

2016
서울대학교 프로그래밍 경시대회

주최 및 주관



후원



2016년 9월 11일

참가자를 위한 도움말

주의 사항

- 대회 시간은 13:00부터 18:00까지입니다. 대회가 진행되는 동안 인터넷 검색 및 전자기기 사용 등을 하실 수 없습니다. 단, 아래의 문서에 한해 대회 진행 중에도 참고하실 수 있으며, 책과 노트 등을 가져오신 경우 역시 참고하실 수 있습니다.
 - STL documentation: <http://www.sgi.com/tech/stl/>
 - JDK documentation: <http://docs.oracle.com/javase/8/docs/api/>
 - C++ reference: <http://en.cppreference.com/w/cpp>
- 대회는 Baekjoon Online Judge(<https://www.acmicpc.net>)를 이용하여 진행됩니다. 별도로 제공되는 계정 정보를 이용하여 로그인하신 뒤 코드 제출 및 결과 확인 등을 하실 수 있습니다.
- 모든 입력은 표준 입력으로 주어지며, 모든 출력은 표준 출력으로 합니다.
- 테스트 케이스가 존재하는 문제의 경우, 테스트 케이스에 대한 출력을 모아서 하실 필요 없이, 각 테스트 케이스를 처리할 때마다 출력해도 괜찮습니다.
- 중요!! 리턴 코드와 표준 오류(standard error, `stderr`) 스트림 출력에 주의하십시오. 프로세스가 0 이 아닌 리턴 코드를 되돌리는 경우나 표준 오류 스트림에 출력을 하는 경우 “런타임 에러”를 받게 됩니다.
- 문제에 대한 질의 사항은 대회 페이지의 질문 기능을 사용해 주시기 바랍니다. 이 때 대답해 드리기 어려운 질문에 대해서는 “답변을 드릴 수 없습니다”로 대답될 수 있으므로 유의하십시오.

채점 결과에 대하여

맞았습니다!! 제출하신 답안이 모든 테스트 데이터를 정해진 시간 안에 통과하여 정답으로 인정되었음을 의미합니다.

컴파일 에러 제출하신 답안 프로그램을 컴파일하는 도중 오류가 발생하였음을 의미합니다.

런타임 에러 제출하신 답안 프로그램을 실행하는 도중 프로세스가 비정상적으로 종료되었음을 의미합니다.

시간 초과 제출하신 답안 프로그램이 정해진 시간 안에 종료되지 않았음을 의미합니다.

틀렸습니다 제출하신 답안 프로그램이 테스트 데이터에 대해 생성한 출력이 출제자의 정답과 일치하지 않음을 의미합니다.

출력 초과 제출하신 답안 프로그램이 지나치게 많은 양의 출력물을 생성하여 강제로 종료되었음을 의미합니다.

출력 형식이 잘못되었습니다 제출하신 답안 프로그램이 정해진 출력 형식을 따르지 않았음을 의미합니다.

만약 여러 가지의 원인으로 인해 “맞았습니다!!” 가 아닌 다른 결과를 얻으셨다면, 그 중 어떤 것도 결과가 될 수 있습니다. 예를 들어 답도 잘못되었고 출력 형식도 잘못된 코드를 제출하신 경우 대부분 “출력 형식이 잘못되었습니다”를 받으시게 되지만, 경우에 따라서 “틀렸습니다”를 받을 수도 있습니다.

Problem A. 치킨 먹고 싶다



서울대학교 301동에는 아는 사람만 아는 “눕치킨”이란 치킨집이 있다. 이 치킨집은 여느 치킨집처럼 치킨을 시킬 때마다 쿠폰을 C 장을 주고, 쿠폰을 F 장 모으면 치킨을 공짜로 시킬 수 있다.

눕치킨의 단골이 아닌 두영이에게는 쿠폰으로 시키는 치킨에는 쿠폰이 떨려나오지 않는다. 하지만 놓치킨의 단골 손님인 상언이에게는 치킨집 아저씨가 쿠폰으로 시키는 치킨에 쿠폰을 주신다. 상언이와 두영이는 둘 다 M 원을 가지고 있고, 치킨의 가격은 P 원이다. 이 때, 상언이는 두영이보다 치킨을 얼마나 더 시켜먹을 수 있을까?

Input

첫 번째 줄에 테스트 케이스의 수 T ($1 \leq T \leq 20,000$) 가 주어지고, 이어서 T 개의 테스트 케이스가 주어진다.

각 테스트 케이스마다 한 줄에 4개의 정수가 주어진다. 이는 순서대로 치킨의 가격 P ($1 \leq P \leq 50,000$), 치킨에 쓸 돈 M ($1 \leq M \leq 1,000,000$), 치킨을 공짜로 시키는데 필요한 쿠폰의 장수 F ($2 \leq F \leq 1,000$), 치킨을 시키면 주는 쿠폰의 장수 C ($1 \leq C < F$) 를 의미한다.

Output

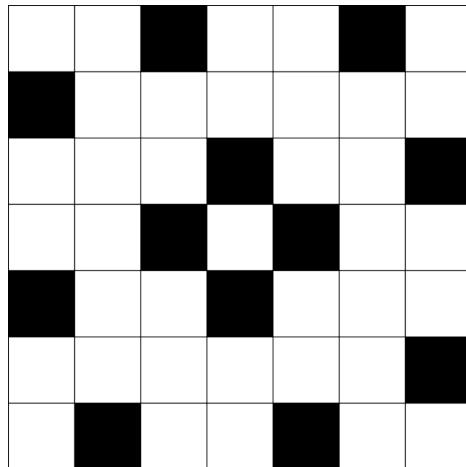
각 테스트 케이스마다, 첫 번째 줄에 상언이가 두영이보다 더 먹을 수 있는 치킨의 수를 출력한다.

