화합물

새로운 혼합 화합물을 만들었는데 이 때 기존의 혼합화합물들과 비교하여 가장 유사도가 높은 혼합 화합물을 찾고 싶다.

혼합 화합물은 항상 5개의 화학식으로 구성되어 있고, 각 화학식은 항상 10개 이하의 원자로 구성 되어 있다.

원자의 종류는 알파벳 소문자 [a~z]만 존재한다.

예를 들어 하나의 화학식은 이런 형태로 구성된다.

a원소 3개 + b원소 2개 = "aaabb"

c원소 3개 + z원소 3개 = "cczczz"

혼합 화합물을 서로 비교할 때는 같은 위치의 화학식끼리 비교한다

화합물1	화합물2	
화학식1 = "aaa"	화학식1 = "aaa"	화학식1 유사도 100점
화학식2 = "bbb"	화학식2 = "bbc"	화학식2 유사도 66점
화학식3 = "ccc"	화학식3 = "ffffff"	화학식3 유사도 0점
화학식4 = "ddd"	화학식4 = "rrere"	화학식4 유사도 0점
화학식5 = "eee"	화학식5 = "bfdgrre"	화학식5 유사도 20점
		총 유사도 186점

혼합 화합물끼리 유사도를 비교할 때 유사도가 100점인 화학식이 최소한 하나도 존재하지않으면 유사도는 0점으로 생각한다

소스코드 설명

---

struct Info {

char a[11], b[11], c[11], d[11], e[11];

};

Info는 혼합 화합물 정보를 저장하는 구조체이다

Int getScore(char *c1,char *c2)

화학식 2개를 넣으면 서로 비교하여 유사도 점수를 반환한다

작성해야 할 함수

Void init()

테스트 케이스마다 초기화를 수행한다

Void ourInfo(Info info)

기존의 혼합 화합물 정도가 매개변수로 주어진다

Int getNewInfoScore(Info info)

새로운 혼합 화합물의 정보가 매개변수로 주어지고, 기존의 혼합 화합물들과 비교해서 **혼합화합물의 유사도 점수가 가장 높을 때의 점수를 반환한다.**

화학식1, 화학식2, … 각 화학식의 가능한 개수는 500개 이하이다

ourInfo , getNewInfoScore 호출횟수는 각각 5000 번 이하이다

example

입력

Test case 개수,

기존의 혼합 화합물 개수 N,

새로운 혼합 화합물 개수 M,

기존의 혼합 화합물 N 개

새로운 혼합 화합물 M 개

2

1 1

aaa

bbb

ccc

ddd

eee

aaa

bbc

ffffff

rrere

bfdgrre

2 2

aa

bb

cc

dd

---

ee

aa

ff

gg

xx

dd

bb

cc

dd

ee

aa

xx

yy

gg

xzqwer

dasdqwe

출력

Total Score : 186

Total Score : 169

소스코드

```
#include <cstdio>

struct Info{
	char a[11], b[11], c[11], d[11], e[11];
};

int _strlen(char *c){
    int len = 0;
    while(*c != '₩0')    len++, c++;
    return len;
}
int _max(int a, int b){return (a > b ? a : b);}
int getScore(char *c1, char *c2){
	int len1 = _strlen(c1);
	int len2 = _strlen(c2);
	int dp[11][11] = {0, };
	for(int i = 0; i < len1; i++){
		for(int j = 0; j < len2; j++){
			if(c1[i] == c2[j])   dp[i + 1][j + 1] = dp[i][j] + 1;
			else dp[i + 1][j + 1] = _max(dp[i + 1][j], dp[i][j + 1]);
		}
	}
	return 200 * dp[len1][len2] / (len1 + len2);
}

/* User Define */
extern void init(){}
extern void ourInfo(Info info){}
extern int getNewInfoScore(Info info){}

int TestCase, ourInfoCount, newInfoCount;
void inputOurCharacter(){
	for(int i = 0; i < ourInfoCount; i++){
		Info oInfo;
		scanf("%s %s %s %s %s", oInfo.a, oInfo.b, oInfo.c, oInfo.d, oInfo.e);
		ourInfo(oInfo);
	}
```

```
}

int proc(){
	int answer = 0;
	for(int i = 0; i < newInfoCount; i++){
		Info nInfo;
		scanf("%s %s %s %s %s", nInfo.a, nInfo.b, nInfo.c, nInfo.d, nInfo.e);
		answer += getNewInfoScore(nInfo);
	}
	return answer;
}

int main(){
	scanf("%d", &TestCase);
	while(TestCase--){
		init();
		scanf("%d %d", &ourInfoCount, &newInfoCount);
		inputOurCharacter();
		printf("Total Score : %d₩n", proc());
	}
	return 0;
}
```