|  |  |  |
| --- | --- | --- |
|  | |  | | --- | | 문제: **(중) [사전A-0031] [2019년 10~11월 PRO 검정 사전 테스트] 자율주행 테스트** | |

**※ 본 테스트는 Professional 사전테스트로   
   문제풀이 과정에서 본인의 힘으로 문제를 풀이하지 않고 타인의 조언을 구하거나             
   본인이 직접 작성하지 않은 자료를 그대로, 혹은 위조, 변조, 참조하여 제출하는 경우 부정행위로   
   판단될 수 있으며, 이러한 경우 6개월간 모든 SW검정 응시가 제한됨을 알려드립니다.   
   (이미 Professional 이상을 취득한 자격자 또한 본 규정은 동일하게 적용됩니다.)   
  
※ 본 문제는 2019년 9월 24일(화) 13시까지(서버시간 기준)만 풀이가능하며,   
   위 기한 내에 문제의 정답을 제출해야 2019년 10~11월 Professional SW검정을 신청할 수 있습니다.   
  
※ 마지막 제출 답안 결과가 최종결과로 반영되니, 답안제출에 유의하여 주시기 바랍니다.   
  
※ 9월 24일(화) 13시 이후에도 문제는 계속 오픈되며, 일반 연습문제와 같이 계속 풀이할 수 있습니다.    
   9월 24일(화) 13시 이전에는 정답자이더라도 타 정답자 코드 조회가 불가능하며   
   이후에는 정답자에 한해 타 정답자 코드 조회가 가능합니다.**

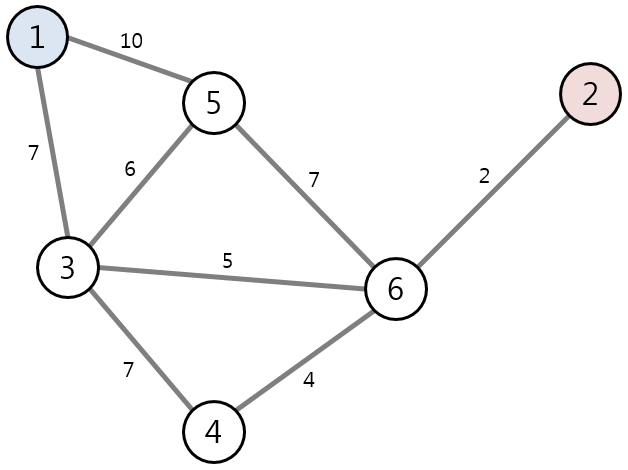
|  |  |
| --- | --- |
| 시간 제한 | **30** 개의 Test Input 입력시 C/C++ **1.5** 초, Java **2.5** 초 |
| 메모리 제한 | Stack : **1** Mbytes  /  Total : **256** Mbytes |
| 코드길이 제한 | **128** Kbytes |
| 채점 | 답안을 제출하면 Test Input에 대한 결과를 판정해서 실시간으로 알려준다.  모든 케이스에 대해 정답일 경우 **Pass**, 그 외의 경우 **Fail** 이다. |

S사에서 자율주행 자동차를 개발하고 있다. 현재 개발된 버전을 테스트하기 위해 도로 주행 테스트를 설계해야 한다.

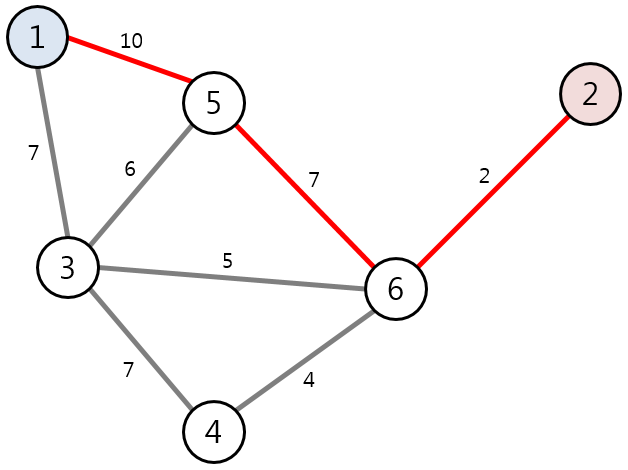
테스트는 N 개의 지점과 M 개의 도로로 구성된 테스트 환경에서 진행되며, 각 도로는 양방향이고, 각 도로를 지나갈 때 유지해야 하는 속도가 정해져 있다.

