```
//[反片語/Ananagrams](1/2)
#define IN "P16IN.txt"
#define OUT "P16OUT.txt"
//***************************************
#include <iostream>
#include <ctime>
using namespace std;
void redir(void);
//***************************************
/* Work Space*/
#include <string>
#include <vector>
#include <algorithm>
#include <map>

string normalize(string &s);
//***************************************
int main(void)
{
    redir(); //redirection
//***************************************
/* Work Space*/
    string s, n;
    vector<string> words, ans;
    map<string, int> cnt;
    int i;

    while(cin >> s){
        if(s[0] == '#'){
            break;
        }
        words.push_back(s);
        n = normalize(s); //'=' is overloaded

        if(!cnt.count(n)){
            cnt[n] = 0;
        }
        cnt[n]++;
    }

    for(i=0; i<words.size(); i++){
        if(cnt[normalize(words[i])] == 1){
            ans.push_back(words[i]);
        }
    }
    sort(ans.begin(), ans.end());

    for(i=0; i<ans.size(); i++){
        cout << ans[i] << endl;
    }
//***************************************
    freopen("CON", "r", stdin); //取消重新導向
    freopen("CON", "w", stdout);

    printf("Time used = %.2f\n", (double)clock()/CLK_TCK); //傳回程式目前為止執行的時間

    system("pause");
    return 0; //the end...
}
```

```
//[反片語/Ananagrams](2/2)
void redir(void)
{
    freopen(IN, "r", stdin);
    freopen(OUT, "w", stdout);
}
//***************************************
/* Work Space*/
//把每個單字標準化(轉換為小寫字母再按進行排序)
string normalize(string &s)
{
    string ans = s; //'=' is overloaded
    int i;

    for(i=0; i<ans.length(); i++){
        ans[i] = tolower(ans[i]);
    }

    sort(ans.begin(), ans.end());
    return ans;
}

//Input(IN) Sample
/*
ladder came tape soon leader acme RIDE lone Dreis peat
 ScAlE orb eye Rides dealer NotE derail LaCes drIed
noel dire Disk amce Rob Dries
#
*/

//Output(OUT)
/*
Disk
NotE
derail
drIed
eye
ladder
soon
*/
```