

A. 약수의 개수

문제

약수는 어떤 수를 나누었을 때 나머지가 0인 수를 말한다.

예를 들어 6의 약수는 다음과 같다.

$$6 \div 1 = 6$$

$$6 \div 2 = 3$$

$$6 \div 3 = 2$$

$$6 \div 6 = 1$$

위의 계산 결과, 나눈 나머지가 모두 0이므로 6의 약수는 1, 2, 3, 6이다.

승균이는 어떤 자연수의 홀수인 약수와 짝수인 약수의 개수가 궁금해졌다.

자연수 N 이 주어지면, N 의 약수들 중에서 홀수의 개수와 짝수의 개수를 출력해주도록 하자.

단, 1이 모든 수의 약수가 되는 사실은 자명하여 1은 제외하고 세기로 하였다.

입력

첫째 줄에 자연수 $N(N \leq 1000000)$ 이 주어진다.

출력

N 의 약수들 중에서 1을 제외한 나머지 약수의 홀수 개수와 짝수 개수를 하나의 줄에 출력한다.

예제 입력

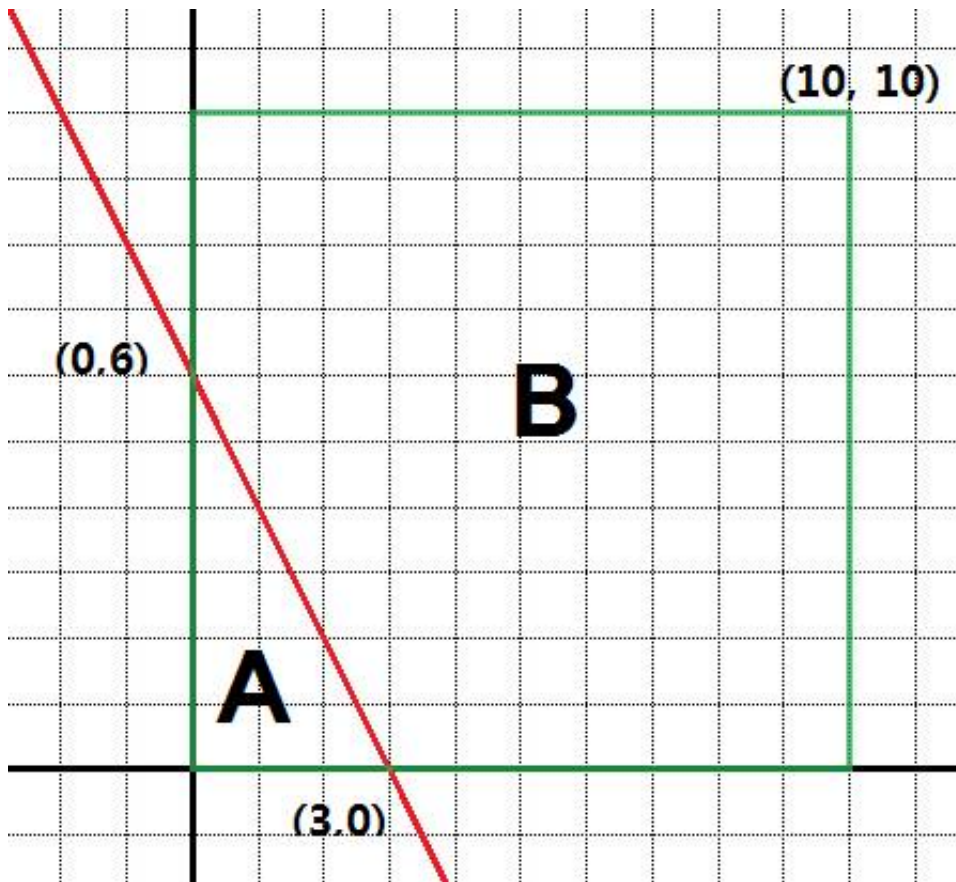
6

예제 출력

1 2

B. 큰 넓이 구하기

문제



그림과 같이 원점을 지나며, 1사분면 위의 점을 지나는 직사각형 안에 두 점을 지나는 직선이 주어진다.

