

Problem A

현수막

Time Limit: 1 Second

ANT가 처음 알고리즘 대회를 개최하게 되면서 현수막을 내걸었다.

생각하는 프로그래밍 대회

저번 학기 영상처리 수업을 듣고 배웠던 지식을 최대한 응용 해보고 싶은 혁진이는 이 현수막에서 글자가 몇 개인지 알아보는 프로그램을 만들려 한다.

혁진이는 우선 현수막에서 글자인 부분은 1, 글자가 아닌 부분은 0으로 바꾸는 필터를 적용하여 값을 만드는데 성공했다. 그런데 혁진이는 이 값을 바탕으로 글자인 부분 1이 상, 하, 좌, 우, 대각선으로 인접하여 서로 연결되어 있다면 한 개의 글자라고 생각만 하였다.

혁진이가 필터를 적용하여 만든 값이 입력으로 주어졌을 때, 혁진이의 생각대로 프로그램을 구현하면 글자의 개수가 몇 개인지 출력하여야.

Input

첫 번째 줄에는 현수막의 크기인 M와 N가 주어진다. ($1 \leq M, N \leq 250$)

두 번째 줄부터 M+1 번째 줄까지 현수막의 정보가 1과 0으로 주어지며, 1과 0을 제외한 입력은 주어지지 않는다.

Output

혁진이의 생각대로 프로그램을 구현했을 때, 현수막에서 글자의 개수가 몇 개인지 출력하여야.

Sample Input 1

```
8 19
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 1 0 1 1 1 1 1 0
0 0 1 0 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0
0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0
0 1 1 1 1 1 0 1 0 1 0 1 0 0 0 1 0 0 0
0 1 0 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 0
0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Output for the Sample Input 1

