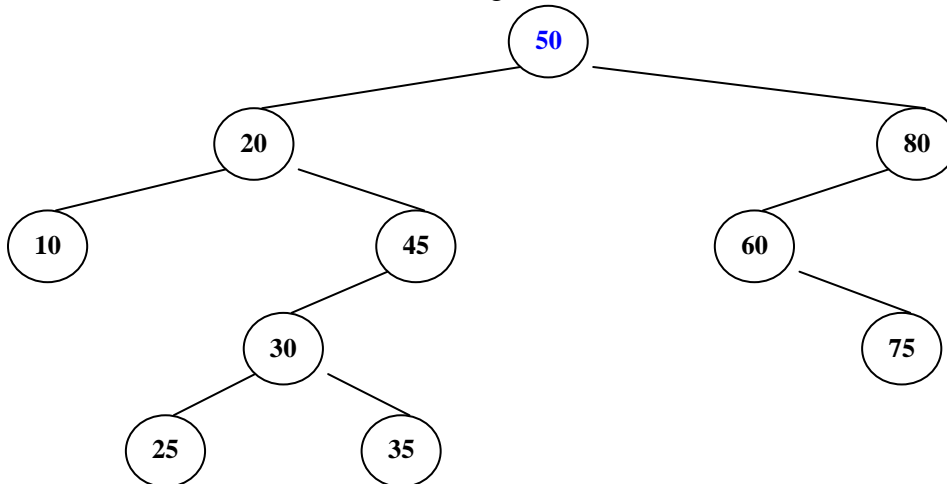


**Review: BINARY SEARCH TREES**

- 1.** Create a binary search tree using the following data entered as a sequential set:  
50, 70, 30, 60, 40, 80, 35.

- 2.** Delete the node 50 from the BST given below.



**Review: BINARY SEARCH TREES**

**3.** Create a binary search tree using the following data entered as a sequential set using the number as a key (not unique!):

{50, 'A'}, {60, 'B'}, {30, 'A'}, {10, 'C'}, {20, 'D'}, {10, 'A'}, { 8, 'G'},  
{12, 'X'}, {40, 'B'}, {11, 'V'}, {15, 'B'}, {10, 'B'}, {10, 'D'}

**4.** Write an iterative algorithm/function named **printSameKey** to print all nodes with the same target key in a BST. The algorithm/function receives the root of the tree and the target key and returns nothing. Your algorithm calls the **searchBST** algorithm/function: it receives the root of the tree and the target key and it returns the address of the node containing the target in case of success or NULL otherwise.

Example: For the binary search tree created in the previous exercise and the target key 10, the output should be: C, A, B, D.