



给定一个字符串 **s**，你可以从中删除一些字符，使得剩下的串是一个回文串。如何删除才能使得回文串最长呢？

输出需要删除的字符个数。

```
1 import java.util.*;
2 public class Main{
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         while(sc.hasNext()) {
6             String str =sc.nextLine();
7             char[] strchar = str.toCharArray();
8             int length= strchar.length;
9             int[][] dp = new int[length][length];
10            for(int j=1;j<length;j++){
11                dp[j-1][j]=strchar[j-1]==strchar[j]?0:1;
12                for(int i=j-2;i>-1;i--){
13                    if(strchar[i]==strchar[j]){
14                        dp[i][j]=dp[i+1][j-1];
15                    }else{
16                        dp[i][j]=Math.min(dp[i+1][j],dp[i][j-1])+1;
17                    }
18                }
19            }
20        }
21    }
22}
```

小 Q 最近遇到了一个难题：把一个字符串的大写字母放到字符串的后面，各个字符的相对位置不变，且不能申请额外的空间。

你能帮帮小 Q 吗？

不能构建额外空间，那么交换移动元素使用位操作的那个版本 `swap()`

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4
5  bool isCap(char c)
6  {
7      if (c >= 'A' && c <= 'Z')
8          return true;
9      else
10         return false;
11 }
```





```
12
13 void mSwap(char &a, char &b)
14 {
15     if (a != b)
16     {
17         a ^= b;
18         b ^= a;
19         a ^= b;
20     }
21 }
22
23 int main()
24 {
25     string s;
26     while (cin >> s)
27     {
28         int len = s.size();
29         int end = len;
30         for (int i = 0; i < end; ++i)
31         {
32             if (isCap(s[i]))
33             {
34                 int j = i;
35                 for (; j < len - 1; ++j)
36                     mSwap(s[j], s[j + 1]);
37                 --end;
38                 --i;
39             }
40         }
41         cout << s << endl;
42     }
43     return 0;
44 }
```

小 Q 今天在上厕所时想到了这个问题：有  $n$  个数，两两组成二元组，差的绝对值最小的有多少对呢？差的绝对值最大的呢？

```
1  #include <iostream>
2  #include <map>
3  #include <utility>
4  using namespace std;
5  // 用一个 map 来存储输入的数，当存在相同的数时不插入新的数，而是将计数值+1
6  int main()
7  {
```

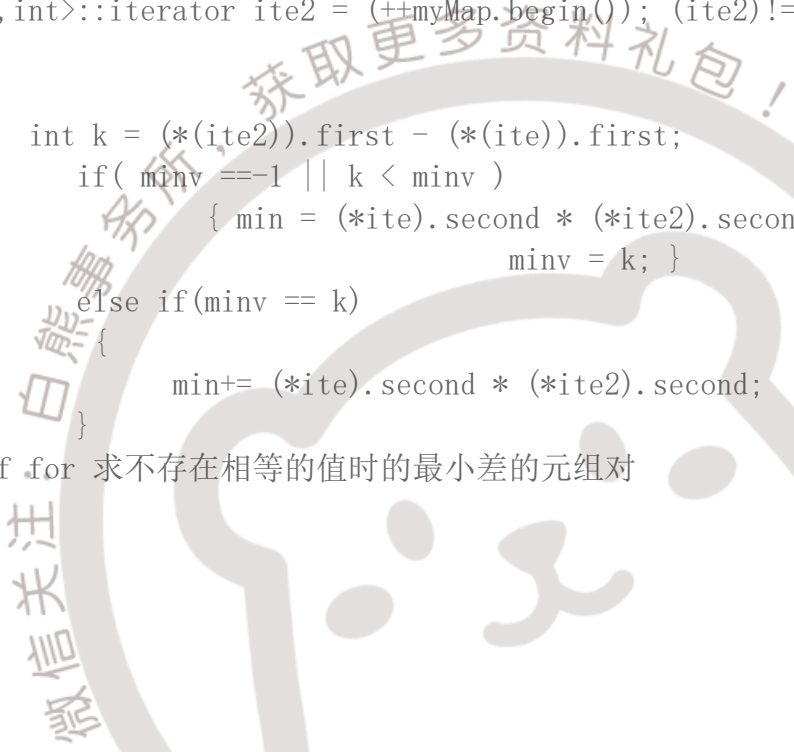
获取更多资料礼包！

微信关注：白熊事务所



```
8         int num;
9         while(cin>>num)
10        {
11            map<int,int> myMap;
12            bool flag = false;
13            for(int i = 0; i < num ; i++)
14            {
15                int k ;
16                cin>>k;
17                map<int,int>::iterator ite;
18                ite = myMap.find(k);
19                if(ite != myMap.end())
20                {
21                    (*ite).second++;flag = true;}
22                else
23                {
24                    myMap.insert(make_pair(k,1));
25                }
26            } // end of for 读取输入的数据
27            map<int,int>::iterator ite = myMap.begin();
28            int min =0;
29            int minv = -1;
30            if(flag) //如果存在相同的数
31            {
32                for( ; ite!= myMap.end(); ite++)
33                {
34                    if((*ite).second > 1)
35                        min += ((*ite).second * ((*ite).second -1 ))/2;
36                } //最小差元组对数等于所有相等的数构成的元组对
37            }
38            else
39            {
40                for( map<int,int>::iterator ite2 = (++myMap.begin()); (ite2)!=
41                    ite2++,ite++ )
42                {
43                    int k = ((*ite2)).first - ((*ite)).first;
44                    if( minv ==-1 || k < minv )
45                        { min = (*ite).second * (*ite2).second;
46                          minv = k; }
47                    else if(minv == k)
48                    {
49                        min+= (*ite).second * (*ite2).second;
50                    }
51                } // end of for 求不存在相等的值时的最小差的元组对
52            }
53        }
```

数





```
52         } // 最小对的个数
53         int max = (*myMap.rbegin()).second * (*myMap.begin()).second;
54     //最大差的对数
55     cout<< min<<" "<<max<<endl;
56
    }
}
```