```
3
```

Problem B

앉았다

Time Limit: 1 Second

섯다는 화투를 이용하여 20장의 카드를 가지고 2명 이상이 경기를 하는 게임이다. 이러한 셋다의 규칙을 단순화한 게임이 바로 '앉았다'이다. 앉았다의 규칙은 1, 2, 3, ..., 9, 10이 쓰인 카드가 각 2장씩 주어지며 총 20장의 카드가 사용되며, 2명이 참가한다.

다음은 앉았다의 경기 방법이다.

1. 두 명의 참가자는 순서대로 20장의 카드 중 무작위로 2장의 카드를 가져온다.
2. 상대방이 이미 가지고 간 카드를 중복해서 가져올 수는 없다. 그리고 자신은 어떤 카드를 가져왔는지 알 수 있지만, 상대방이 어떤 카드를 가져갔는지는 알 수 없다.
3. 서로의 패를 공개한다.
4. 강한 족보의 패를 가진 사람이 이긴다. 만약 두 참가자가 같은 족보의 패를 가졌다면 비긴다.

족보 (위에 있는 족보일수록 더 강한 족보이다)

10땡: 두 패가 모두 10

9땡: 두 패가 모두 9

...

2땡: 두 패가 모두 2

1땡: 두 패가 모두 1

9끗: 땡이 아니고, 두 패를 더했을 때 일의 자리의 수가 9

8끗: 땡이 아니고, 두 패를 더했을 때 일의 자리의 수가 8

...

1끗: 땡이 아니고, 두 패를 더했을 때 일의 자리의 수가 1

0끗: 땡이 아니고, 두 패를 더했을 때 일의 자리의 수가 0

예를 들어, 영학이의 패가 1과 1이고, 상대의 패가 10과 9라고 하자. 그렇다면 영학이는 1땡이고, 상대는 9끗이다. 영학이가 더 강한 족보를 가졌으므로 영학이가 상대를 이긴다.

영학이는 자신이 어떤 패를 받았는지 보았고, 상대방의 패는 모르는 상태이다.

영학이는 영악해서 앉았다 게임에서 자신이 이길 확률이 높을 때만 돈을 베팅한다.

영악한 영학이를 위하여 영학이의 패가 입력으로 주어졌을 때 이길 수 있는 확률을 구하여야.

Input

영학이의 패를 뜻하는 두 개의 정수 A, B가 주어진다. ($1 \leq A, B \leq 10$)

Output

영학이가 이길 확률을 소수점 이하 셋째 자리까지 반올림해서 출력하시오.

반드시 뒤따르는 0도 출력해야 한다. 예를 들어, '0.700'을 출력했다면 정답이고, '0.7'을 출력했다면 오답이다.

Sample Input 1

1 1

Output for the Sample Input 1

0.941

Sample Input 2

1 2

Output for the Sample Input 2

0.275

Sample Input 3

1 9

Output for the Sample Input 3

0.000

Sample Input 4

10 10

Output for the Sample Input 4

1.000

Problem C

용감한 용사 진수