직선을 기준으로 사각형을 나누었을 때, 분할된 두 영역 중 큰 넓이를 출력하시오.

직선은 항상 사각형 위의 두 점과 접하도록 주어진다.

직선과 사각형이 만나는 두 점 중 한 점은 사각형의 좌측변 위, 다른 점은 사각형의 아랫변 위에 놓여져 있다.

입력

첫 번째 줄에 사각형의 오른쪽 위쪽 점 (a, b) 를 가리키는 두 정수가 공백으로 구분되어 주어진다. ($1 \leq a, b \leq 1000$)

두 번째 줄에 직선의 y절편인 c 의 값이 주어진다. ($0 < c \leq a$) 세 번째 줄에 직선의 x절편인 d 의 값이 주어진다. ($0 < d \leq b$)

출력

직선에 의해 분할된 두 영역 중 큰 넓이를 소수점 첫 번째 자리까지 출력하시오.

예제 입력 1

```
10 10
10
10
```

예제 출력 1

```
50.0
```

예제 입력 2

```
11 11
6
3
```

예제 출력 2

```
112.0
```

예제 입력 3

```
10 10
3
3
```

예제 출력 3

```
95.5
```

C. a2Z Number

문제

현대 인류는 10진수를 사용하지만, 과거에는 2진법, 7진법, 12진법도 사용하였다.

미래에는 계산법이 바뀐다면 다른 진법을 사용할 것이다. 현대 문자들을 사용하여 이를 유추해 보자한다.

X진법으로 표현된 숫자 N이 주어진다. 이를 10진법으로 바꾸어 출력하는 프로그램을 작성하시오.

10을 넘어가는 경우에는 대문자 알파벳, 소문자 알파벳으로 대신 표현한다.

크기는 0, ... 9, A, B, ... Z, a, b, ... z 의 순서로 증가한다.

입력

첫째 줄에 테스트케이스의 개수 T가 주어진다.

다음 T줄에 걸쳐 각 줄마다 X와 N이 공백을 사이에 두고 주어진다.

출력

각 테스트케이스마다 10진법으로 변환한 숫자를 한 줄씩 출력한다.

변환된 숫자는 항상 2,147,483,647를 넘지 않으며, 표현될 수 없는 경우는 없다.

예제 입력

```
4
8 777
10 150530
16 22F0
60 alps
```

예제 출력

511

150530

8944

7948314

D. 최솟값과 최대값

문제

MAX, MIN, 그리고 임의의 정수로만 이루어진 수열이 있습니다. 이 수열은 두 수 중의 최대값 또는 최솟값을 여러 차례 구하기 위한 표현식인데, 식을 모두 계산하면 결과로 단 하나의 정수만 남습니다. 예를 들어, $\text{MAX MIN } 4 \ 6 \ 5$ 는 다음과 같이 계산됩니다.

1. $\text{MIN } 4 \ 6$ 을 계산합니다. 그 결과는 4와 6 둘 중에서 작은 값인 4가 됩니다.
2. $\text{MAX MIN } 4 \ 6 \ 5$ 를 계산합니다. $\text{MIN } 4 \ 6$ 이 4와 같으므로, 이 식은 $\text{MAX } 4 \ 5$ 로 표현할 수 있습니다. 따라서 그 결과는 두 수 중 큰 값인 5입니다.

즉, MAX와 MIN을 함수의 형태로 표현하면, 위의 식은 $\text{MAX}(\text{MIN}(4, 6), 5) = 5$ 와 같이 나타낼 수 있습니다.

정확한 규칙은 다음과 같습니다.

- MAX(MIN) 표현식은 반드시 $\text{MAX(MIN)} \langle \text{표현식 또는 정수} \rangle \langle \text{표현식 또는 정수} \rangle$ 와 같은 형태를 취합니다.
- MAX 표현식은 MAX를 뒤따르는 두 개의 정수 또는 표현식의 결과값 중 큰 값을 결과값으로 합니다.
- MIN 표현식은 MIN을 뒤따르는 두 개의 정수 또는 표현식의 결과값 중 작은 값을 결과값으로 합니다.