Sample input and output

standard input	standard output
2	0
10000 50000 5 1	1
10000 250000 5 1	

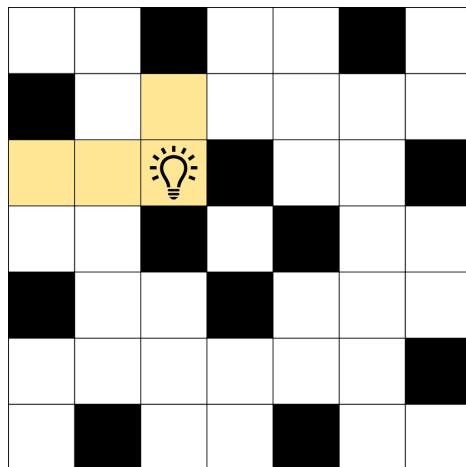
Problem B. Light Up

Light Up(<http://www.puzzle-light-up.com/>)은 온라인으로 할 수 있는 퍼즐 게임이다. 이 게임의 규칙은 간단하다. $N \times N$ 크기의 격자판이 주어지는데, 이 격자판의 각 칸은 검정색 정사각형 혹은 흰색 정사각형으로 구성되어 있다. 이 게임의 목표는 흰색 정사각형에 백열 전구를 아주 잘 놓아, 모든 흰색 정사각형에 불이 들어오게 하는 것이다. 만약 어떤 흰색 정사각형과 같은 세로줄 혹은 같은 가로줄에 백열 전구가 놓여 있고, 사이에 검정색 정사각형이 없다면 그 흰색 정사각형은 불이 켜진 상태가 된다.



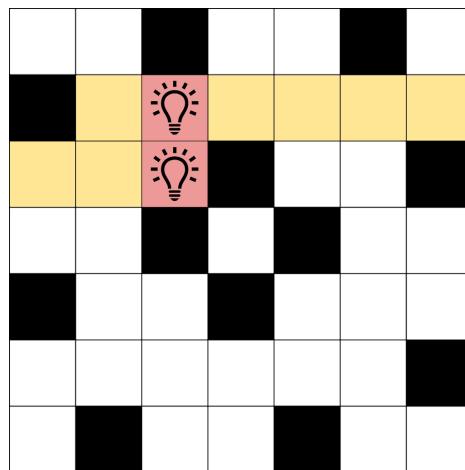
〈그림 1〉

그림 1과 같은 격자판이 주어졌을 때, (3, 3)의 위치에 백열 전구를 배치하면 그림 2와 같은 상황이 된다.



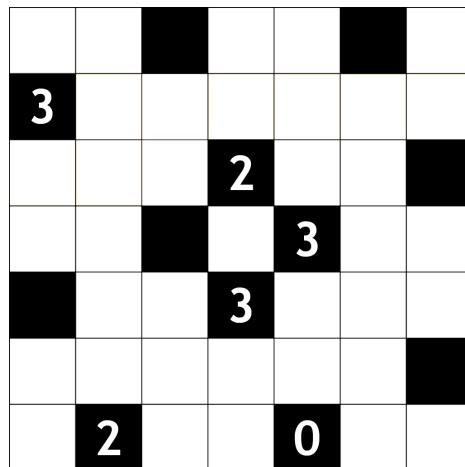
〈그림 2〉

그런데 이미 불이 켜진 흰색 정사각형에 다른 백열 전구를 놓으면, 백열 전구가 과열되기 때문에 배치를 할 수 없게 된다. 즉, 그림 3 같이 (2, 3)과 (3, 3)에 백열 전구를 배치하는 것은 불가능하다.



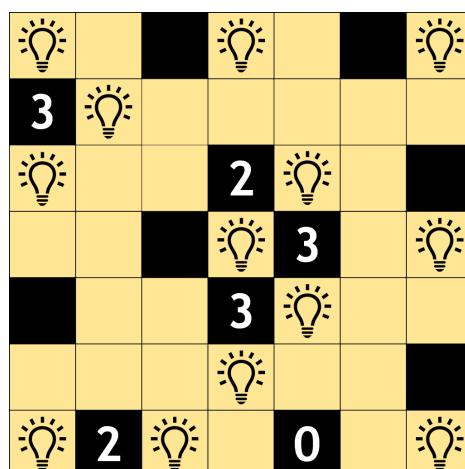
〈그림 3〉

몇몇 검은 정사각형에는 숫자가 쓰여 있는데, 이는 그 정사각형과 변을 공유하는 네 개의 정사각형 중 백열 전구가 놓여있어야 하는 정사각형의 개수를 의미한다. 그림 4와 같은 상황을 살펴보자.



〈그림 4〉

그림 4와 같은 판이 주어졌을 때, 백열 전구를 잘 배치하는 방법은 그림 5와 같다.



〈그림 5〉

임의의 격자판이 주어졌을 때, 퍼즐을 해결하는 방법을 알아내자.

Input

첫 번째 줄에 테스트 케이스의 수 T 가 주어진다. ($1 \leq T \leq 30$)

테스트 케이스의 첫 번째 줄에 격자판의 크기 N 이 주어진다. ($1 \leq N \leq 7$)

테스트 케이스의 두 번째 줄부터 N 개의 줄에 걸쳐 N 개의 숫자가 주어진다. i 번째 줄의 j 번째 숫자는 격자판의 (i, j) 위치의 정사각형에 대한 정보 R_{ij} 이다. R_{ij} 가 -2 인 경우 흰색 정사각형, R_{ij} 가 -1 인 경우 검은색 정사각형, 0 이상 4 이하인 경우 그 숫자가 적힌 검은색 정사각형이다.

Output

N 개의 줄에 걸쳐 각 줄에 0 또는 1 인 N 개의 숫자를 출력한다. 퍼즐을 해결할 수 있도록 백열 전구를 배치한 후, 백열 전구가 배치된 칸은 1 , 그렇지 않은 칸은 0 으로 표현하여 출력한다.

답이 항상 존재하는 입력만 주어지고, 두 개 이상의 답이 존재하는 경우 그 중 하나만을 출력한다.

