**★☆☆☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：11058: Encoding**

**解題者：游騰岳**

**解題日期：2024年3月6日**

**題意：**

對一個字符串進行編碼，根據一組初始的編碼策略和額外的規則，對原始字符串中的字母進行替換。初始的編碼策略指定了如何替換原始字符串中的每個字母，而額外的規則則根據位置進行替換。初始編碼策略:輸入26行，指定每個字母的初始編碼策略。接下來是一個整數R，表額外規則的數量，然後是R行，每行包含一個規則PXY，表示自位置P開始，將字母X替換為Y。

如果對於同一個字母在同一個位置有多個規則，則執行最後出現的規則。

**題意範例：**

Sample Input:

ufrn

t

o

w

k

q

z

f

n

y

i

c

m

s

j

n

r

g

l

d

s

u

s

g

y

e

u

10

0 q t

0 j f

1 v d

1 f d

1 r o

2 e p

2 v e

3 y p

3 t m

3 u k

Sample Output:

Case #1: The encoding string is udoj.

**解法：**

我用一個Class儲存所有額外的規則

重最後一個規則開始執行

設立array將執行過的位置記起來!

避免重複套用規則

**解法範例：**

無。

**討論：**

無。

**程式：**

#include <iostream>

#include <string>

using namespace std;

class Rule{

 public:

 Rule();

 void set(int p,char x,char y);

 int get\_P();

 char get\_X();

 char get\_Y();

 private:

 int p;

 char x,y;

};

Rule::Rule(){}

void Rule::set(int P,char X,char Y){

 p=P;

 x=X;

 y=Y;

}

int Rule::get\_P(){return p;}

char Rule::get\_X(){return x;}

char Rule::get\_Y(){return y;}

int main() {

 string S;

 char re[26],X,Y;

 int length,temp,extra,P,rul[100],caseNumber=1;

 while (cin >> S) {

 for(int i=0;i<100;i++)

 rul[i]=0;

 length = S.size();

 for(int i=0;i<26;i++)

 cin >> re[i];

 cin >> extra;

 Rule s1[1000];

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

 cin >> P >> X >> Y;

 s1[i].set(P,X,Y);

 }

 for(int i=extra-1;i>=0;i--){

 for(int j=s1[i].get\_P();j<length;j++){

 if(S[j] == s1[i].get\_X() && rul[j]==0){

 S[j] = s1[i].get\_Y();

 rul[j]=1;

 }

 }

 }

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

 temp = S[i]-'a';

 if(rul[i]==0)

 S[i] = re[temp];

 }

 cout << "Case #"<< caseNumber << ": The encoding string is ";

 cout << S <<"."<<endl<<endl;

 caseNumber++;

 }

 return 0;

}