

행복 서버테스트 출처

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	512 MB	3326	2250	1975	74.697%

문제

코이 초등학교에 새로 부임하신 교장 선생님은 어린 학생들의 행복감과 학생들의 성적 차이 관계를 알아보기로 했다. 그래서 이전 성적을 조사하여 학생들의 시험 점수 차이 변화를 알아보려고 한다.

예를 들어서 2016년 학생 8명의 점수가 다음과 같다고 하자.

27, 35, 92, 75, 42, 53, 29, 87

그러면 가장 높은 점수는 92점이고 가장 낮은 점수는 27점이므로 점수의 최대 차이는 65이다. 한편 2017년 학생 8명의 점수가 다음과 같았다.

85, 42, 79, 95, 37, 11, 72, 32

이때 가장 높은 점수는 95점이고 가장 낮은 점수는 11점이므로 점수의 최대 차이는 84이다.

N 명 학생들의 점수가 주어졌을 때, 가장 높은 점수와 가장 낮은 점수의 차이를 구하는 프로그램을 작성하시오

입력

표준 입력으로 다음 정보가 주어진다. 첫 번째 줄에는 학생 수 N 이 주어진다. 다음 줄에는 N 명의 학생 점수가 공백 하나를 사이에 두고 주어진다.

출력

표준 출력으로 가장 높은 점수와 가장 낮은 점수의 차이를 출력한다.

제한

모든 서버테스트에서 $2 \leq N \leq 1,000$ 이고 입력되는 학생들의 점수는 0 이상 1,000 이하의 정수이다.

서브테스크 1 (23점)

학생 수가 2명인 경우만 존재한다.

서브테스크 2 (29점)

점수가 낮은 점수부터 높은 점수까지 순서대로 주어진다.

서브테스크 3 (48점)

클래의 제약 조건 이외에 아무 제약 조건이 없다.

예제 입력 1 복사

```
5
27 35 92 75 42
< >
```

예제 입력 2 복사

```
8
85 42 79 95 37 11 72 32
< >
```

예제 출력 1 복사

```
65
< >
```

예제 출력 2 복사

```
84
< >
```

수빈이와 수열 출처 다국어 분류

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	32 MB	1920	1501	1373	81.484%

문제

수빈이는 심심해서 수열을 가지고 놀고 있다. 먼저, 정수 수열 A 를 쓴다. 그리고 그 아래에 정수 수열 A 의 해당 항까지의 평균값을 그 항으로 하는 정수 수열 B 를 쓴다.

예를 들어, 수열 A 가 1, 3, 2, 6, 8이라면, 수열 B 는 $1/1, (1+3)/2, (1+3+2)/3, (1+3+2+6)/4, (1+3+2+6+8)/5$, 즉, 1, 2, 2, 3, 4가 된다.

수열 B 가 주어질 때, 수빈이의 규칙에 따른 수열 A 는 뭘까?

입력

첫째 줄에는 수열 B 의 길이만큼 정수 $N(1 \leq N \leq 100)$ 이 주어지고, 둘째 줄에는 수열 B_i 를 이루는 N 개의 정수가 주어진다. ($1 \leq B_i \leq 10^9$)

출력

첫째 줄에는 수열 A 를 이루는 N 개의 정수를 출력한다. ($1 \leq A_i \leq 10^9$)

예제 입력 1 복사

```
1
2
```

예제 출력 1 복사

```
2
```

예제 입력 2 복사

```
4
3 2 3 5
```

예제 출력 2 복사

```
3 1 5 11
```

예제 입력 3 복사

```
5
1 2 2 3 4
```

예제 출력 3 복사

```
1 3 2 6 8
```

$B \rightarrow A$

$$b_i = \frac{\sum a_i}{i}$$

$$i \cdot b_i = \sum a_i$$

$\hookrightarrow a_i$ sum을 곱

이름궁합 테스트 출처

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	256 MB	605	352	311	61.952%

문제

시유이는 좋아하는 여성이 생기면 가장 먼저 이름궁합부터 본다. 이름궁합을 보는 방법은 간단하다. 먼저 이름을 알파벳 대문자로 적는다. 각 알파벳 대문자에는 다음과 같이 알파벳을 적는데 필요한 확수가 주어진다. 예를 들어, 두 사람의 이름인 LEESIYUN, MIYAWAKISAKURA 를 같이 표현했을 때 다음과 같이 먼저 주어진 이름부터 한 글자씩 적는다.

