Aim: Training on binary trees.

Think about a simple “English-to-Turkish Dictionary” project in which a dictionary is to be implemented as a “binary search tree”. To be used in this project, the sample implementations of the binary search tree routines may be found at http://cslibrary.stanford.edu/110/BinaryTrees.html. The Dictionary, which is the binary search tree data structure in your project, will contain the nodes which include the word information as declared below:

```c
struct Entry {
    char* EnglishVersion; // the pointer to the English word
    char* TurkishVersion; // the pointer to the Turkish translation of the related English word
};
typedef struct Entry ENTRY;

struct Node {
    ENTRY word;
    struct Node* left; // left child of the binary tree
    struct Node* right; // right child of the binary tree
};
typedef struct Node NODE;
```

Implement the “insertNode” and “translateFromEnglishToTurkish” routines for the binary search tree of the project, and also implement a sample main function to test the project.

```c
/* sample main may be as follows: */
int main() {
    NODE * dictionaryRoot=NULL; // The Dictionary includes no words at the beginning.
    ENTRY temporaryWord; char tempWord[50]; char* resultTest; int i;
    for (i=0;i<3;i++) {
        printf("Give the English version of the word:");
        scanf("%s", tempWord);
        temporaryWord.EnglishVersion=(char*)malloc(sizeof(strlen(tempWord)+1)*sizeof(char));
        strcpy(temporaryWord.EnglishVersion,tempWord);

        printf("Give the Turkish translation of the word:");
        scanf("%s", tempWord);
        temporaryWord.TurkishVersion=(char*)malloc(sizeof(strlen(tempWord)+1)*sizeof(char));
        strcpy(temporaryWord.TurkishVersion,tempWord);
        dictionaryRoot=insert(dictionaryRoot, temporaryWord);
    }
    resultTest=translateFromEnglishToTurkish(dictionaryRoot,"programming");
    if(resultTest!=NULL)
        printf("The English translation of "programming" is %s\n", resultTest);
    else
        printf("The word is not available in the dictionary\n");
    return 0;
}
```