Demonstração do funcionamento da estrutura.** No arquivo principal.c, implemente uma demonstração de todas as funções implementadas. Abaixo, segue um exemplo de criação de árvore utilizando as funções descritas acima.

```
/* Cria nós como folhas */
ArvVar* a = arv_v_cria('a');
ArvVar* b = arv_v_cria('b');
ArvVar* c = arv_v_cria('c');
ArvVar* d = arv_v_cria('d');
ArvVar* e = arv_v_cria('e');
ArvVar* f = arv_v_cria('f');
ArvVar* g = arv_v_cria('g');
ArvVar* h = arv_v_cria('h');
ArvVar* i = arv_v_cria('i');
ArvVar* j = arv_v_cria('j');
/* monta a hierarquia */
arv_v_insere(c,d);
arv_v_insere(b,e);
arv_v_insere(b,c);
arv_v_insere(i,j);
arv_v_insere(g,i);
arv_v_insere(g,h);
arv_v_insere(a,g);
arv_v_insere(a,f);
arv_v_insere(a,b);
```



Exemplo de hierarquia da árvore



Representação esquemática da árvore com as “listas de filhos”