Question 8:

struct AVLTree\* newAVLTree();

void addAVLTree(struct AVLTree \*tree, TYPE val);

void treeSort (TYPE data[], int n)

{

struct AVLTree\* treeSortAVL = newAVLTree();

for(int I = 0; I < n; i++){

addAVLTree(treeSortAVL, data[i]);

}

int count = 0;

\_treeSortHelper(treeSortAVL->root, data, &count);

}

void \_treeSortHelper(AVLNode \*cur, TYPE \*data, int \*count)

{

if(cur != NULL){

\_treeSortHelper(cur->left, data, count)

data[\*count] = cur->val;

(\*count)++;

\_treeSortHelper(cur->right, data, count);

}

}

Question 22:

struct hlink {  
TYPE value;  
struct hlink \*next;  
};  
struct hashTable {  
struct hlink \*\* table;  
int tablesize;  
int count;  
};

int hashTableContains (struct hashTable \* ht, TYPE testElement) {

int idx = HASH(testElement) % count;

if (idx < 0){

idx+= count;

}

struct hlink\* theOne = ht->table[idx]

while(theOne != NULL){

if(theOne->value = testElement){

return 1;

}

theOne = theOne->next;

}

return 0;  
}