Time Limit: 1 Second

N명의 적 병사가 있다. 적의 각 병사는 힘, 민첩, 지능의 3가지 능력치를 가진다. 용감한 용사 진수도 힘, 민첩, 지능의 3가지 능력치를 가진다.

적의 각 병사에 대해,

1. 적 병사가 가진 힘보다 진수의 힘이 크거나 같고,
2. 적 병사가 가진 민첩보다 진수의 민첩이 크거나 같고,
3. 적 병사가 가진 지능보다 진수의 지능이 크거나 같으면,
진수는 그 적 병사를 이길 수 있다.

용감한 용사 진수에게 스태트 포인트를 주면 똑똑한 진수는 자기가 최대한 많은 적을 이길 수 있도록 스태트 포인트를 스스로 분배한다.

N명의 병사들 스태트 포인트가 주어졌을 때, 진수가 적어도 K명의 병사를 이길 수 있게 하는 최소의 스태트 포인트를 구하여라.

Input

첫 번째 줄에는 N명의 병사 수와 용감한 용사 진수가 이겨야 할 K명의 병사 수가 주어진다.
($1 \leq K \leq N \leq 100$)

두 번째 줄부터 N+1 번째 줄까지 각 줄마다 병사들의 힘, 민첩, 지능을 세 개의 음이 아닌 정수로 주어진다. ($0 \leq \text{힘, 민첩, 지능} \leq 1000000$)

Output

용감한 용사 진수가 적어도 K명의 병사를 이길 수 있게 하는 최소의 스태트 포인트를 출력하여라.

Sample Input 1

3 3
10 5 5
5 10 5
5 5 10

Output for the Sample Input 1

30

Sample Input 2

3 1
234 23 342
35 4634 34
46334 6 789

Output for the Sample Input 2

599

Sample Input 3

3 2
30 30 30
10 500 10
50 10 50

Output for the Sample Input 3

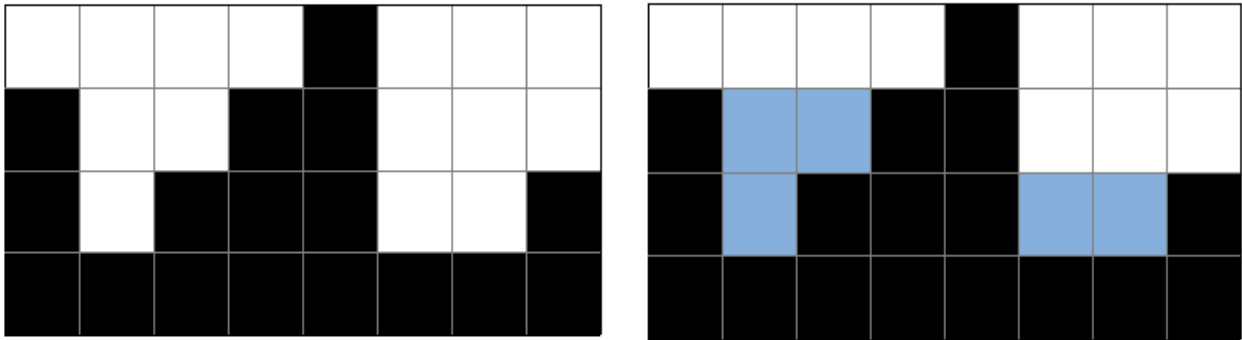
130

Problem D

빗물

Time Limit: 1 Second

2차원 세계에 블록이 쌓여있다. 비가 오면 블록 사이에 빗물이 고인다.



비는 충분히 많이 온다. 고이는 빗물의 총량은 얼마일까?

Input

첫 번째 줄에는 2차원 세계의 세로 길이 H와 2차원 세계의 가로 길이 W가 주어진다.

($1 \leq H, W \leq 500$)

두 번째 줄에는 블록이 쌓인 높이를 의미하는 0이상 H이하의 정수가 2차원 세계의 맨 왼쪽 위치부터 차례대로 W개 주어진다. 따라서 블록 내부의 빈 공간이 생길 수 없다. 또 2차원 세계의 바닥은 항상 막혀있다고 가정하여도 좋다.

Output

2차원 세계에서는 한 칸의 용량은 1이다. 고이는 빗물의 총량을 출력하여라.

빗물이 전혀 고이지 않을 경우 0을 출력하여라.

Sample Input 1

