

```

//Khai bao Nut
struct Tnode{
    int next[K];
    int p;
    int stop;
    // Mot so thong tin bo sung
    int cnt;
};

// Cay Trie
Tnode Trie[maxT];
int nT=0;

// Ham them 1 tu nguoc
void AddWord(char* s) {
    int sz=strlen(s);
    int v=0;
    for(int i=0;s[i];--i) {
        int k=s[i]-'a';
        if (Trie[v].next[k]==0) {
            ++nT;
            Trie[v].next[k]=nT;
            Trie[nT].p=v;
        }
        v=Trie[v].next[k];
    }
    ++Trie[v].stop;
    while (v) ++Trie[v].cnt, v=Trie[v].p;
    ++Trie[0].cnt;
}

// Ham dem so tu bang s
int CountWord(char* s) {
    int v=0;
    for(int i=0; s[i];++i) {
        int k=s[i]-'a';
        if (Trie[v].next[k]==0) return 0;
        v=Trie[v].next[k];
    }
    return Trie[v].stop;
}

```

```
}
```

```
// Ham dem tien to bang s
```

```
int CountPrefix(char* s) {  
    int v=0;  
    for(int i=0;s[i];++i) {  
        int k=s[i]-'a';  
        if (Trie[v].next[k]==0) return 0;  
        v=Trie[v].next[k];  
    }  
    return Trie[v].cnt;  
}
```

```
// Ham dem so xau <=s
```

```
int CountLE(char* s) {  
    int v=0, res=0;  
    for(int i=0; s[i];++i) {  
        int k=s[i]-'a';  
        for(int l=0;l<k;++l)  
            res += Trie[Trie[v].next[l]].cnt;  
        if (Trie[v].next[k]==0) return res;  
    }  
    return res+Trie[v].stop;  
}
```

```
// Ham dem so xau >s
```

```
int CountG(char* s) {  
    int v=0, res=0;  
    for(int i=0; s[i]; ++i) {  
        int k=s[i]-'a';  
        for(int l=k+1;l<K;++l)  
            res += Trie[Trie[v].next[l]].cnt;  
    }  
    return res; // return res+Trie[v].cnt neu >=  
}
```