Sample input and output

standard input	standard output
2	1 0 0 0 0 0 0
7	0 0 0 1 0 0 0
-2 -2 -2 -2 -2 0 -2	0 1 0 0 0 1 0
1 -2 -2 -2 -2 -2 -2	0 0 0 0 1 0 0
-2 -2 1 -2 2 -2 -2	0 0 0 0 0 0 1
-2 -2 -2 -2 -2 -2 -2	0 0 0 0 1 0 0
-2 -2 0 -2 2 -2 -2	1 0 0 0 0 0 1
-2 -2 -2 -2 -2 -2 2	1 0 0 1 0 0 1
-2 1 -2 -2 -2 -2 -2	0 1 0 0 0 0 0
7	1 0 0 0 1 0 0
-2 -2 -1 -2 -2 -1 -2	0 0 0 1 0 0 1
3 -2 -2 -2 -2 -2 -2	0 0 0 0 1 0 0
-2 -2 -2 2 -2 -2 -1	0 0 0 1 0 0 0
-2 -2 -1 -2 3 -2 -2	1 0 1 0 0 0 1
-1 -2 -2 3 -2 -2 -2	
-2 -2 -2 -2 -2 -2 -1	
-2 2 -2 -2 0 -2 -2	

Problem C. 즉흥 여행

당신은 세계 각부 순위에서 찾아볼 수 있는 대부호인 상일이의 비서이다. 당신의 상사인 상일이는 여행을 즉흥적으로 가는 것을 즐긴다. 상일이가 즉흥적으로 여행을 가는 방식은 다음과 같다.

1. 자신이 있는 도시의 공항에 간다.
2. 그 공항에서 출발할 수 있는 항공편 중 임의의 항공편 티켓을 구입한다. 이 때, 각각의 항공편의 티켓을 구입하는 확률은 모두 동일하다.
3. 구입한 항공편을 타고 다른 도시로 이동한다.

이제, 위의 과정을 한 번의 여정이라고 정의하자. 상일이는 한 번의 여정으로 여행을 했다고 만족하지 못하는 성격이기 때문에, 한 번의 여행에 정확히 K 번의 여정을 해야 한다. 들어갈 때는 마음대로지만 나올 때는 아닌 공항도 있다. 이 경우 더 이상 여정을 진행하지 못하여, 상황에 따라 상일이는 영원한 여행을 하게 될 수도 있다.

상일이는 여행 중에 업무와 관련한 이야기를 듣기 싫어하기 때문에, 당신은 어쩔 수 없이 휴가를 얻게 되었다. 하지만, 마냥 휴가를 즐기고 있을 당신이 아니다. 당신은 상일이가 갈 수 있는 여행 코스 중 가장 확률이 높은 여행 코스를 따라 미행하기로 결심하였다. 또한 여행이 끝남과 동시에 업무로 다시 돌아가야 하므로, 그 여행 코스의 최종 목적지 공항 근처에 업무 관련 자료를 미리 보내려 한다.

상일이가 갈 수 있는 여행 코스 중 가장 확률이 높은 여행 코스의 최종 공항을 알아내자.

Input

첫 번째 줄에 테스트 케이스의 수 T 가 주어진다.

각 테스트 케이스의 첫 번째 줄에 전 세계 공항의 개수 N 과 한 여행에 해야 하는 여정의 수 K 가 주어진다.
($2 \leq N \leq 100, 1 \leq K \leq 1,000$)

각 테스트 케이스의 두 번째 줄부터 N 개의 줄에 걸쳐 각 공항의 IATA 공항 코드가 주어진다. 이 공항 코드는 3자리 알파벳 대문자로 이루어진다. 여행을 출발할 때 가는 공항의 IATA 공항 코드는 ICN이다.

각 테스트 케이스의 그 다음 줄부터 N 개의 줄에 걸쳐 각 줄에 N 개의 숫자가 주어진다. i 번째 줄의 j 번째 숫자는 i 번 공항에서 출발하여 j 번 공항에 도착하는 항공편의 개수 S_{ij} 이다. ($0 \leq S_{ij} \leq 100, S_{ii} = 0$)

Output

각 테스트 케이스마다, 첫 번째 줄에 상일이가 갈 수 있는 여행 코스 중 가장 확률이 높은 여행 코스의 최종 공항의 IATA 공항 코드를 출력한다.

Sample input and output

standard input	standard output
1 4 3 ICN FUK PEK FNJ 0 2 3 0 2 0 3 0 1 2 0 5 0 0 5 0	PEK

Notes

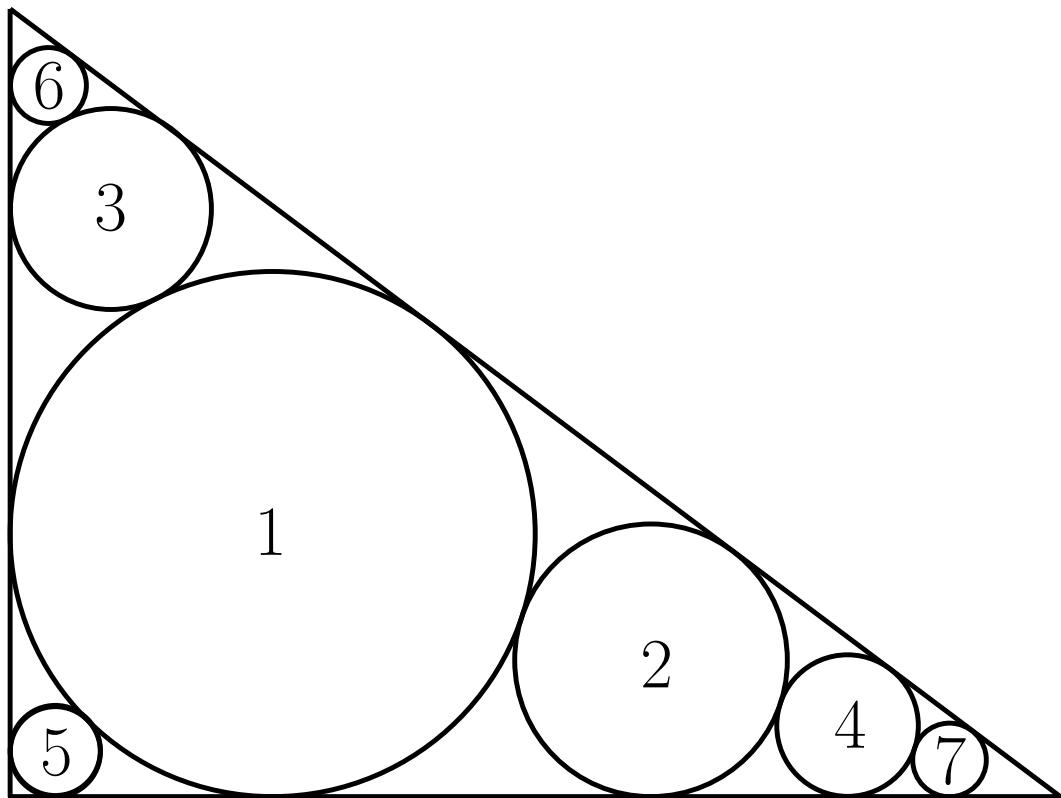
예제 입력에서, 아래 10가지 여행 경로가 있고, 각 확률은 다음과 같다.

