```
/* URAL - 1635 -  Mnemonics and Palindromes
A palindrome partition is the partitioning of a string such that each separate substring is a palindrome.
For example, the string "ABACABA" could be partitioned in several differentways, such as
{"A","B","A","C","A","B","A"}, {"A","BACAB","A"}, {"ABA","C","ABA"}, or {"ABACABA"}, among others.
Given a string s of uppercase letters with length no more than 4000.
In the first line Print the minimum possible number of substrings in a palindrome partition of s. and
In the second line output palindromes from the optimal decomposition separated by a space.
If several answers are possible, output any of them.
*/

#include<bits/stdc++.h>
using namespace std;
string s;
int n,dp[4005],ispalindrome[4005][4005];
vector< pair<int,int> >vv;

int palindrome(int i,int j){
    if(i>=j)return ispalindrome[i][j]=1;
    if(ispalindrome[i][j]!=-1)return ispalindrome[i][j];
    int ret = 0;
    if(s[i]==s[j])ret = palindrome(i+1,j-1);
    return ispalindrome[i][j] = ret;
}

int fun(int i){
    if(i==n)return 0;
    if(dp[i]!=-1)return dp[i];
    int ret = 1000000;
    for(int j=i; j<n; j++){
        if(palindrome(i,j)==1){
            ret = min(ret,1+fun(j+1));
        }
    }
    return dp[i] = ret;
}

void path(int i){
    if(i==n)return;
    int ret = fun(i);
    for(int j=i; j<n; j++){
        if(palindrome(i,j)==1){
            int cnt = 1 + fun(j+1);
            if(cnt==ret){
                vv.push_back(make_pair(i,j));
                path(j+1);
                break;
            }
        }
    }
}
```

```
51.
52. int main(){
53.     ios::sync_with_stdio(false); cin.tie(0); cout.tie(0);
54.     cin>>s; n = s.size();
55.     memset(ispalindrome,-1,sizeof(ispalindrome));
56.     memset(dp,-1,sizeof(dp));
57.     int ans = fun(0);
58.     cout<<ans<<endl;
59.     path(0);
60.     for(int i=0; i<vv.size(); i++){
61.         int Start = vv[i].first;
62.         int End = vv[i].second;
63.         for(int j=Start; j<=End; j++){
64.             cout << s[j];
65.         }
66.         if(i==(int)vv.size()-1)cout << endl;
67.         else cout << " ";
68.     }
69.     return 0;
70. }
```