입력

입력은 여러 테스트 케이스로 구성됩니다. 입력의 첫 줄은 테스트 케이스의 개수를 나타내는 수 $T(1 \leq T \leq 30)$ 가 주어집니다. 각 테스트 케이스마다 한 줄의 입력이 주어지는데, 이 줄에는 MAX, MIN, 또는 정수들로만 구성된 수열이 공백을 기준으로 나누어져 차례대로 주어집니다. 정수의 범위는 1 이상 10000 이하입니다. 수열의 길이는 3 이상 100 이하이고, 위에서 명시된 문제의 조건대로 항상 정확한 표현식을 이루도록 수열이 주어집니다.

출력

각 테스트 케이스마다 입력으로 들어온 수열의 표현식을 모두 계산한 결과값을 한 줄에 출력합니다.

예제 입력

MAX MIN 4 6 5

MIN 3 MAX 1 2

MAX MAX 8 10 MAX 11 9

예제 출력

5

2

11

E. 티모장인 오복실

문제

티모^[1]장인 오복실 리븐^[1]장인 송평캔과 탑 라인에서 마주치게 되었다. 송평캔을 반드시 이기고 싶은 오복실은 재빨리 구글에 티모학개론을 검색해서 로딩시간동안 정독했다.

티모의 평q평^[2]은 완벽했다. 리븐을 딸 생각에 가득 찬 티모는 눈이 먼 나머지 맵리딩^[3]을 전혀 안하고 있었다! 어느 날 갑자기 아래에서 대머리 땡중 리신이 속 올라오더니 리신의 Q평E평RQ평^[2]과 리븐의 EQR히드라W평Q평Q평R^[2] 콜라보레이션에 티모는 찢겨버리고 만다. 티모는 맵리딩의 중요성을 뼈저리게 느꼈다.

그 후, 깨달음을 얻은 티모는 1000판의 승리 통계를 통해 자신만의 “네맘내맘 맵리딩 타이밍북”을 만들었다. 이 책에 적힌 시간마다 맵을 확인하면 적의 위치를 80%정도 유추할 수 있다고 한다!

내가 한 판 동안 맵리딩했던 순간을 모두 기록했다가 책에 적힌대로 나의 타이밍을 보정하고 싶다면, 매 순간을 최소한 얼마만큼 보정하면 되었는 지 출력하는 프로그램을 만들어주자!

입력

첫째 줄에 “네맘내맘 맵리딩 타이밍북” 책에 적힌 시간의 갯수 N 과 내가 귀환한 시간의 갯수 K 가 주어진다. 둘째 줄에 책에 적힌 시간들 A_1, A_2, \dots, A_N 이 시간 순서대로 주어진다. 셋째 줄에 기록된 나의 맵리딩 시간 B_1, B_2, \dots, B_K 가 시간 순서대로 주어진다. 섬세한 독자를 위해 책에 적힌 숫자의 단위는 초(sec)로 제공되고 있다.

단, $1 \leq N, K \leq 1,000,000$ 이고, $1 \leq A_i, B_i \leq 1,234,567$ 이다.

출력

내가 귀환한 시간들에 대해 각각 얼마만큼 보정하면 완벽한 타이밍이 나올 지 출력하시오. 예를 들어, 책에는

4	9	14	19
---	---	----	----

 마다 확인하라고 적혀있고, 나는

3	11
---	----

 에 확인했다면 각각

+1	-2
----	----

 씩 보정하여

4	9
---	---

 와 일치시킬 수 있다. 만약 보정할 수 있는 시간이 여러 개라면, 더 빠른 시간에 확인할 수 있도록 보정한다.