알파벳	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
확수	3	2	1	2	4	3	1	3	1	1	3	1	3	2	1	2	2	2	1	2	1	1	1	2	2	1

두 사람의 이름을 알파벳 대문자로 표현한 뒤, 한 글자씩 번갈아가며 적는다.

예시: L M E I E Y S A I W Y A U K N I S A K U R A

예시처럼 이름이 남을 경우엔 뒤에 남은 글자인 **SAKURA**를 맨 뒤에 적는다. 그리고 나서 알파벳을 대응하는 숫자로 바꾸고 각 숫자와 그 숫자의 오른쪽 숫자와 더한 것을 밑에 적는다. 더한 숫자가 **10이 넘을 경우엔 일의 자리 수만 남긴다.** 이 과정을 반복하여 숫자가 2개만 남았을 때 남은 숫자가 두 사람의 궁합이 좋을 확률이 된다.

과정을 자세히 나타내면 다음과 같다.
 초기 상태 : 1 3 4 1 4 2 1 3 1 1 2 3 1 3 2 1 1 3 3 1 2 3
 한번 수합 : 4 7 5 5 6 3 4 4 2 3 5 4 4 5 3 2 4 6 4 3 5
 두번 수합 : 1 2 0 1 9 7 8 6 5 8 9 8 9 8 5 6 0 0 7 8
 세번 수합 : 3 2 1 0 6 5 4 1 3 7 7 7 7 3 1 6 0 7 5
 ...
 19번 수합 : 5 7 0
 20번 수합 : 2 7
 따라서 LEESIYUN과 MIYAWAKISAKURA의 궁합이 좋을 확률이 27%이다.

입력

첫 번째 줄에 이름의 길이 N 과 M 을 받는다. ($2 \leq N, M \leq 100$)

다음 줄에 이름 A 와 B 를 입력받는다. 이름은 반드시 알파벳 대문자만 주어진다.

출력

A 와 B 의 이름궁합이 좋을 확률을 %로 출력한다. 단, 십의 자리가 0일 경우엔 일의 자리만 출력한다.

예제 입력 1 복사

```
8 14
LEESIYUN MIYAWAKISAKURA
```

예제 출력 1 복사

```
27%
```

예제 입력 2 복사

```
2 2
AB CD
```

예제 출력 2 복사

```
77%
```

예제 입력 3 복사

```
3 2
BOJ IN
```

예제 출력 3 복사

```
1%
```

$$\begin{array}{r}
 54 + 53 + 46 \\
 \hline
 107 \quad 99 \\
 \hline
 77 \quad 77
 \end{array}$$

→ tip

결과 동일

→ 4자리 에 대한 연산, 결국 맨아지막에만 적용해주면 되는 부분

보너스 점수 출처

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	512 MB	832	637	588	78.610%

문제

승고한 알고리즘 캠프 퀴즈 타임이 시작되었다! PS 기초, 동적 계획법, 파라메트릭 서치, 욕제의 생일, 탐색, 그리디, 최단경로 알고리즘, 구데 기검, 서로스 집합, 최소 신장 트리, 최소 공통 조상, 세그먼트 트리, 코드포스에서 C++로 높은 수준의 난수를 생성하는 방법, 최대 유량, 블록 편집, 스타트링크 사무실에 있는 게임용 컴퓨터의 RAM의 총 용량 등등 수많은 주제를 총망라하고 있는 이 미니 대회는 수많은 참가자들의 도전으로 오늘도 맞나고 있고, 제출된 OX표의 개수는 셀 수 없을 정도이다.

운영진들은 이 OX표들을 채점하고, 점수를 계산해낸 다음, 시상식을 진행하며 화기에매하게 행사를 마무리해야 한다. 승고한 알고리즘 캠프 퀴즈 타임에서 OX표의 점수는 다음과 같이 계산된다.

- OX표에 N 개의 문제들이 있을 때, 1번 문제, 2번 문제, ..., N 번 문제 순으로 채점된다.
- 문제는 뒤로 갈수록 어려워지기 때문에, i 번 문제의 기본 점수는 i 점이다.
 - 문제를 맞으면 그 문제의 기본 점수(즉, i 번 문제의 경우 i 점)를 획득하며, 틀리면 얻지 못한다.
- 기본 점수와 별개로, '보너스 점수'라는 값이 존재한다. 이는 처음에는 0점이다.
 - 문제를 맞으면 그 때의 '보너스 점수'를 획득하고, '보너스 점수'의 값이 1점 증가한다.
 - 문제를 틀리면 '보너스 점수'를 얻지 못하고, '보너스 점수'의 값이 0점으로 초기화된다.

