



EXERCÍCIO 8 – ÁRVORE BINÁRIA

- **Prazo para entrega: 12/06/2013 – 23:55:00**

- **Atenção:**

1. **Arquivo:** o nome do arquivo referente ao código-fonte deverá respeitar o seguinte padrão: <número do RA>_ED1_EX<número do exercício>.c. Exemplo: 123456_ED1_EX08.c;
2. **E/S:** tanto a entrada quanto a saída de dados devem ser “secas”, ou seja, não devem apresentar frases explicativas. Siga o modelo fornecido e apenas complete as partes informadas.
3. **Identificadores de variáveis:** escolha nomes apropriados;
4. **Documentação:** inclua comentários e indentação no programa.

- **Descrição**

Considere uma árvore binária em que cada nó armazenará um valor inteiro (`int`). Sua tarefa é aplicar os conceitos de árvores binárias e implementar funções simples de manipulação, tais como inserção e percursos.

O programa disponibiliza um menu principal composto por seis opções:

- (1) **Modo inserção:** habilita o modo de inserção e oferece um sub-menu. Um apontador auxiliar de inserções é utilizado, sempre começando apontado para a raiz da árvore:
 - (1.1) **Inserir à esquerda:** inserir um nó-filho à esquerda do nó apontado pelo auxiliar;
 - (1.2) **Inserir à direita:** inserir um nó-filho à direita do nó apontado pelo auxiliar;
 - (1.3) **Mover para esquerda:** mover o auxiliar para o seu nó-filho esquerdo atual;
 - (1.4) **Mover para a direita:** mover o auxiliar para o seu nó-filho direito atual;
 - (1.5) **Mover para a raiz:** mover o auxiliar para apontar para o nó raiz;
 - (1.6) **Sair do modo inserção:** finalizar o modo inserção e retorna ao menu principal.
- (2) **Percurso:**
 - (2.1) **Pré-ordem:** imprime o percurso pré-ordem na árvore;
 - (2.2) **Em ordem:** imprime o percurso em ordem na árvore;
 - (2.3) **Pós-ordem:** imprime o percurso pós-ordem na árvore;

- (3) **Representação aninhada:** imprime a representação aninhada da árvore;
- (4) **Representação indentada:** imprime a representação indentada da árvore;
- (5) **Informações:** imprime a altura, o número de nós, a soma de todos os elementos, o maior e o menor elemento da árvore;
- (6) **Sair:** libera a memória alocada e encerra o programa.

Complete o arquivo ex08.c

Você deve apenas completar as operações nos lugares indicados e não realizar nenhuma alteração na função *main*. Você precisará utilizar comandos de saída de dados (**printf**) nas operações de percurso e representação da árvore. Nesse sentido, leia atentamente as seguintes recomendações abaixo para que não ocorram problemas com o *OnlineJudge*:

- Nas operações de percurso (pré-ordem, em ordem ou pós-ordem), imprima o elemento do nó utilizando a máscara de inteiros (**%d**), seguido de um único espaço. Não é preciso quebrar linha!
- Na representação aninhada, para cada nó, você deve imprimir abre chaves e o elemento, utilizando máscara de inteiros (**{%d}**). Caso o elemento possua filhos, estes devem ser separados por vírgula. Por fim, imprima o fecha chaves, sem espaços ou quebras de linha!
- Na representação indentada, você deve utilizar uma tabulação horizontal (**\t**) para cada nível de profundidade do nó. O elemento deve então ser impresso com máscara de inteiros, seguido de uma única quebra de linha (**%d\n**).

Complete as seguintes funções:

- **criaRaiz:** cria o nó raiz da árvore.
- **insereEsq:** insere um novo nó como filho esquerdo do nó enviado como argumento.
- **insereDir:** insere um novo nó como filho direito do nó enviado como argumento.
- **preorder:** realiza o percurso pré-ordem na árvore.
- **inorder:** realiza o percurso em ordem na árvore.
- **postorder:** realiza o percurso pós-ordem na árvore.
- **libera:** libera a memória alocada de todos os nós da árvore.
- **altura:** retorna a altura da árvore.
- **numeroNos:** retorna o número de nós da árvore.
- **soma:** retorna a soma de todos os elementos da árvore.
- **maiorElemento:** retorna o valor do maior elemento da árvore.
- **menorElemento:** retorna o valor do menor elemento da árvore.
- **representacaoAninhada:** imprime a representação aninhada da árvore.

- **representacaoIndentada**: imprime a representação indentada da árvore.

Exemplo de E/S (os comentários entre parênteses não deverão ser exibidos):

Entrada	Saída
10 (raiz da árvore)	
1 (modo inserção)	
1 5 (inserir 5 à esquerda da raiz)	
3 (mover para o filho à esquerda: 5)	
1 2 (inserir 2 à esquerda de 5)	
2 4 (inserir 4 à direita de 5)	
5 (retornar para a raiz)	
2 80 (inserir 80 à direita da raiz)	
4 (mover para o filho à direita: 80)	
1 19 (inserir 19 à esquerda de 80)	
2 30 (inserir 30 à direita de 80)	
4 (mover para o filho à direita: 30)	
1 45 (inserir 45 à esquerda de 30)	
2 18 (inserir 18 à direita de 30)	
6 (finalizar modo inserção)	
2 1 (percurso pré-ordem)	10 5 2 4 80 19 30 45 18
2 2 (percurso em ordem)	2 5 4 10 19 80 45 30 18
2 3 (percurso pós-ordem)	2 4 5 19 45 18 30 80 10
3 (repr. aninhada)	{10,{5,{2},{4}}},{80,{19},{30,{45},{18}}}}
4 (repr. indentada)	10 <div> 5 <div> 2 <div>4</div> </div> </div> 80 <div> 19 <div>30 <div>45</div> </div> </div> 18
5 (informações)	4, 9 (altura e quantidade de nós)
	213, 80, 2 (soma dos nós, maior e menor valor)
6 (sair)	

- **Casos de testes:**

1. **Caso 1:** avalia opções 1 (1.1, 1.2, 1.3, 1.4, 1.5, 1.6), 2 (2.1, 2.2, 2.3) e 6;
2. **Caso 2:** opções avaliadas no Caso 1 + opção 5;
3. **Caso 3:** opções avaliadas no Caso 2 + opção 3;
4. **Caso 4:** opções avaliadas no Caso 2 + opção 4;
5. **Caso 5:** avalia todas as opções.

- **Cuidados:**

1. **Erros de compilação:** nota **zero** no exercício
2. **Tentativa de fraude:** nota **zero** para todos os envolvidos.