이번 테스트는 각 도로의 속도에 맞추어 도로를 달리도록 설계하였다.

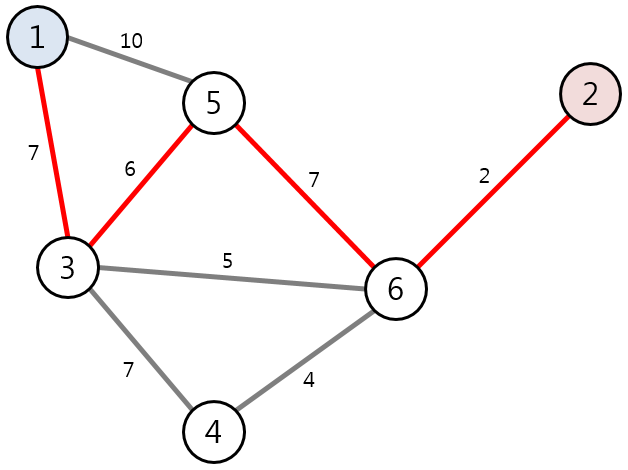
단, 변속기 제어 장치의 성능 문제로 테스트 운행 중 자동차가 달리는 최저속도와 최고속도의 차이가 너무 크면 테스트 진행이 어려울 수 있다.

  
[그림1]

[그림1]은 N=6, M=8, 시작 지점 1, 도착 지점 2 인 테스트 환경이다.

  
[그림2]

테스트 경로를 [그림2]와 같이, 1 -> 5 -> 6 -> 2 로 설계한다면, 경로 중 최저속도는 2 이고, 최고속도는 10 이므로 최저속도와 최고속도의 차이는 8 이다.

  
[그림3]

하지만 테스트 경로를 [그림3]과 같이 1 -> 3 -> 5 -> 6 -> 2 로 설계한다면, 경로 중 최저속도는 2 이고, 최고속도는 7 이므로 최저속도와 최고속도의 차이는 5 이며, 주어진 테스트 환경에서 찾을 수 있는 최솟값이다.

테스트 설계를 위해 테스트 환경의 정보를 입력 받아, 시작 지점에서 도착 지점까지 경로 중 최저속도와 최고속도 차이가 가장 작은 경로의 최저속도와 최고속도의 차이를 계산하는 프로그램을 작성하시오.   
    
  
**[제한사항]**

1. 지점의 수 N 은 1 이상 1,000 이하의 정수이며, 각 지점은 1 부터 N 사이의 숫자로 표현한다.

2. 도로의 수 M 는 1 이상 4,000 이하의 정수이다.

3. 각 도로마다 유지해야 하는 속도는 1 이상 1,000,000,000(10억) 이하의 정수이다.

4. 시작 지점에서 도착 지점까지 주어진 도로를 이용하여 도달하지 못하는 경우는 주어지지 않는다.

**[입력]**

맨 처음 테스트 케이스의 개수 C 가 주어지며 그 다음 C 개의 테스트 케이스가 주어진다.

각 테스트 케이스는 여러 줄로 구성되며 첫 줄에 N, M 의 값이 공백으로 구분되어 주어진다.

그 다음 M개의 줄에 각 도로의 정보를 나타내는 정수 A, B, S 가 공백으로 구분되어 주어진다. A 와 B 는 도로가 연결하고 있는 두 지점의 번호이며, S 는 해당 도로에서 유지해야 하는 속도이다.

다음 줄에 시작 지점의 번호와 도착 지점의 번호가 공백으로 구분되어 주어진다.

