```
#include <stdio.h>
#include <stdlib.h>

/* A binary tree node has data, pointer to left child
   and a pointer to right child */
struct Node
{
   int data;
   struct Node* left;
   struct Node* right;
};

void printnodes_KDist(struct Node *root , int k)
{
   if(root == NULL)
      return;
   if( k == 0 )
   {
      printf( "%d ", root->data );
      return ;
   }
   else
   {
      printnodes_KDist( root->left, k-1 ) ;
      printnodes_KDist( root->right, k-1 ) ;
   }
}

struct Node* newNode(int data)
{
  struct Node* node = (struct Node*)
                      malloc(sizeof(struct Node));
  node->data = data;
  node->left = NULL;
  node->right = NULL;

  return(node);
}

/* Driver program to test above functions*/
int main()
{

  struct Node *root = newNode(1);
  root->left        = newNode(2);
  root->right       = newNode(3);
  root->left->left  = newNode(4);
  root->left->left->left  = newNode(9);
```

```
  root->left->right = newNode(5);
  root->left->right->right = newNode(10);
  root->right->left = newNode(8);

  printnodes_KDist(root, 3);


  return 0;
}
```

Output: 9 10

Time complexity: O(n)
Space complexity: O(n)