- ICN → FUK → ICN → FUK : $0.4 \times 0.4 \times 0.4 = 0.064$
- ICN → FUK → ICN → PEK : $0.4 \times 0.4 \times 0.6 = 0.096$
- ICN → FUK → PEK → ICN : $0.4 \times 0.6 \times 0.125 = 0.03$
- ICN → FUK → PEK → FUK : $0.4 \times 0.6 \times 0.25 = 0.06$
- ICN → FUK → PEK → FNJ : $0.4 \times 0.6 \times 0.625 = 0.15$
- ICN → PEK → ICN → FUK : $0.6 \times 0.125 \times 0.4 = 0.03$
- ICN → PEK → ICN → PEK : $0.6 \times 0.125 \times 0.6 = 0.045$
- ICN → PEK → FUK → ICN : $0.6 \times 0.25 \times 0.4 = 0.06$
- ICN → PEK → FUK → PEK : $0.6 \times 0.25 \times 0.6 = 0.09$
- ICN → PEK → FNJ → PEK : $0.6 \times 0.625 \times 1 = 0.375$

Problem D. 피자 배치

상언이의 단골 피자가게는 직각삼각형 모양의 테이블과 완벽한 원 모양의 피자로 유명하다. 이 가게의 사장님은 한 테이블에서 피자를 주문하면 그 테이블로 직접와서 테이블이 피자로 가득 찰 때까지 피자를 계속 만든다.

사장님은 피자 하나를 새로 만들 때마다 다음과 같은 그만의 놀라운 방법을 사용한다. 현재 이미 만들어진 다른 피자들과 겹치지 않으면서 (겹할 수는 있다) 테이블의 변 중 두개 이상에 접하도록 만들 수 있는 피자 중 가장 넓이가 큰 피자를 만든다. 상언이는 문득 이 테이블에 만들어질 k 번째 피자의 넓이가 궁금해졌다.



Input

첫 번째 줄에 테스트 케이스의 수 T ($1 \leq T \leq 10,000$) 가 주어지고, 이어서 T 개의 테스트 케이스가 입력된다.

각 테스트 케이스는 한 줄에 3개의 정수가 주어진다. 이는 직각삼각형의 직교하는 두 변 a, b 의 길이와 k 를 의미한다. ($1 \leq a, b \leq 10^9$, $1 \leq k \leq 100$)

Output

각 테스트 케이스마다, 첫 번째 줄에 k 번째 피자의 넓이를 출력한다. 출제진의 답과 절대 오차 또는 상대 오차가 10^{-6} 이하일 시 정답으로 인정한다.

Sample input and output

standard input	standard output
1	3.1415926
3 4 1	

Problem E. 읽어내기

고대 유적을 발굴하던 성원이는 커다란 비석을 찾아내었고, 학계에 보고하기 위해 그 비석에 적힌 글을 해독하고 있다. 하지만 이 비석은 오랜 세월을 거치며 글자들이 지워져 잘 보이지 않게 되었다.

이 비석이 만들어질 당시의 사람들은 비석에 왕의 이름을 써 놓는 것을 좋아했다. 비석의 글자들 중 일부(연속하지 않아도 된다)를 골라 순서대로 조합하여 왕의 이름이 되는 경우의 수가 많을수록 더 좋아했다고 한다. 또 이 당시 사람들은 말을 길게 하는 것을 좋아하지 않아 왕의 이름은 항상 5글자 이내였다고 한다.

방사선 연대 측정을 통해 이 비석이 묻힐 당시의 왕의 이름을 알게 된 성원이는 이 비석에 왕의 이름을 읽어낼 수 있는 경우의 수가 몇 가지인지 알고 싶게 되었다. 그런데 비석의 글씨는 잘 보이지 않으므로 연구가 진행됨에 따라 비석의 일부의 해석이 달라질 수 있고, 그 때마다 이름을 읽어내는 경우의 수를 알고 싶어한다. 성원이를 도와 경우의 수를 계산하는 프로그램을 작성해 주자.

Input

첫 번째 줄에 테스트 케이스의 수 T 가 주어진다.

각 테스트 케이스의 첫 번째 줄에는 비석의 글자 수 $N(1 \leq N \leq 200,000)$ 과 왕의 이름의 길이 $M(1 \leq M \leq 5)$, 그리고 비석의 해석이 바뀌는 횟수 $Q(0 \leq Q \leq 100,000)$ 가 주어진다. 둘째 줄에는 비석에 적힌 글의 첫 번째 해석이 하나의 문자열로 주어진다. 이는 영어 알파벳 대문자로만 구성되어 있다. 셋째 줄에는 왕의 이름이 하나의 문자열로 주어지며, 마찬가지로 영어 알파벳 대문자로만 구성되어 있다. 넷째 줄부터 Q 개의 줄에는 한 줄마다 두 개의 정수 A_i, B_i 와 문자열 S_i 가 주어진다. ($1 \leq A_i \leq B_i \leq N, \text{len}(S_i) = B_i - A_i + 1$) 이는 비석의 A_i 번째 문자부터 B_i 번째 문자까지의 해석이 S_i 로 바뀐다는 뜻이다. S_i 의 길이 총합은 2,000,000 이하이다.