**[출력]**

각 테스트 케이스에 대해 #x (x는 테스트 케이스 번호, 1부터 시작) 을 출력하고 공백을 하나 둔 다음,

주어진 테스트 환경에서 시작 지점에서 도착 지점으로 가는 경로 중 최저속도와 최고속도의 차이가 가장 작은 경로의 최저속도와 최고속도의 차이를 출력한다.

**[입출력 예]**

(입력)   
3                     // 케이스 개수   
6 8   
5 1 10   
3 5 6   
2 6 2   
5 6 7   
1 3 7   
4 6 4   
3 6 5   
3 4 7   
1 2   
6 8   
1 6 16   
4 1 12   
6 4 20   
2 6 16   
3 5 7   
2 4 5   
3 2 1   
6 3 1   
1 5   
7 9   
7 2 17   
3 5 1   
2 1 2   
7 1 11   
6 4 24   
5 6 9   
6 3 2   
2 5 15   
2 3 14   
7 4

(출력)   
#1 5   
#2 11   
#3 15

(sample\_input.txt 에 대한 출력)   
#1 5   
#2 7   
#3 17   
#4 0   
#5 5163   
#6 21786   
#7 960891237   
#8 19603   
#9 25266   
#10 23936

문제: **(중) [기출A-0028] 제품의 일련번호 2**

|  |  |
| --- | --- |
| 시간 제한 | **20** 개의 Test Input 입력시 C/C++ **1**초, Java **1.5**초 |
| 메모리 제한 | Stack : **1** Mbytes  /  Total : **256** Mbytes |
| 코드길이 제한 | **128** Kbytes |
| 채점 | 답안을 제출하면 Test Input에 대한 결과를 판정해서 실시간으로 알려준다.  모든 케이스에 대해 정답일 경우 **Pass**, 그 외의 경우 **Fail** 이다. |

한 제조회사에서는 제품 일련번호의 일정 부분에 알파벳 소문자로 구성된 문자열을 이용하고 있다.  
이 문자열은 생산된 순서를 의미하기 때문에 제품이 생산된 순서대로 일정 개수의 알파벳을 열거하여 붙이고 있다.  
가령 a, b, c, d 4개의 알파벳 소문자를 이용하는 경우 첫 번째로 생산된 제품부터 순서대로  
  
abcd, abdc, acbd, acdb, adbc, adcb, bacd, badc, bcad, bcda, bdac, bdca, ..., dabc, dacb, dbac, dbca, dcab, dcba  
  
의 문자열이 제품에 붙여지게 된다. 이 중 adbc는 5번째 생산된 제품에 붙여지는 일련번호에 사용이 되고  
bdca는 12번째 생산된 제품에 붙여지는 문자열이 된다. 그리고 두 제품의 일련번호 사이에는 6개의 제품이 생산되었음을 알 수 있다.  
  
일련번호의 문자열에 사용되는 알파벳의 개수 N 이 주어지고 두 개의 문자열이 주어질 때  
두 일련번호 문자열 사이에 몇 개의 제품이 생산되었는지 개수를 출력하는 프로그램을 작성하시오.  
단, 같은 문자열은 입력 값으로 주어지지 않으며 N개의 알파벳을 이용하는 경우  
a부터 알파벳 순서대로 N개까지 알파벳 소문자를 이용하여 일련번호 문자열을 만들게 된다.  
즉, 6개의 알파벳을 이용하는 경우 a, b, c, d, e, f 6개의 알파벳을 이용하여 문자열을 만들게 된다.

**[제한조건]**알파벳의 개수 N 은 **4 이상 18 이하의 정수**이다.

**[입력]**최초 테스트케이스의 개수 T 가 주어지며 다음 줄부터 T 개의 테스트 케이스가 주어진다.   
각 테스트 케이스는 한 줄로 구성이 되며 처음 일련번호에 사용되는 알파벳의 개수 N 이 주어진다.   
이후 공백을 사이에 두고 두 개의 일련번호 문자열이 공백으로 구분되어 주어진다.