```
4 4
3 0 1 4
```

Output for the Sample Input 1

```
5
```

Sample Input 2

```
4 8
3 1 2 3 4 1 1 2
```

Output for the Sample Input 2

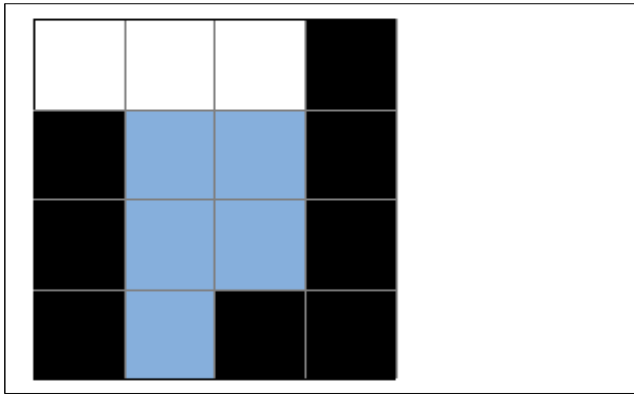
```
5
```

Sample Input 3

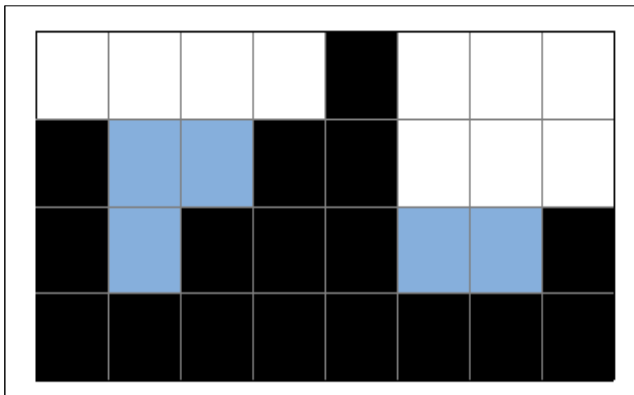
Output for the Sample Input 3

3 5 0 0 0 2 0	0
------------------	---

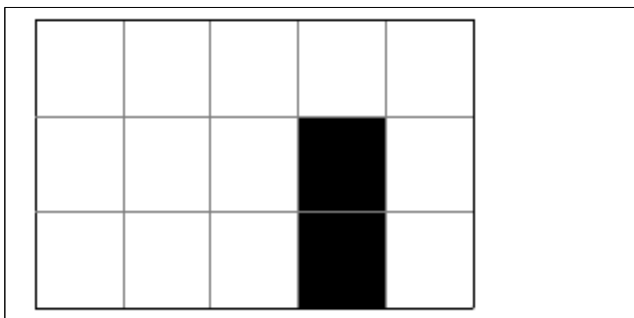
Hint 1



Hint 2



Hint3



Problem E

우유 축제

Time Limit: 1 Second

영학이는 딸기우유, 초코우유, 바나나우유를 좋아한다.

입맛이 매우 까다로운 영학이는 자신만의 우유를 마시는 규칙이 있다.

1. 맨 처음에는 딸기우유를 한 팩 마신다.
2. 딸기우유를 한 팩 마신 후에는 초코우유를 한 팩 마신다.
3. 초코우유를 한 팩 마신 후에는 바나나우유를 한 팩 마신다.
4. 바나나우유를 한 팩 마신 후에는 딸기우유를 한 팩 마신다.

영학이는 우유 축제가 열리고 있는 우유거리에 왔다. 우유 거리에는 우유 가게들이 일렬로 늘어서 있다. 영학이는 우유 거리의 시작부터 끝까지 걸으면서 우유를 사먹고자 한다. 각각의 우유 가게는 딸기, 초코, 바나나 중 한 종류의 우유만을 취급한다. 각각의 우유 가게 앞에서, 영학이는 우유를 사마시거나, 사마시지 않는다. 우유거리에는 사람이 많기 때문에 한 번 지나친 우유 가게에는 다시 갈 수 없다. 영학이가 마실 수 있는 우유의 최대 개수를 구하여라.

Input

첫째 줄에 우유 가게의 수 N 이 주어진다. ($1 \leq N \leq 1000$)

둘째 줄에는 우유 가게 정보가 우유 거리의 시작부터 끝까지 순서대로 N 개의 정수로 주어진다.

0은 딸기우유만을 파는 가게, 1은 초코우유만을 파는 가게, 2는 바나나우유만을 파는 가게를 뜻하며, 0, 1, 2 외의 정수는 주어지지 않는다.

Output

영학이가 마실 수 있는 우유의 최대 개수를 출력하시오.

Sample Input 1