민성이는 열렬함에 승고한 알고리즘 캠프 퀴즈 타임의 OX표를 채점해야 하는 업무를 맡게 되었다. 수많은 OX표를 볼 생각에 머리가 지끈거리는 민성이는 프로그램을 세워 이를 자동화하려고 한다. 시상식까지 4시간밖에 남지 않은 민성이를 도와 점수를 계산해주자.

입력

첫 번째 줄에 OX표의 길이인 자연수 N 이 주어진다. ($1 \leq N \leq 10,000$) $\rightarrow O(N)$

두 번째 줄에 OX표를 의미하는 문자열 S 가 주어진다. S 는 0(알파벳 대문자 O, ASCII 코드 79)와 X(알파벳 대문자 X, ASCII 코드 88)로만 구성되어 있으며, 길이는 N 이다.

문자열 S 의 i 번째 글자가 0 이면 해당 참가자가 i 번째 문제를 맞았음을 의미하고, X 이면 틀렸음을 의미한다.

출력

첫 번째 줄에 입력으로 들어온 OX표의 점수를 출력한다.

예제 입력 1 복사

```
8
X000X00X
```

예제 출력 1 복사

```
26
```

민성이는 다음과 같이 총 26점을 받게 된다.

- 1번 문제를 틀렸으므로 점수를 얻지 못하고 보너스 점수가 0점으로 초기화된다. (총 점수 0점)
- 2번 문제를 맞았으므로 기본 점수 2점과 보너스 점수 0점을 획득하며, 보너스 점수가 1점으로 증가한다. (총 점수 2점)
- 3번 문제를 맞았으므로 기본 점수 3점과 보너스 점수 1점을 획득하며, 보너스 점수가 2점으로 증가한다. (총 점수 6점)
- 4번 문제를 맞았으므로 기본 점수 4점과 보너스 점수 2점을 획득하며, 보너스 점수가 3점으로 증가한다. (총 점수 12점)
- 5번 문제를 틀렸으므로 점수를 얻지 못하고 보너스 점수가 0점으로 초기화된다. (총 점수 12점)
- 6번 문제를 맞았으므로 기본 점수 6점과 보너스 점수 0점을 획득하며, 보너스 점수가 1점으로 증가한다. (총 점수 18점)
- 7번 문제를 맞았으므로 기본 점수 7점과 보너스 점수 1점을 획득하며, 보너스 점수가 2점으로 증가한다. (총 점수 26점)
- 8번 문제를 틀렸으므로 점수를 얻지 못하고 보너스 점수가 0점으로 초기화된다. (총 점수 26점)

걸그룹 마스터 준석이 출처

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초 (추가 시간 없음)	128 MB	857	533	356	62.898%

→ dictionary

문제

정우는 소문난 걸그룹 덕후이다. 정우의 친구 준석이는 걸그룹을 좋아하지만 이름을 잘 외우지 못한다는 문제가 있었다. 정우는 친구를 위해 걸그룹 개인과 팀의 이름을 검색하여 외우게 하는 퀴즈 프로그램을 만들고자 한다.

입력

첫 번째 줄에는 총 입력 받을 걸그룹의 수 N ($0 < N < 100$)과 맞혀야 할 문제의 수 M ($0 < M < 100$)을 입력받는다.

두 번째 줄부터는 각 걸그룹마다 팀의 이름, 걸그룹의 인원 수, 멤버의 이름을 한 줄씩 차례대로 입력받는다. 팀과 멤버의 이름은 최대 100글자이며, 모든 글자는 알파벳 소문자이다. 하나의 걸그룹이나 서로 다른 두 걸그룹에 이름이 같은 두 멤버가 있는 경우는 없다.

그 다음 줄부터는 M 개의 퀴즈를 입력받는다. 각각의 퀴즈는 두 줄로 이루어져 있으며, 팀의 이름이나 멤버의 이름이 첫 줄에 주어지고 퀴즈의 종류를 나타내는 0 또는 1이 두 번째 줄에 주어진다. 퀴즈의 종류가 0일 경우 팀의 이름이 주어지며, 1일 경우 멤버의 이름이 주어진다.