**[출력]**각 테스트 케이스에 대하여 #x(x는 테스트 케이스 번호를 의미)를 출력하고 공백을 하나 둔 다음   
두 일련번호 사이에 생산된 제품의 개수를 출력한다.

**[입출력 예]**(입력)  
5  
4 adbc bdca  
5 ebdac eadcb  
6 ecfabd dfcaeb  
8 dgceabhf fhagbcde  
10 cfhjdabegi cbhagfdije

(출력)  
#1 6  
#2 4  
#3 76  
#4 10606  
#5 125108

문제: **(중상) [교육P-0033] 제품의 일련번호 3**

|  |  |
| --- | --- |
| 시간 제한 | **100** 개의 Test Input 입력시 C/C++ **1**초, Java **1.5**초 |
| 메모리 제한 | Stack : **1** Mbytes  /  Total : **256** Mbytes |
| 코드길이 제한 | **128** Kbytes |
| 채점 | 답안을 제출하면 Test Input에 대한 결과를 판정해서 실시간으로 알려준다.  모든 케이스에 대해 정답일 경우 **Pass**, 그 외의 경우 **Fail** 이다. |

※ Java의 경우 java.math 패키지를 사용할 수 없음.  
  
한 제조회사에서는 제품 일련번호의 일정 부분에 알파벳 소문자로 구성된 문자열을 이용하고 있다.  
이 문자열은 생산된 순서를 의미하기 때문에 제품이 생산된 순서대로 일정 개수의 알파벳을 열거하여 붙이고 있다.  
가령 a, b, c, d 4개의 알파벳 소문자를 이용하는 경우 첫 번째로 생산된 제품부터 순서대로  
  
abcd, abdc, acbd, acdb, adbc, adcb, bacd, badc, bcad, bcda, bdac, bdca, ..., dabc, dacb, dbac, dbca, dcab, dcba  
  
의 문자열이 제품에 붙여지게 된다. 이 중 adbc는 5번째 생산된 제품에 붙여지는 일련번호에 사용이 되고  
bdca는 12번째 생산된 제품에 붙여지는 문자열이 된다. 그리고 두 제품의 일련번호 사이에는 6개의 제품이 생산되었음을 알 수 있다.  
  
일련번호의 문자열에 사용되는 알파벳의 개수 N 이 주어지고 두 개의 문자열이 주어질 때  
두 일련번호 문자열 사이에 몇 개의 제품이 생산되었는지 개수를 출력하는 프로그램을 작성하시오.  
단, 같은 문자열은 입력 값으로 주어지지 않으며 N개의 알파벳을 이용하는 경우  
a부터 알파벳 순서대로 N개까지 알파벳 소문자를 이용하여 일련번호 문자열을 만들게 된다.  
즉, 6개의 알파벳을 이용하는 경우 a, b, c, d, e, f 6개의 알파벳을 이용하여 문자열을 만들게 된다.

**[제한조건]**알파벳의 개수 N 은 **4 이상 26 이하의 정수**이다.

**[입력]**최초 테스트케이스의 개수 T 가 주어지며 다음 줄부터 T 개의 테스트 케이스가 주어진다.   
각 테스트 케이스는 한 줄로 구성이 되며 처음 일련번호에 사용되는 알파벳의 개수 N 이 주어진다.   
이후 공백을 사이에 두고 두 개의 일련번호 문자열이 공백으로 구분되어 주어진다.

**[출력]**각 테스트 케이스에 대하여 #x(x는 테스트 케이스 번호를 의미)를 출력하고 공백을 하나 둔 다음   
두 일련번호 사이에 생산된 제품의 개수를 출력한다.

**[입출력 예]**(입력)  
5  
4 adbc bdca  
5 ebdac eadcb  
6 ecfabd dfcaeb  
8 dgceabhf fhagbcde  
10 cfhjdabegi cbhagfdije

(출력)  
#1 6  
#2 4  
#3 76  
#4 10606  
#5 125108