예제 입력

4 9 14 19 29

3 11 19 24

예제 출력

+1 -2 +0 -5

[1] : League Of Legends의 캐릭터 이름

[2] : 스킬 콤보 순서 (입롤 아님)

[3] : 맵을 주기적으로 확인하여 적과 아군의 위치를 파악하는 행동

F. 그립기는 거꾸로 해도 그립기

문제

팰린드롬(Palindrome)이란, 거꾸로 읽어도 제대로 읽는 것과 같은 문장이나 낱말이다. 예를 들면, 토마토, 별똥별, 야 이 똥똥한 똥똥이야, 대한총기공사 공기총 한 대는 팰린드롬이지만 이강민 허접, 난쟁이 똥자루, 트롤 김창영 등은 말은 맞아도 팰린드롬이라고 할 수 없다.

우리는 어떠한 단어가 주어지면, 여기에 문자를 더하거나 빼서 팰린드롬으로 만들 수 있다. 명료하게 이야기하자면 더하거나 빼는 행동은 다음과 같다.

- 어떤 위치에 문자를 하나 삽입한다.
- 어떤 위치의 문자를 하나 삭제한다.

단어가 주어졌을 때, 위 행동을 반복하여 팰린드롬으로 만든다면 최소한 몇 번의 행동을 해야 할까?

입력

한 줄에 공백없는 단어가 입력된다. 단어의 길이는 1000자리를 넘지 않는다.

출력

최소한 몇 번의 행동을 해야 팰린드롬으로 바꿀 수 있는 지 출력하시오.

예제 입력

aabcbba

예제 출력

2

G. diff

문제

diff는 두 파일의 차이점을 구하고 출력해주는 프로그램입니다. 하나의 파일을 다른 파일로 변환하기 위한 가장 적은 수의 행 단위의 삭제와 삽입 연산을 구합니다. 캐나다의 유명한 프로그래머인 Dean Kim은 diff가 행 단위로만 동작하는데 불만을 느끼고, 문자 단위로 동작하는 새로운 diff 프로그램을 작성할 예정입니다.

두 문자열 S와 T가 주어졌을때, 문자열 S가 T로 변환되기 위한 최소의 삽입, 삭제 연산을 구하는 이 프로그램의 동작 방식은 다음과 같습니다. 먼저, 문자열 S와 T로 부터 가장 긴 공통 부분 수열(longest common subsequence)을 구합니다. 간단히 말하면, S와 T에서 공통으로 나타난 가장 긴 문자열이지만, 순서만 유지한다면 문자들이 S와 T로부터 연이어서 구성될 필요가 없는 문자열입니다.

```
a b c d f g h j q z
a b c d e f g i j k r x y z
```

위의 두 문자열에서 가장 긴 공통 부분 수열은 `a b c d f g j z` 입니다. 또한 위의 문자열을 S, 아래 문자열을 T라고 할 때, 삭제된 문자들은 `h q` 이고 삽입된 문자들은 `e i k r x y` 입니다.

Dean Kim을 도와 두 문자열 S와 T가 주어졌을때, 가장 긴 공통 부분 수열과 삽입 횟수, 삭제 횟수를 구하는 프로그램을 작성하세요.

입력

입력의 첫 줄은 테스트 케이스의 개수를 나타내는 수 $T(1 \leq T \leq 30)$ 가 주어집니다. 각 테스트 케이스마다 두 줄에 걸쳐 S와 T가 차례로 주어집니다. S와 T는 알파벳 소문자만으로 구성된 100자 이하, 2자 이상의 문자열입니다.

출력

각 테스트 케이스마다 한 줄에 가장 긴 공통 부분 수열, 삽입 횟수, 삭제 횟수를 공백을 기준으로 나누어 차례로 출력하세요.

- 가장 긴 공통 부분 수열이 두 가지 이상 존재할 수 있다면, 그 중에서 사전 순으로 가장 빠른 문자열을 고르세요.
- 공통 부분 수열이 존재하지 않는 테스트 케이스의 경우, 삽입 횟수와 삭제 횟수만 출력합니다.