출력

첫 번째 줄부터 차례대로 퀴즈에 대한 답을 출력한다. 퀴즈의 종류가 0일 경우 해당 팀에 속한 멤버의 이름을 사전순으로 한 줄에 한 명씩 출력한다. 퀴즈의 종류가 1일 경우 해당 멤버가 속한 팀의 이름을 출력한다.

예제 입력 1 복사

```
3 4
twice
9
jihyo
dahyeon
mina
momo
chaeyoung
jeongyeon
tzuyu
sana
nayeon
blackpink
4
jisu
lisa
rose
jenny
redvelvet
5
wendy
irene
seulgi
yeri
joy
sana
1
wendy
1
twice
0
rose
1
```

예제 출력 1 복사

```
twice
redvelvet
chaeyoung
dahyeon
jeongyeon
jihyo
mina
momo
nayeon
sana
tzuyu
blackpink
```

이중
인원수
(인원)

APC는 왜 서버테스크 대회가 되었을까? **서버테스크 출처**

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	512 MB	769	371	333	73.510%

문제

2019년 올해도 어김없이 아주대학교 프로그래밍 경시대회(Ajou Programming Contest, APC)가 열렸다! 올해 새롭게 APC의 종강독을 맡게 된 준표는 대회 출제 과정 중 큰 고민에 빠졌다. APC에 참가하는 참가자들이 너무 다양해 대회 문제 난이도 설정이 너무 어렵기 때문이다. APC는 프로그래밍 대회에 익숙하지 않은 학생들과 전공생이 아닌 학생들도 대거 참가하기 때문에 모두가 풀거나 도전할 수 있는 난이도 커브를 갖춰야 한다. 또한 '경인지역 6개대학 연합 프로그래밍 경시대회 shake!'에 참가할 학교 대표 10인을 선발하기 위한 대표 선발전으로서의 변별력도 갖추어야 하며, 외부인들이 따로 참가할 수 있는 Open Contest가 동시에 진행되기 때문에 소위 '고인물'들에게 한 시간도 안되어 대회가 정복당하는 일도 막고 싶다. 여기에 APC 출제진인 준표, 민영, 현정, 준서는 문제를 준비하는 데 무척 고생을 했기 때문에 참가자들이 모든 문제를 한번씩은 읽어주었으면 하는 소망도 가지고 있다.

육신 그득한 준표는 고민끝에 이 수많은 니즈를 충족시키기 위한 한가지 해결책을 제안했다. 하나의 문제를 제한조건을 통해 **쉬운 버전**과 **어려운 버전**으로 나누어 쉬운 버전만 맞더라도 **부분점수를 주는 서버테스크 문제로 대회를 구성하는 것**이다. 또한 이렇게 만들어진 문제를 쉬운 버전의 난이도순으로 배치하려 한다.

위와 같이 문제를 준비하면 프로그래밍 대회에 익숙하지 않은 사람은 앞에서부터 따라가면서 도전해볼 수 있어 쉬운 문제를 찾는 데 시간을 쓰지 않을 수 있고, 어려운 버전으로 학교 대표 선발을 위한 변별력을 유지할 수 있으며, 모든 문제가 읽히길 바라는 출제진의 소망도 이를 수 있을 것이다!

	문제번호(쉬운→어려운 난이도순)							
	A	B	C	D	E	F	G	H
쉬운버전 (문제당 300점)	☆	★	☆☆	★★	★★☆	★★★	★★★★	★★★★☆
어려운버전 (문제당 40점)	★★☆	★★★☆☆	★★★★☆	★★★	★★★	★★★☆☆	★★★★	★★★★★

<그림1> 출제중 평가한 문제들의 난이도 예시 (예제2)

<!-- 아래 이야기는 픽션입니다 -->

현정이는 APC에 한 번이라도 나가보고 싶다는 소망이 있다. 하지만 이 소망은 여태까지 단 한 번도, 그리고 앞으로도 이루어질 리 없기 때문에 현정이가 입버릇처럼 하게 된 말이 있는데...

- 현정 : 아-- 나도 APC 참가만 했으면 상금 받는 건데-----
- 준표 : ... 그건 아닌 것 같은데?

현정이의 근거 없는 자신감이 눈앞선 준표는 출제 중에 평가한 문제 난이도를 통해 현정이의 예상 점수를 알려주고 현정이가 현실을 받아들일 수 있도록 도와주고자 한다.