Output

각 테스트 케이스마다, $Q+1$ 개의 줄에 한 줄에 하나의 정수를 출력한다. i 번째 줄에는 비석의 i 번째 해석에서 왕의 이름을 읽을 수 있는 경우의 수를 출력한다. 답이 커질 수 있으므로 1,000,000,007로 나눈 나머지를 출력한다.

Sample input and output

standard input	standard output
<pre>1 20 5 1 MIDASMIDASMIDASMIDAS MIDAS 2 19 MMMIIIIIDDDDAAAASSSS</pre>	<pre>56 1024</pre>

Problem F. 표본의 수 구하기

몇몇 뉴스 기사들은 여론조사 결과를 참조하여 작성한다. 이 때, 가독성을 위해 실제 투표 수보다는 퍼센트로 결과를 나타내는 경우가 많은데, 이 경우 표본의 수가 얼마나 되는지 궁금해질 때가 문득 생긴다.

예 혹은 아니오로 선택 가능한 여론조사 질문에 대해, ‘예’에 해당하는 투표 수의 퍼센트가 주어질 때 가능한 최소 표본의 수를 알아내자. 이 때, 투표 결과는 항상 넷째 자리에서 반올림을 한다.

Input

첫 번째 줄에 테스트 케이스의 수 T 가 주어진다.

각 테스트 케이스의 첫 번째 줄에 투표 수의 퍼센트가 소수점 아래 셋째 자리까지 주어진다.

Output

각 테스트 케이스마다 한 줄에 가능한 최소 표본의 수를 출력한다.

Sample input and output

standard input	standard output
3	66667
0.001	2
50.000	20
15.000	

Problem G. 비밀번호

스포츠, 특히 프로야구에서 ‘비밀번호’ 라 함은 암흑기의 등수를 지칭하는 말이다. 예를 들어, 한국 프로야구의 롯데 자이언츠는 2001년부터 2007년까지 매년 8등, 8등, 8등, 8등, 5등, 7등, 7등을 순서대로 찍으며 ‘8-8-8-8-5-7-7’이라는 비밀번호를 작성하였다. 또한, 같은 리그의 LG 트윈스는 2003년부터 2012년까지 ‘6-6-6-8-5-8-7-6-6-7’이라는 비밀번호를 작성하였고, 한화 이글스는 2008년부터 현재까지 ‘5-8-8-6-8-9-9-6’이라는 비밀번호를 작성하고 있다.

비밀번호를 한국 프로야구 뿐만 아니라 모든 스포츠의 프로 리그에 확장시키기 위해, 다음과 같이 ‘비밀번호’를 정의하였다.

1. 비밀번호는 암흑기의 등수를 순서대로 나열한 다음, 그 등수 사이에 하이픈(‘-’)을 넣은 형태이다. 예를 들어, 8등, 11등, 7등, 6등을 순서대로 한 경우 ‘8-11-7-6’이 비밀번호가 된다.
2. 비밀번호에 등장하는 숫자는 (해당 리그의 팀 수)의 절반보다 큰 수여야 한다. 예를 들어, 8팀 혹은 9팀이 있는 리그에서는 5 이상의 숫자로만 이루어져야 한다.
3. 비밀번호의 길이는 L_1 이상 L_2 이하이다. 비밀번호의 후보가 여러 개가 될 수 있는데, 비밀번호를 읽을 때 임팩트가 강렬할 수 있도록 내림차순으로 가장 빠른 것을 선택한다. 예를 들어, 8팀이 있는 리그에서 8등, 7등, 2등, 8등, 7등, 7등을 했고, $L_1 = 2$ $L_2 = 3$ 인 경우, ‘8-7-7’이 비밀번호가 된다. 만약, 8등, 7등, 2등, 7등, 7등을 한 경우라면 ‘8-7’이 비밀번호가 된다.

K 팀이 있는 스포츠 리그가 있다. 이 스포츠 리그에 있는 팀의 N 년 동안 등수가 주어졌을 때, 이 팀의 비밀번호와 이 팀의 비밀번호가 반복된 횟수를 알아내자.

Input

첫 번째 줄에 테스트 케이스의 수 T 가 주어진다.

각 테스트 케이스의 첫 번째 줄에 네 개의 정수 K , N , L_1 , L_2 가 순서대로 공백으로 분리되어 주어진다.
($1 \leq K \leq 100,000$, $1 \leq N \leq 200,000$, $1 \leq L_1 \leq L_2 \leq N$)

각 테스트 케이스의 두 번째 줄에 이 팀의 N 년간 등수 정보인 N 개의 정수가 주어진다. i 번째에 주어지는 정수 R_i 는 i 번째 해의 등수를 의미한다. ($1 \leq R_i \leq K$)

Output

각 테스트 케이스마다 두 줄을 출력한다.

각 테스트 케이스에 대해 첫 번째 줄에 비밀번호를 출력하고, 두 번째 줄에 비밀번호가 등장하는 횟수를 출력한다.

Sample input and output

standard input	standard output
2	8-8-8-8
10 10 2 4	7
8 8 8 8 8 8 8 8 8	8-7-6-7-7
8 18 3 8	1
6 2 6 6 6 8 5 8 7 6 7 3 4 8 7 6 7 7	

Problem H. Professor KCM

KCM 교수는 수업 시간에 악랄한 질문을 하기로 유명하다. 어떤 학생에게 질문을 던졌을 때 대답하지 못하거나 틀린 답을 하면 그 학생의 성적에는 C, D가 빛발치게 된다. 오늘도 학생들을 매의 눈으로 바라보며 어떤 질문을 할지 고민 중이던 그는 갑자기 엄청나게 악랄한 질문을 떠올렸다.

우선 칠판에 수업 중이던 내용을 다 지우고는 임의의 자연수들을 미친듯이 막 적어대기 시작했다. 그러고는 학생들을 향해 이렇게 외쳤다.

“자, 이제 게임을 시작하지. 여러분은 힘을 모아서 내가 내는 질문의 답을 구해야 할거야. 힘을 모아 구한 답이 맞다면 나는 앞으로 수업 시간에 질문을 하지 않겠다. 하지만 틀린다면 여러분의 학점에 F가 빛발친다!”