```
7  
0 1 2 0 1 2 0
```

Output for the Sample Input 1

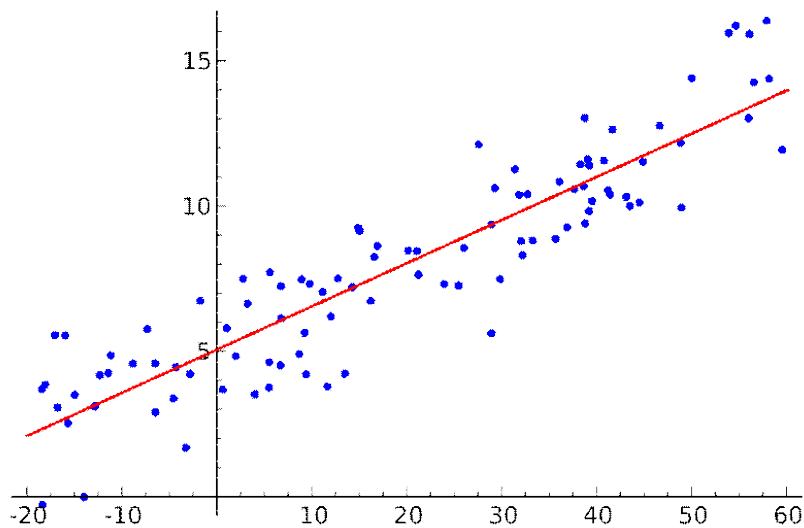
```
7
```


Problem F 성적표

Time Limit: 1 Second

동하는 이번 학기에도 F 학점을 받아 학사경고를 받았다. 더 이상 학사경고를 받지 않기 위해 동하는 공부를 하기로 결심했다. 동하는 공부를 얼마나 하면 얼마큼 성적이 나오는지 궁금했다. 그래서 동하는 N명의 학생들에 대해 공부시간과 그에 따른 시험점수를 조사했다.

동하는 i 번째 학생은 x_i 시간의 공부를 했고, y_i 점의 점수를 받았다는 사실을 알았다. 이 데이터로부터 공부를 x 시간 했으면 시험 성적은 $f(x) = ax + b$ 일 것이라고 추정하려 한다. 동하를 위해 a 와 b 를 구해주자.



공부시간에 따른 시험성적을 예측하는 함수 $f(x)$ 는, 데이터 (x_i, y_i) 에 대해, $f(x_i)$ 와 y_i 의 차가 작아야 할 것이다. ($i = 1, 2, \dots, N$)

이 개념을 수학적으로 형식화한 것이 RSS (Residual Sum of Squares)이다. RSS 가 작으면 예측모형이 데이터에 잘 맞는 것이라고 할 수 있다.

$$RSS = \sum_{i=1}^n (y_i - f(x_i))^2$$

100이하의 모든 양의 정수 a 와 b 중, RSS 를 최소화하는 a 와 b 를 구하여라.
오버플로우에 주의하여라.

Input

첫 번째 줄에는 데이터의 수 N 이 주어진다. ($2 \leq N \leq 100$)

두 번째 줄부터 $N+1$ 번째 줄까지, 각 줄에는 i 번 학생의 공부시간 x_i 와 시험점수 y_i 가 순서대로 주어진다. ($1 \leq x_i, y_i \leq 1000$, x_i 와 y_i 는 정수)

Output

RSS 가 최소인 a 와 b 를 한 줄에 공백으로 구분하여 출력하여라.

RSS 가 최소인 쌍 (a, b) 가 여러 개인 경우는 입력으로 주어지지 않는다.

Sample Input 1**Output for the Sample Input 1**

4	1 1
1 2	
2 3	
3 4	
4 5	

Problem G

우유 도시

Time Limit: 1 Second

영학이는 딸기우유, 초코우유, 바나나우유를 좋아한다.

입맛이 매우 까다로운 영학이는 자신만의 우유를 마시는 규칙이 있다.

1. 맨 처음에는 딸기우유를 한 팩 마신다.
2. 딸기우유를 한 팩 마신 후에는 초코우유를 한 팩 마신다.
3. 초코우유를 한 팩 마신 후에는 바나나우유를 한 팩 마신다.
4. 바나나우유를 한 팩 마신 후에는 딸기우유를 한 팩 마신다.

저번 축제에서 수많은 우유를 마셨지만 더욱 우유에 갈증을 느낀 영학이는 우유 여행을 떠났다. 맛있는 우유를 찾아 떠난 영학이는 수많은 우유 가게로 둘러 쌓인 어느 도시에 도착했다. 이 도시는 정사각형 형태의 2차원 격자 모양으로 남북으로 N개, 동서로 N개, 총 $N*N$ 개의 우유 가게들이 있다. 영학이는 도시의 서북쪽 끝 (1, 1)에서 출발해서 동남쪽 아래 (N, N)까지 가면서 우유를 사 마신다.

각각의 우유 가게는 딸기, 초코, 바나나 중 한 종류의 우유만을 취급한다. 각각의 우유 가게 앞에서, 영학이는 우유를 사 마시거나, 사 마시지 않는다. So coooooo~한 영학이는 오직 동쪽 또는 남쪽으로만 움직이기 때문에 한 번 지나친 우유 가게에는 다시 가지 않는다.

영학이가 마실 수 있는 우유의 최대 개수를 구하여라.

Input

첫번째 줄에는 우유 도시의 영역 크기 N이 주어진다. ($1 \leq N \leq 1000$)
두번째 줄부터 N+1 번째 줄까지 우유 도시의 정보가 주어진다.
0은 딸기우유만을 파는 가게, 1은 초코우유만을 파는 가게, 2는 바나나우유만을 파는 가게를 뜻하며, 0, 1, 2 외의 정수는 주어지지 않는다.

Output

영학이가 마실 수 있는 우유의 최대 개수를 출력하시오.

Sample Input 1