현정이는 L 만큼의 역량을 가지고 있어 **L보다 작거나 같은 난이도의 문제를 풀 수 있다**. 또한 현정이는 코딩이 느리기 때문에 대회 시간이 부족에 **K개보다 많은 문제는 해결할 수 없다**. 어떤 문제에 대해 쉬운 버전을 해결한다면 **100점**을 얻고, 어려운 버전을 해결한다면 여기에 40점을 더 받아 **140점**을 얻게 된다. 어려운 버전을 해결하면 쉬운 버전도 같이 풀리게 되므로, 한 문제를 해결한 것으로 계산한다.

→ 난이도 스도플
~~~~~

현정이가 APC에 참가했다면 최대 몇점을 얻을 수 있었을지 알려주자.

입력

첫 줄에 문제의 개수  $N$ , 현정이의 역량  $L$ , 현정이가 대회중에 풀 수 있는 문제의 최대 개수  $K$ 가 주어진다. 둘째 줄부터  $N$ 개의 줄에 걸쳐 1 -  $N$ 번째 문제의 쉬운 버전의 난이도  $sub1$ , 어려운 버전의 난이도  $sub2$  가 순서대로 주어진다.

출력

현정이가 APC에 참가했다면 받았을 점수의 최대값을 출력한다.

제한

- 1 ≤  $N$  ≤ 100
- 1 ≤  $L$  ≤ 90
- 1 ≤  $sub1$  ≤  $sub2$  ≤ 90

서버테스크 1 (100점)

- $K = N$

서버테스크 2 (40점)

- $0 \leq K \leq N$

예제 입력 1 복사

```
4 8 4
1 6
4 5
6 10
9 12
```

예제 출력 1 복사

```
380
```

1번, 2번 문제의 어려운 버전을 해결해 2×140 = 280점을, 3번 문제의 쉬운 버전을 해결해 100점을 얻어 총 380점을 얻는다. 현정이가 4문제를 풀 수 있을 정도로 대회 시간은 충분하지만, 4번 문제는 현정이에겐 너무 어려워서 풀 수 없다.

예제 입력 2 복사

```
8 7 5
2 5
3 6
4 8
5 6
6 9
6 1
7 10
```

예제 출력 2 복사

```
660
```

예제 입력 3 복사

```
6 9 5
1 6
3 10
4 5
5 20
7 12
8 15
9 30
14 14
```

예제 출력 3 복사

```
550
```

힌트

예제2, 3은 서버테스크1에서는 나오지 않는다. 전통적으로 APC는 쉬운 버전의 문제를 먼저 푸는 것이 정신건강에 안정적인 득점을 위해 좋다.

→ input 들어올때  
easy, hard question 경각  
↓  
answer limit에 맞게 풀기

$$\rightarrow 140 + 140 + 100 = 380$$

$$\begin{aligned} & \text{2개씩 가장 낮은 난이도 7을 4번 이용} \\ & [1, 2, 3, 7] \times 140 \\ & \text{양지4} \\ & = 660 \end{aligned}$$

$$\begin{aligned} & [1, 3] \times 140 \\ & + 100 + 100 + 100 \\ & = 550 \end{aligned}$$

## The candy war 출처

☆

| 시간 제한 | 메모리 제한 | 제출  | 정답  | 맞은 사람 | 정답 비율   |
|-------|--------|-----|-----|-------|---------|
| 1 초   | 128 MB | 356 | 221 | 185   | 65.603% |

### 문제

알고리즘 유치원 선생님이 행하는 간식시간이 되자 아이들에게 사탕을 나누어 주려고 하였다. 하지만 욕심 많고 제멋대로인 유치원 아이들은 차례대로 받으라는 선생님의 말을 무시한 채 마구잡이로 사탕을 집어 갖고 많은 사탕을 집어 간 아이가 있는가 하면 사탕을 거의 차지하지 못하고 우는 아이도 있었다.

말로 타일라도 아이들이 말을 듣지 않아 영희는 한 가지 놀이를 제안했다. 일단 모든 아이들이 **원으로 둘러 앉는다**. 그리고 모든 아이들은 동시에 자기가 가지고 있는 **사탕의 짝 번을 오른쪽 아이**에게 준다. 만약 이 짝과 홀수개의 사탕을 가지게 된 아이가 있을 경우 선생님이 한 개를 보충해 짝수로 만들어 주기로 했다. 흥미로워 보이는 이 놀이에 아이들은 참여 했고 이 과정을 몇 번 거치자 자연스럽게 모든 **아이들이 같은 수의 사탕을 가지게 되어 소란은 종료되었다**.

자기가 가진 사탕의 번을 옆에 오른쪽에 앉은 아이에게 주는 과정과 선생님이 사탕을 보충해 주는 과정을 묶어서 1 순환이라고 할 때 몇 번의 순환을 거치면 모든 아이들이 같은 수의 사탕을 가지게 되는지 계산 해보자. 단, **처음부터 홀수개의 사탕을 가지고 있으면 선생님이 짝수로 보충을 먼저 해주며 이 경우 순환수에 들어가지 않는다**. 선생님은 충분한 수의 사탕을 갖고 있다고 가정하자.

↳ 짝수의 홀수개의 경우 → 짝수로 맞춰준다.

### 입력

입력은 표준입력(standard input)을 통해 받아들인다. 입력의 첫 줄에는 테스트 케이스의 개수 T가 주어진다. 각각의 테스트 케이스의 첫 줄에는 아이의 인원 N ( $1 \leq N \leq 10$ ) 이 주어지고 그 다음 줄에는 각 아이들이 초기에 가지고 있는 사탕의 개수  $C_i$  ( $1 \leq i \leq N$ ,  $1 \leq C_i \leq 30$ )가 주어진다. 분배 시  $C_1$ 의 오른쪽에는  $C_2$ 가,  $C_2$ 의 오른쪽에는  $C_3$ 가..... 같은 식으로 맞게 되며  $C_N$ 의 오른쪽에는  $C_1$ 이 맞게 된다.

### 출력

출력은 표준출력(standard output)을 통하여 출력한다. 각 테스트 케이스에 대하여 모든 아이가 같은 개수의 사탕을 가질 때까지 몇 순환이 걸리는지 출력하시오.

#### 예제 입력 1 복사

