LCS – Making Palindrome:

In this problem I have given a string .I have to make it palindrome as minimum (insert/change) possible:

Idea: I have reverse the string .and compared this string with the given string using LCS:

CODE:

```
#include<bits/stdc++.h>
using namespace std;
int dp[1005][1005];
string a,b;
int ln1,ln2;
int mxx = 1000000;
int cn;
int lcs(int i,int j)
{
   if(i == ln1)return (ln2-j);
   if(j == ln2)return (ln1-i);

   if(dp[i][j]!=-1)
      return dp[i][j];

   int ret1 = mxx,ret2 = mxx,ret3 = mxx,ret4 = mxx;

   if(a[i] == b[j])
      ret1 =  lcs(i+1,j+1);
   else
   {
      ret4 = 1 + lcs(i+1,j);///changed
      ret3 = 1 + lcs(i,j+1); ///insert;
   }
   return dp[i][j] = min(ret4,min(ret1,min(ret2, ret3)));
}
void path(int i, int j)
{
    if(i == ln1 || j == ln2)
    {
       if(i == ln1)
       {
          for(int k=j;k<ln2;k++)
            cout<<b[k];
          return;
       }
       else
       {
          for(int k=i;k<ln1;k++)
            cout<<a[k];
          return;
       }
    }

if(a[i] == b[j])
{
   cout<<a[i];
   path(i+1,j+1);
}
else
{
   int  ret4 = 1 + lcs(i+1,j);
   int ret3 = 1 + lcs(i,j+1);
   if(ret4<ret3)
   {
      cout<<a[i];
      path(i+1,j);
   }
   else
   {
    cout<<b[j];
      path(i,j + 1);
```

```
        }
    }
}
int main()
{
    //freopen("out.txt","wt",stdout);
    int t;
    while(cin>>a)
    {
        b = a;
        reverse(b.begin(),b.end());

        ln1 = a.size(),ln2 = b.size();
        memset(dp,-1,sizeof(dp));
        int x = lcs(0,0);
        cout<<x/2<<" ";
        path(0,0);
        cout<<endl;
    }
}
```