**SOLUTION**

class Solution {

public:

Solution(){

ios::sync\_with\_stdio(false);

std::cin.tie(nullptr);

std::cout.tie(nullptr);

}

int minDistance(string word1, string word2) {

int m = word1.length();

int n = word2.length();

int dp[m+1][n+1];

for(int i = 0; i <= m; i++){

for(int j = 0; j <= n; j++){

if(i == 0)

dp[i][j] = j;

else if(j == 0)

dp[i][j] = i;

else{

if(word1[i-1] == word2[j-1])

dp[i][j] = dp[i-1][j-1];

else

dp[i][j] = 1 + min({dp[i-1][j-1],dp[i-1][j],dp[i][j-1]});

}

}

}

return dp[m][n];

}

};

**TIME COMPLEXITY= O(M\*N)**

**SPACE COMPLEXITY= O(M\*N)**