```
4
0 1 2 2
1 1 1 1
2 2 2 2
0 0 1 0
```

Output for the Sample Input 1

```
5
```

Sample Input 2

```
5
0 1 0 0 0
1 2 1 1 1
2 0 1 2 0
0 0 0 0 1
1 1 1 1 2
```

Output for the Sample Input 2

```
9
```

Problem H

이산수학 과제

Time Limit: 1 Second

민호는 이산수학 강의를 듣는다.

어느 날 교수님께서 Positive rational numbers are countable에 대해 증명해 주시고, 역시나 과제를 내주셨다.

양의 유리수는 다음 그림처럼 열거할 수 있다.

$\frac{1}{1}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$...
$\frac{2}{1}$	$\frac{2}{2}$	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{2}{5}$...
$\frac{3}{1}$	$\frac{3}{2}$	$\frac{3}{3}$	$\frac{3}{4}$	$\frac{3}{5}$...
$\frac{4}{1}$	$\frac{4}{2}$	$\frac{4}{3}$	$\frac{4}{4}$	$\frac{4}{5}$...
$\frac{5}{1}$	$\frac{5}{2}$	$\frac{5}{3}$	$\frac{5}{4}$	$\frac{5}{5}$...
⋮	⋮	⋮	⋮	⋮	⋮

첫 번째 유리수는 1/1, 두 번째 유리수는 2/1, 세 번째 유리수는 1/2, 네 번째 유리수는 3/1, 다섯 번째 유리수는 2/2, ... 이다.

1/1, 2/2, 3/3, ... 은 다르게 취급 하는것에 유의하여야 한다.

위 그림처럼 모든 유리수에 순차적으로 번호를 붙였을 때, N번째 유리수를 구하여야.

과제가 하기 싫은 민호는 컴공과답게 N번째 유리수를 구하는 프로그램을 만들려고 한다.

Input

첫 번째 줄에 양의 정수 N 이 주어진다. ($1 \leq N \leq 1000$)

Output

N 번째 유리수가 a/b 일 때, 분자 a , 분모 b 를 공백으로 구분하여 a , b 를 출력하여라.

Sample Input 1

1

Output for the Sample Input 1

1 1

Sample Input 2

2

Output for the Sample Input 2

2 1

Sample Input 3

5

Output for the Sample Input 3

2 2

Problem I

관리자는 누구?

Time Limit: 1 Second

BIG 프로젝트로 동문 커뮤니티 웹사이트를 만들려고 한다. 동문 커뮤니티의 관리자는 홈페이지와 선후배 네트워크를 담당하는 일을 한다. 동문 커뮤니티 제작 프로젝트를 맡으신 Mr.Nam 교수님은 동문 커뮤니티 운영을 맡을 중요한 관리자를 뽑아달라는 부탁을 동아리에 전달했다.

회의 결과 다음과 같은 규칙으로 매년 각 동아리에서 관리자를 선출하자는 의견이 나왔다.

1. 각 동아리에서는 동아리원 N명을 선출하여, 그 중 백준 온라인 저지 알고리즘 문제를 푼 개수가 가장 많은 사람을 그 동아리의 관리자 후보로 선출한다.
2. 각 동아리에서 뽑힌 후보들 중 가장 문제를 많이 푼 후보가 최종적으로 관리자가 된다.

충남대학교 컴퓨터공학과는 다양한 분야의 동아리들이 활동하고 있다.

17년 기준으로 PROBRAIN, GROW, ARGOS, ADMIN, ANT, MOTION, SPG, COMON, ALMIGHTY 이렇게 총 9개의 동아리가 있고,

각 동아리에는 최소 N명의 동아리원이 있을 때, 동문 커뮤니티 관리자는 어느 동아리에서 선출될 것인지 알아내어라.

Input

첫 번째 줄은 동아리원의 수 N이 주어진다. ($1 \leq N \leq 100$)

두 번째 줄부터 마지막 줄 까지 각 동아리별로 N명의 동아리원들의 백준 온라인 저지 알고리즘 문제를 푼 개수가 순서대로 주어진다.

같은 동아리의 동아리원들은 문제를 푼 개수가 같을 수 있지만, 각 동아리에서 선출된 후보들은 문제 푼 개수가 서로 다르다.

각 행은 같은 동아리원으로 구성되어 있으며, 정확히 다음과 같은 동아리명의 순서로 주어진다.

PROBRAIN, GROW, ARGOS, ADMIN, ANT, MOTION, SPG, COMON, ALMIGHTY

