```
#include <iostream>
#include <iomanip>
#include <string>
#include <fstream>

void readToSentinel (std::string words[]);
void extractAllTerms (std::string words[], int used, std::string key_words[], int& terms2);
int search (std::string words[], int terms1, std::string key_words[], int terms2);
void count (std::string words[], int used, int counter[], int terms2, std::string key_words[]);
void over (const int counter[], const std::string key_words[], const int terms2);
void maxpos (int counter[], int i, std::string key_words[]);

using namespace std;

ifstream text ("test.txt");
const int SIZE = 500;

int main()
{
    ifstream text;
    string words[SIZE];

    readToSentinel (words);

    return 0;
}

void readToSentinel (string words[])
{
    int used = 0, counter[SIZE], terms2;
    string key_words[SIZE];
    text >> words[used];
    while (words[used] != "XXX" && used < SIZE)
    {
        used++;
        text >> words[used];
    }
    if (used == SIZE)
    {
        cout << "TOO MANY TERMS, ARRAY FULL!!" << endl;
    }
    else
    {
        extractAllTerms (words, used, key_words, terms2);
        count (words, used, counter, terms2, key_words);
    }
}

void extractAllTerms (string words[], int used, string key_words[], int& terms2)
{
    int i, terms1;
    terms2 = 0;

    for (terms1 = 0; terms1 < used; terms1++)
    {
        if (words[terms1] != key_words[terms2])
        {
            i = search (words, terms1, key_words, terms2);
            if (i < 0)
            {
                key_words[terms2] = words[terms1];
                terms2++;
            }
        }
    }
}
```

```
int search (string words[], int terms1, string key_words[], int terms2)
{
    int i = 0;
    while (i < terms2 && key_words[i] != words[terms1])
        i++;

    if (i == terms2)
        return -1;
    else
        return 1;
}

void count (string words[], int used, int counter[], int terms2, string key_words[])
{
    int k = 0;
    for (int j = 0; j < terms2; j++)
    {
        k = 0;
        while (k < used)
        {
            if (key_words[j] == words[k])
                counter[j]++;
            k++;
        }
    }
    over (counter, key_words, terms2);
}

void over (const int counter[], const string key_words[], const int terms2)
{
    string terms[terms2];
    int termst[terms2], j = 0;
    for (int i = 0; i < terms2; i++)
    {
        if (counter[i] > 5)
            {
                termst[j] = counter[i];
                terms[j] = key_words[i];
                j++;
            }
    }

    for (int i = 0; i < j; i++)
    {
        cout << terms[i] << " " << termst[i] << endl;
    }
    maxpos (termst, j, terms);
}

void maxpos (int counter[], int i, string key_words[])
{
    int maxi = counter[0];
    string maxw;
    for (int j = 0; j < i; j++)
    {
        if (counter[j] >= maxi)
        {
            maxi = counter[j];
            maxw = key_words[j];
        }
    }
    cout << "\nThe term that appeared the most was: ";
    cout << maxw << ", it appeared " << maxi << " times" << endl;
}
```