학생들은 이번이 마지막 희망이라 생각하고 교수의 말을 주의깊게 들었다.

“내가 지금 칠판에 자연수들을 막 적었지? 여러분은 여기에 특정한 연산을 마음대로 여러번 수행할 수 있어. 이 연산이 뭐냐하면 특정한 두 숫자 x, y 를 고른 다음에 그 둘을 지워. 그리고 그 자리에 $\gcd(x, y)$ 와 $\text{lcm}(x, y)$ 를 적는거야. 이 연산을 반복하면 칠판에 적혀있는 숫자들이 막 바뀌겠지? 그리고나서 적당한 때가 되면 여러분은 그 때 남아있던 숫자들 중에 가장 큰 숫자를 골라 나에게 제출해야 해. 이 때 만들 수 있는 결과의 최대값은 무엇일까? 바로 이것이 질문이야... 하하...”

마침 그 자리에서 수업을 듣고 있던 doju는 강의실에서 도주하고 싶었지만 KCM 교수의 연구실 학생들이 내려와 강의실 문을 잠가버려 도주할 수 없었다. 이제 교수의 질문을 피할 수 없게 된 doju는 여러분에게 도움을 요청했다. 교수의 질문에 대한 답을 구해서 doju가 도주할 수 있게 해주자.

Input

첫 번째 줄에 테스트 케이스의 수 T 가 주어진다.

각 테스트 케이스는 두 줄로 구성되어 있다. 첫 번째 줄에 교수가 적은 자연수의 개수 N ($1 \leq N \leq 1,000,000$)이 주어진다. 두 번째 줄에 교수가 적은 자연수 N 개가 공백으로 구분되어 주어진다. 각 수는 1 이상 1,000 이하이다.

모든 테스트 케이스에서 N 의 합이 2,000,000을 넘지 않는다.

Output

각 테스트 케이스마다 한 줄씩 교수의 질문에 대한 답을 출력한다. 답이 너무 커질 수 있으므로 1,000,000,007로 나눈 나머지만을 출력한다.

Sample input and output

standard input	standard output
1	120
5	
1 30 3 20 8	

Notes

첫번째로 20과 3을 고른다. 연산을 수행하면 두 수는 1과 60으로 바뀐다.

두번째로 60과 8을 고른다. 연산을 수행하면 두 수는 4와 120으로 바뀐다.

이 상태에서 120을 골라 제출하면 된다. 이것이 만들 수 있는 수 중에 최대값이다.

Problem I. Torres del Paine

남아메리카 대륙의 끝 파타고니아 지방에는 아름다운 풍경으로 유명한 토레스 델 파이네 국립공원이 있다. 거대한 산봉우리, 빙하와 호수, 숲과 강이 어우러진 절경을 보러 해마다 수많은 사람들이 찾는다. 이 곳을 제대로 감상하기 위한 방법으로 토레스 델 파이네를 W 모양으로 가로지르는 W 트레킹 코스가 있는데 약 3박 4일 정도 소요된다. 남아메리카 여행을 갔으나 일정 문제로 이 곳에 3박 4일 동안 있을 수 없었던 브라이언이 2020년에 다시 한 번 가보겠다고 하니 관심 있는 사람은 같이 여행하는 것도 좋을 것 같다. (광고 아님)



토레스 델 파이네에는 깎아지른듯한 세 개의 화강암 봉우리가 나란히 있는데, 높이가 2,800m에 달하고 각각의 이름은 Torre Norte(북쪽 봉우리), Torre Central(가운데 봉우리), Torre Sur(남쪽 봉우리)이다. 이 세 개의 봉우리는 완전히 일직선으로 놓여있지 않기 때문에 보는 위치에 따라 세 개의 봉우리가 보이는 순서는 바뀔 수 있다.

토레스 델 파이네를 구경하던 브라이언은, 공원 안에서 봉우리를 현재와 같은 순서로 볼 수 있는 지점이 얼마나 되는지 궁금해졌다. 공원의 전체 영역과 세 봉우리의 위치가 주어졌을 때, 세 봉우리를 순서대로 볼 수 있는 영역의 넓이를 구하는 프로그램을 작성해보자.

Input

첫 번째 줄에 테스트 케이스의 수 T 가 주어진다.

각 테스트 케이스는 두 줄로 이루어져 있다. 첫 번째 줄에는 실수 네 개가 주어지며, 이는 공원 영역을 나타낸다. 공원 영역은 직사각형이며, 왼쪽 아래 점의 좌표 x_0, y_0 와 오른쪽 위 점의 좌표 x_1, y_1 이 주어진다. ($-10^9 \leq x_0, y_0, x_1, y_1 \leq 10^9$)

다음 줄에는 실수 여섯 개가 주어지며, 이는 세 개의 봉우리 A, B, C 의 좌표 $x_a, y_a, x_b, y_b, x_c, y_c$ 를 의미한다. 세 점은 모두 공원 영역 안에 위치해 있으며 (즉 $x_0 < x_a, x_b, x_c < x_1$ 를 만족하고 $y_0 < y_a, y_b, y_c < y_1$ 를 만족함) 세 점이 일직선상에 놓여있는 경우는 입력으로 주어지지 않는다.

Output

각 테스트 케이스마다 한 줄씩 공원 영역 안에서 봉우리가 순서대로 보이는 영역의 넓이를 출력한다. 출제진의 답과 절대 오차 또는 상대 오차가 10^{-6} 이하일 시 정답으로 인정한다. 어떤 점에서 봉우리가 순서대로 보인다는 것은 해당 지점에서 시계방향으로 A, B, C 가 순서대로 보임을 의미한다.

Sample input and output

standard input	standard output
3	7759.5238095238100868
0 0 100 100	35
50 90 60 40 30 20	745.48070523149726796
0 0 10 10	
8 8 8 9 9 8	
-100 -100 100 100	
-86 -80 -29 -98 54 1	

Notes

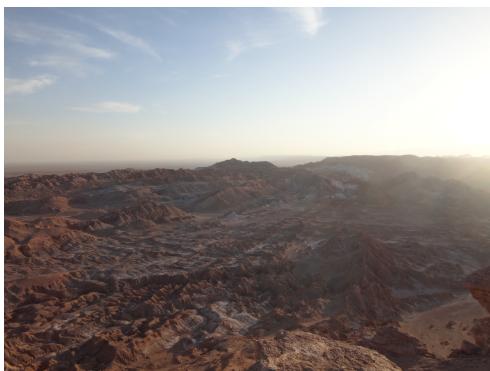
아래 사진은 문제 풀이와 관련이 없습니다. 심심할 때 보세요.