각 동아리원들이 문제를 푼 개수를 K라 할 때 ($1 \leq K \leq 1000$) 이다.

Output

PROBRAIN, GROW, ARGOS, ADMIN, ANT, MOTION, SPG, COMON, ALMIGHTY중

동문 커뮤니티 관리자로 선발되는 후보가 소속한 동아리의 이름을 대문자로 출력한다.

Sample Input 1**Output for the Sample Input 1**

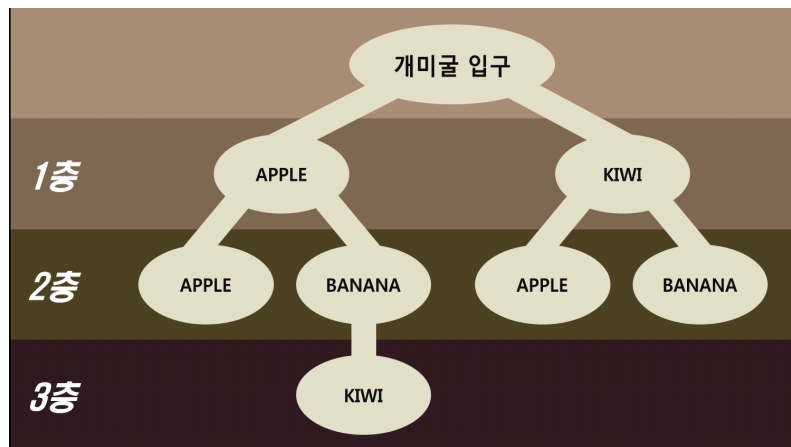
5	ANT
100 200 300 400 500	
300 300 300 300 300	
600 500 400 300 200	
400 400 400 400 400	
100 200 300 400 677	
111 222 333 444 555	
666 555 444 333 222	
123 234 345 456 567	
654 543 432 321 210	

Problem J 개미굴

Time Limit: 1 Second

개미는(똥똥) 오늘도(똥똥) 열심히(똥똥) 일을 하네.
개미는 아무말도 하지 않지만 땀을 뻘뻘 흘리면서 매일 매일을 살길 위해서 열심히 일을 하네.
한 치 앞도(똥똥) 모르는(똥똥) 험한 이 세상(똥똥) 그렇지만(똥똥) 오늘도 행복한 개미들!

우리의 천재 공학자 윤수는 이 개미들이 왜 행복한지 궁금해졌다. 행복의 비결이 개미가 사는 개미굴에 있다고 생각한 윤수는 개미굴의 구조를 알아보기 위해 로봇 개미를 만들었다.
로봇 개미는 센서가 있어 개미굴의 각 층에 먹이가 있는 방을 따라 내려가다 더 이상 내려갈 수 없으면 그 자리에서 움직이지 않고 신호를 보낸다. 이 신호로 로봇 개미는 개미굴 각 층을 따라 내려오면서 알게 된 각 방에 저장된 먹이 정보를 윤수한테 알려줄 수 있다.



로봇 개미 개발을 완료한 윤수는 개미굴 탐사를 앞두고 로봇 개미를 테스트 해보기 위해 위 그림의 개미굴에 로봇 개미를 투입했다. (로봇 개미의 수는 각 개미굴의 저장소를 모두 확인할 수 있을 만큼 넣는다.)

다음은 로봇 개미들이 윤수에게 보내준 정보다.

KIWI BANANA

KIWI APPLE

APPLE APPLE

APPLE BANANA KIWI

(공백을 기준으로 왼쪽부터 순서대로 로봇 개미가 각 층마다 지나온 방에 있는 먹이 이름을 뜻한다.)

윤수는 로봇 개미들이 보내준 정보를 바탕으로 다음과 같이 개미굴의 구조를 손으로 그려봤다.

APPLE

--APPLE

--BANANA

----KIWI

KIWI

--APPLE

--BANANA

(개미굴의 각 층은 "--" 로 구분을 하였다.

또 같은 층에 여러 개의 방이 있을 때에는 사전 순서가 앞서는 먹이 정보가 먼저 나온다.)

윤수는 개미굴을 연구하면서 개미들이 먹이를 저장할 때 같은 입구로 들어왔다면 같은 층에 있는 방에는 반드시 서로 다른 먹이를 저장하지만, 서로 다른 층에는 같은 먹이가 있을 수 있다는 것을 알아내었다.

(위 그림에서 2층에 같은 먹이가 있는 것처럼 보이지만 개미굴의 입구가 다르므로 이 역시 가능하다.)

우리의 천재 공학자 윤수는 복잡한 개미굴들을 일일이 손으로 그리기 힘들어 우리에게 그려달라고 부탁했다.

한치 앞도 모르는 험한 이세상 그렇지만 오늘도 행복한 개미들!

행복의 비결을 알기 위해 윤수를 도와 개미굴이 어떤 구조인지 확인해보자.

Input

첫 번째 줄은 로봇 개미가 각 층을 따라 내려오면서 알게 된 먹이의 정보 개수 N 개가 주어진다. ($1 \leq N \leq 1000$)

두 번째 줄부터 $N+1$ 번째 줄까지, 각 줄의 시작은 로봇 개미 한마리가 보내준 먹이 정보 개수 K 가 주어진다. ($1 \leq K \leq 15$)

다음 K 개의 입력은 로봇 개미가 왼쪽부터 순서대로 각 층마다 지나온 방에 있는 먹이 정보이며 각 먹이의 이름 길이 t 는 ($1 \leq t \leq 10$)이다.

Output

개미굴의 시각화된 구조를 출력하여라.

개미굴의 각 층을 "--" 로 구분하며, 같은 층에 여러개의 방이 있을 때에는 사전 순서가 앞서는 먹이 정보가 먼저 나온다.

Sample Input 1

```
4
2 KIWI BANANA
2 KIWI APPLE
2 APPLE APPLE
3 APPLE BANANA KIWI
```

Output for the Sample Input 1

```
APPLE
--APPLE
--BANANA
----KIWI
KIWI
--APPLE
--BANANA
```