**动态规划算法：给定两个字符串str1，str2，在给定三个整数ic,dc,rc，分别代表插入，删除和替换一个字符的代价。返回将str1编辑成str2的代价，比如， str1="abc",str2="adc",ic=5,dc=3,rc=2,从str1到str2，将'b'换成'd'代价最小，所以返回2.**

**实验代码：**

#include <iostream>

#include <string>

#include <algorithm>

#include <vector>

using namespace std;

int main(){

string str1 = "ab12cd3";

string str2 = "abcdf";

//cin>>str1;

//cin>>str2;

const int M = str1.length();

const int N = str2.length();

//vector<int> p(M+1,0);

//vector<vector<int>> dp(N+1,p);

int dp[10][10] = {};

int ic=5,dc=3,rc=2;

//int ic = 1,dc=1,rc=1;

dp[0][0] = 0;

for (int i = 1;i<N+1;i++)

dp[0][i] = ic\*i;

for (int i = 1;i<M+1;i++)

dp[i][0] = dc\*i;

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

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

int x = min(dc+dp[i+1][j],dp[i][j+1]+ic);

if (str1[i]!=str2[j])

dp[i+1][j+1] = min(dp[i][j] + rc,x);

else

dp[i+1][j+1] = min(dp[i][j],x);

}

}

cout << dp[M][N] << endl;

}