Cristo Redentor



Iguazu Falls



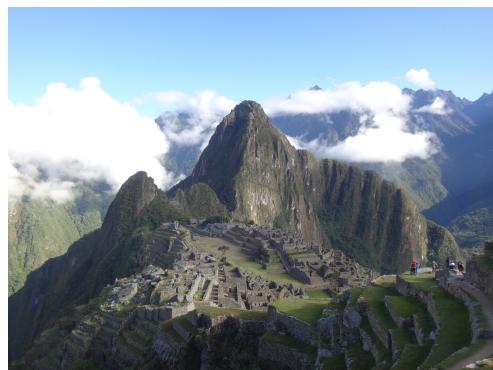
Atacama Desert



Perito Moreno Glacier



Salar de Uyuni



Machu Picchu

Problem J. 승현이와 승현이

석환나라에는 조승현이라는 이름을 가진 사람이 두 명이 있다. 헷갈리니까 한 명은 조승현13 이라 하고 한 명은 조승현16 이라 하자. 이 둘은 원래 서로의 존재를 모르고 있다가 얼마전에 뉴스를 보고 알게 되었다. 자신과 같은 이름을 가진 사람이 존재한다는 것이 너무 신기했던 그 둘은 연락처를 알아내서 서로 연락하는 사이가 되었다.

어느날 그 둘은 상대방이 사는 도시에 대해 궁금해졌다. 전화로 서로의 도시에 대해 설명해주다가 지친 그 둘은 결국 상대방의 도시에 여행을 가기로 결정했다.

석환나라는 N 개의 도시로 이루어져 있고 도시마다 번호가 1부터 N 까지 붙어있다. 그 사이에 도시들을 연결하는 도로 M 개가 있다. 하나의 도로는 도시 두개를 연결하고 그 사이를 양방향으로 다닐 수 있게 한다. 그리고 임의의 한 도시에서 출발하여 도로를 적당히 거쳐가면 다른 모든 도시에 도착할 수 있음이 보장되어있다.

현재 조승현13은 S 번 도시에 있고 조승현16은 E 번 도시에 살고있다. 이 둘은 상대방에게 자신의 도시로 가는 길을 설명해주기 위해 전화로 연락을 유지하면서 다음과 같은 방법으로 여행한다.

1. 0일차에 조승현13은 S 번 도시에, 조승현16은 E 번 도시에 있다.
2. i 일차 ($i \geq 0$)의 아침에 둘은 전화로 연락하여 오늘 누가 움직일지 결정한다. 하루에 둘 중 한 명만 움직일 수 있다.
3. 이동하기로 결정된 사람은 현재 위치한 도시에 연결된 도로 중 하나를 골라 그 도로를 따라 이동하고, 반대편 도시에 도착하면 이동을 마친다. 이 과정은 석양이 지기 전에 항상 완료된다.
4. i 일차의 석양이 진 후에 둘은 다시 전화로 연락하여 서로의 생사를 확인한다.
5. 생사를 확인한 직후에 조승현13이 E 번 도시에, 조승현16이 S 번 도시에 있는 상태라면 여행을 종료한다. 그렇지 않다면, 숙소에서 자고 일어난 후 2번 과정으로 돌아가 반복한다.

전화로 연락하기 위해서는 각자 갖고 있는 전화기가 일정 수준 이상의 무선 신호 출력을 낼 수 있어야 한다. 각 도시 i 마다 무선 신호가 잘 퍼지는지 아닌지를 나타내는 양의 정수 C_i 가 있을 때, 두 도시 a, b 사이에서 전화를 하려면 전화기가 $C_a \times C_b$ 이상의 출력을 낼 수 있어야 한다. 이 법칙은 이상한 일이지만 두 명이 같은 도시 안에 있어도 여전히 적용된다.

두 조승현은 여행을 시작하기전에 똑같은 전화기를 하나씩 사서 여행이 끝날 때 까지 그 전화기만 사용할 것이다. 전화기의 가격은 전화기가 낼 수 있는 출력에 비례하기 때문에, 어떻게 여행하느냐에 따라 필요한 전화기의 가격이 변한다. S 와 E 가 주어졌을 때, 여행을 무사히 마치기 위해 필요한 전화기의 출력의 최소값을 구해서 두 조승현을 만족시켜주자.

Input

주의: 이 문제의 입력은 하나의 테스트 케이스로 구성되어 있다.

첫 번째 줄에 도시의 수 N ($2 \leq N \leq 500$)과 도로의 수 M ($1 \leq M \leq 3,000$)이 주어진다. 두 번째 줄에 N 개의 정수가 주어진다. i 번째 정수는 C_i ($1 \leq C_i \leq 40,000$) 를 나타내며 이것은 i 번 도시의 무선 신호 상수를 나타낸다.

이후로 M 개의 줄에 걸쳐 2개의 정수 a, b ($1 \leq a, b \leq N, a \neq b$) 가 공백으로 구분되어 주어지는데, 이것은 도시 a 와 도시 b 사이를 잇는 도로가 존재함을 뜻한다.

그 다음 줄에 질문의 수 Q ($1 \leq Q \leq {}_nC_2$)가 주어진다. 이후로 Q 개의 줄에 걸쳐 2개의 정수 S, E ($1 \leq S, E \leq N, S \neq E$) 가 공백으로 구분되어 주어지는데, 이것은 처음에 조승현13이 S 번 도시에 있고 조승현16이 E 번 도시에 있을 때의 상황을 뜻하는 것이다.

Output

Q 개의 줄에 각 질문에 대해 여행을 성공적으로 마치기 위해 필요한 전화기의 출력력의 최소값을 구하여 출력한다.