예제 입력

```
5
abcdfghjqz
abcdefghijklrxyz
efgbcefgh
abcdefghi
abxyzcde
xyz
vwxyzabcde
abcdevwxyz
ab
xyz
```

예제 출력

```
abcdfgjz 6 2
bcefgh 3 3
xyz 0 5
abcde 5 5
3 2
```

H. 건설 로봇

문제

가까운 미래 20XX년, 과학 기술의 발달로 모든 건물을 로봇만으로 세우는 일이 가능해졌습니다. 정 씨는 건설 로봇들이 분주하게 일하고 있는 현장에서 로봇 통제와 관리 담당을 맡고 있습니다. 그런데 정 씨는 현장에 투입된 로봇들 중 일부는 일하지 않고 쉬는 모습을 발견할 수 있었습니다. 그래서 쉬는 로봇들을 모두 작업에 투입시키려 했지만, 아래의 이유들로 그것이 불가능하다는 것을 깨달았습니다.

- 공사 현장에는 로봇이 수행할 여러 작업들이 있습니다.
- 모든 로봇들은 한 번에 하나의 작업만을 수행할 수 있습니다.
- 로봇의 기종에 따라 수행 가능한 작업과 불가능한 작업이 있습니다.
- 작업마다 최대 투입 가능한 로봇의 수가 제한되어 있습니다.

이러한 상황에서 전직 컴퓨터공학 박사였던 정 씨는 최대한 노는 로봇 없이 가능한 많은 로봇이 일하게 만들려고 합니다. 즉, 가능한 많은 로봇들이 어떤 작업이든 맡아 수행해야 한다는 뜻입니다. 정 씨를 돕기 위해 이를 검사해주는 프로그램을 작성해보세요. 자세한 요구사항은 아래와 같습니다.

현장에는 총 W 개의 작업이 있고, 각 작업은 1부터 W 까지 번호가 매겨져 있습니다. 로봇이 아무리 많아도, 작업 i ($1 \leq i \leq W$)에는 최대 L_i 개의 로봇이 투입될 수 있습니다. 로봇들은 총 K 개의 기종 중에 어떤 하나의 기종에 속합니다. 각 기종은 1부터 K 까지 번호가 매겨져 있습니다. 기종 j ($1 \leq j \leq K$)에 속하는 로봇은 N_j 개만큼 존재합니다. 즉, 현장에 존재하는 모든 로봇들의 수는 $N_1 + N_2 + N_3 + \dots + N_K$ 입니다. 또한, 모든 작업과 로봇 기종에 대한 쌍 i, j 에 대하여, 작업 수행 가능 여부 C_{ij} 가 존재합니다. C_{ij} 는 0 또는 1의 값을 가지는데, 1인 경우 기종 j 에 속하는 로봇은 작업 i 를 수행 가능하다는 의미이며, 0이면 불가능함을 뜻합니다.

이 상황에서 최대한 많은 로봇이 일할 때, 작업에 투입한 모든 로봇의 수를 구하면 됩니다.

입력

입력의 첫 줄은 테스트 케이스의 개수를 나타내는 수 T ($1 \leq T \leq 30$)가 주어집니다. 각 테스트 케이스마다 첫 줄에는 W ($1 \leq W \leq 50$)와 K ($1 \leq K \leq 50$)가 주어집니다. 바로 다음 줄에는 $L_1, L_2, L_3, \dots, L_W$ 가 주어집니다. ($1 \leq L_i \leq 10000$) 그리고 K 개의 줄이 입력으로 더 주어집니다. K 개의 줄 중에서 j 번째 줄에 주어지는 입력은 차례대로 N_j , 그 다음 $C_{1j}, C_{2j}, C_{3j}, \dots, C_{Wj}$ 가 주어집니다. ($1 \leq C_{ij} \leq 10000$)

출력

각 테스트 케이스마다 한 줄에 최대한 많은 로봇들이 작업에 투입되었을 때, 투입된 모든 로봇의 수를 출력하세요.

예제 입력

```
3
4 3
10 10 10 10
30 1 1 0 0
40 0 1 1 0
20 0 0 1 1
4 3
20 30 20 20
30 1 1 0 0
40 0 1 1 0
20 0 0 1 1
5 5
10 20 40 20 60
30 1 0 0 0 1
30 1 1 1 0 0
30 0 1 1 1 0
30 0 0 0 1 0
30 0 0 0 0 1
```

예제 출력

```
40
90
140
```