```
4
5
2 4 7 8 9
1
9
6
10 5 13 2 7 8
4
3 4 4 3
```

#### 예제 출력 1 복사

```
6
0
4
0
```

2 6 13 21 30  
4 9  
7 8  
동 6개씩

## Mixing Milk

출처 다국어

☆

| 시간 제한 | 메모리 제한 | 제출  | 정답  | 맞은 사람 | 정답 비율   |
|-------|--------|-----|-----|-------|---------|
| 2 초   | 512 MB | 604 | 448 | 401   | 74.122% |

### 문제

Farming is competitive business -- particularly milk production. Farmer John figures that if he doesn't innovate in his milk production methods, his dairy business could get creamed!

Fortunately, Farmer John has a good idea. His three prize dairy cows Bessie, Elsie, and Mildred each produce milk with a slightly different taste, and he plans to mix these together to get the perfect blend of flavors.

To mix the three different milks, he takes three buckets containing milk from the three cows. The buckets may have different sizes, and may not be completely full. He then pours bucket 1 into bucket 2, then bucket 2 into bucket 3, then bucket 3 into bucket 1, then bucket 1 into bucket 2, and so on in a cyclic fashion, for a total of 100 pour operations (so the 100th pour would be from bucket 1 into bucket 2). When Farmer John pours from bucket *a* into bucket *b*, he pours as much milk as possible until either bucket *a* becomes empty or bucket *b* becomes full.

Please tell Farmer John how much milk will be in each bucket after he finishes all 100 pours.

100번 cyclic 수행

### 입력

The first line of the input file contains two space-separated integers: the capacity  $c_1$  of the first bucket, and the amount of milk  $m_1$  in the first bucket. Both  $c_1$  and  $m_1$  are positive and at most 1 billion, with  $c_1 \geq m_1$ . The second and third lines are similar, containing capacities and milk amounts for the second and third buckets.

### 출력

Please print three lines of output, giving the final amount of milk in each bucket, after 100 pour operations.

#### 예제 입력 1 복사

capacity / amount

#### 예제 출력 1 복사

```
10 3
11 4
12 5
```

```
0
10
2
```

### 힌트

In this example, the milk in each bucket is as follows during the sequence of pours:

```
Initial State: 3 4 5
1. Pour 1->2: 0 7 5
2. Pour 2->3: 0 0 12
3. Pour 3->1: 10 0 2
4. Pour 1->2: 0 10 2
5. Pour 2->3: 0 0 12
(The last three states then repeat in a cycle ...)
```