Sample input and output

standard input	standard output
5 5	5
5 1 3 100 12	3
1 2	100
2 3	
3 1	
4 3	
4 5	
3	
1 2	
2 3	
5 3	

Problem K. 검역소

연약한 사람들이 모여 사는 나라가 있다. 이 곳에는 N 개의 도시들이 있고, 두 도시 사이를 연결하는 길이 $N - 1$ 개 있어 어느 두 도시도 오직 하나의 경로로만 서로 통행할 수 있게 되어 있다.

이 곳에는 몇 년에 한 번씩 전염병이 창궐하여 큰 피해가 일어난다. 정부에서는 이 문제를 해결하기 위해 $N - 1$ 개의 길들 중 K 개의 길에 검역소를 운영하려고 한다. 검역소는 감염된 사람이 지나가지 못하게 함으로서 전염병이 전파될 수 없는 장벽과 같은 역할을 해 준다.

하지만 검역소들만으로는 전염병이 일어나지 못하게 할 수는 없다. 따라서 어떤 사람이 전염병에 감염될 경우를 대비하여 치료제를 비축해 두려고 한다. 어떤 한 사람이 전염병에 감염될 때에도 전염병에 걸릴 수 있는 모든 사람들이 치료제를 하나씩 받을 수 있게 하기 위해 비축해야 할 치료제의 최소 개수를 구하여라.

Input

첫 번째 줄에 테스트 케이스의 수 T 가 주어진다.

각 테스트 케이스의 첫 번째 줄에는 도시의 개수를 의미하는 자연수 $N(2 \leq N \leq 100,000)$ 과 설치할 수 있는 검역소의 개수 $K(1 \leq K \leq N - 1)$ 가 주어진다.

두 번째 줄에는 N 개의 자연수가 주어지며, i 번째 자연수는 i 번 도시의 인구 $X_i(1 \leq X_i \leq 1,000,000,000)$ 을 의미한다.

세 번째 줄부터 $N - 1$ 개의 줄에는 도로의 정보가 한 줄에 하나씩 주어진다. 한 줄에는 두 개의 정수 $A_i, B_i(1 \leq A_i \leq N, 1 \leq B_i \leq N)$ 가 주어지며 이것은 A_i 번 도시와 B_i 번 도시가 도로로 연결되어 있음을 의미한다.

Output

전염병이 퍼질 경우에 대비해 정부에서 비축해야 하는 치료제의 개수를 하나의 정수로 출력한다.

Sample input and output

standard input	standard output
<pre>1 5 2 3 9 7 2 5 1 3 2 4 3 5 4 3</pre>	11

Notes

3 번 도시와 5 번 도시를 있는 도로와 4 번 도시와 3 번 도시를 있는 도로에 검역소를 설치하면 치료제를 11 인분만 비축해도 된다. 1 번 도시에 전염병이 발생할 경우 1 번 도시와 3 번 도시의 10 명의 사람들이, 2 번 도시에 전염병이 발생할 경우 2 번 도시와 4 번 도시의 11 명의 사람들이, 3 번 도시에 발생할 경우 10 명, 4 번 도시에 발생할 경우 11 명, 5 번 도시에 발생할 경우 5 명이 감염될 수 있으므로 어느 경우에도 11 인분의 치료제로 충분히 전염병을 막을 수 있다.

Problem L. 직사각형의 개수

$N \times M$ 크기의 격자판이 있다. 격자판의 각 칸은 1에서 K 사이의 숫자 하나가 적혀 있다. 편의상 i 행 j 열 ($1 \leq i \leq N, 1 \leq j \leq M$)의 격자를 (i, j) 로 표시한다.

주어진 격자판 내에서 k 종류의 숫자가 적혀 있는 크기가 $n \times m$ 인 서로 다른 직사각형의 개수 $C_{k,n,m}$ 을 모두 구하는 프로그램을 작성하라. 여기서 직사각형은 $1 \leq x_1 \leq x_2 \leq N, 1 \leq y_1 \leq y_2 \leq M$ 를 만족하는 네 개의 정수 x_1, y_1, x_2, y_2 에 의해 결정되며, $x_1 \leq x \leq x_2$ 와 $y_1 \leq y \leq y_2$ 를 만족하는 모든 격자 (x, y) 를 일컫는다. 이 직사각형의 크기는 $(x_2 - x_1 + 1) \times (y_2 - y_1 + 1)$ 이 된다.

예를 들어 $N = M = K = 3$ 일 때, 다음과 같이 숫자가 적혀 있다고 하자.

1	2	3
1	2	2
1	1	3

$x_1 = 2, y_1 = 1, x_2 = 3, y_2 = 2$ 라면, 4개의 격자 $(2, 1), (2, 2), (3, 1), (3, 2)$ 가 모두 한 직사각형에 속한다. 이 직사각형만 떼어 적으면 다음과 같다.

1	2
1	1

이 직사각형에는 1 혹은 2밖에 없으므로 두 종류의 수가 쓰인 직사각형이다. 두 직사각형이 서로 다르다는 것은 $[x_1, y_1, x_2, y_2]$ 가 서로 다르다는 것이다.

Input

주의 : 이 문제의 입력은 하나의 테스트 케이스로 구성되어 있다.

첫 번째 줄에 세 정수 N, M, K ($1 \leq N, M \leq 512, 1 \leq K \leq 9$)가 공백으로 구분되어 주어진다.

두 번째 줄부터 N 개의 줄에 걸쳐 격자판의 각 칸에 적힌 숫자를 의미하는 M 개의 숫자가 공백 없이 주어진다. 각 숫자는 1에서 K 사이의 숫자이다.

Output

모든 C 값을 출력하기에는 시간이 너무 오래 걸리기 때문에, 첫 번째 줄에 다음의 값을 $1,000,000,007$ 로 나눈 나머지를 출력한다.

$$\prod_{k=1}^K \prod_{n=1}^N \prod_{m=1}^M (C_{k,n,m} + k \times n \times m)$$

Sample input and output

standard input	standard output
2 2 2 11 12	81000
2 4 4 1223 1233	199080176
3 3 3 123 122 113	45470123