나의 행렬곱셈 답사기

스페셜 저지

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	128 MB	520	338	295	64.270%

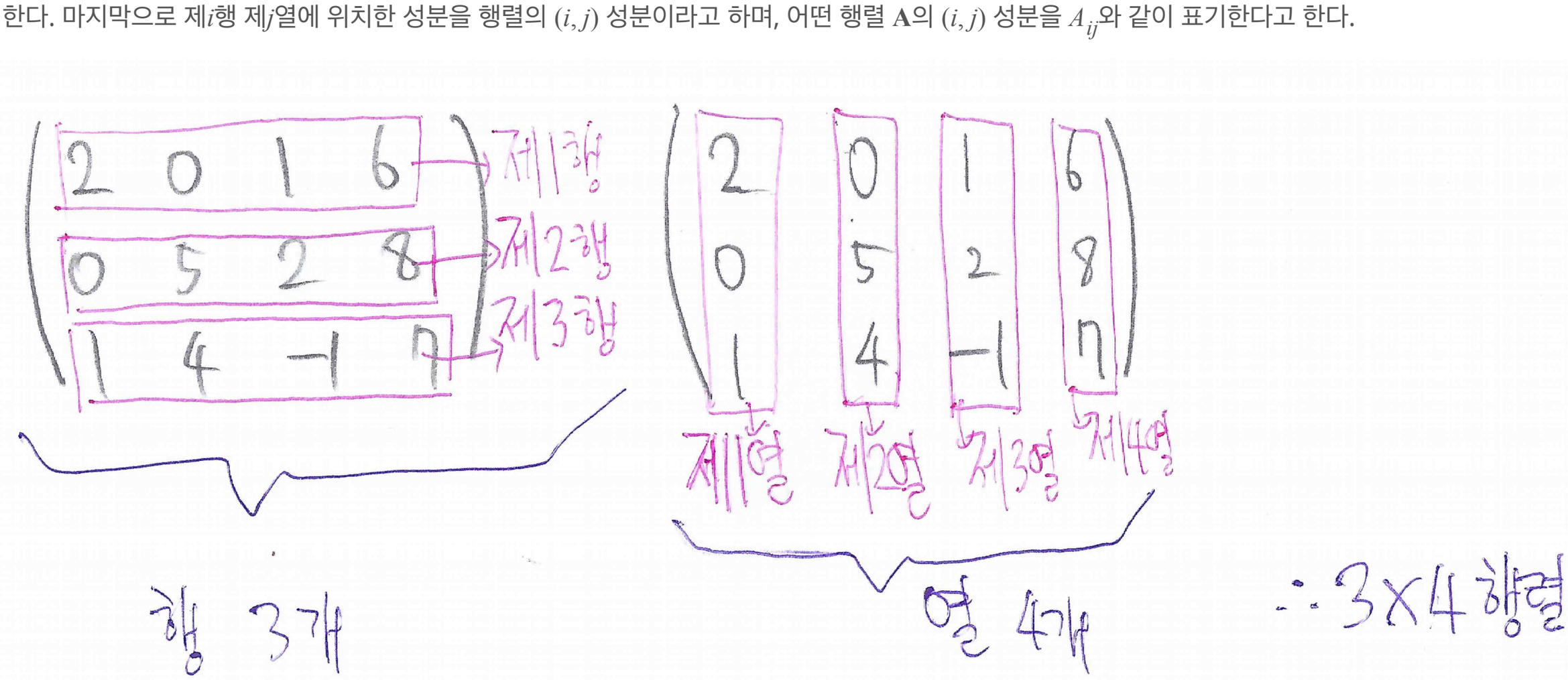
문제

계산은 사람에게나 컴퓨터에게나 상당히 번거로운 일인 것 같다. 특히 n개의 행렬 $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_n$의 곱, 즉 $\mathbf{M}_1\mathbf{M}_2\cdots\mathbf{M}_n$같은 것은 정말이지 계산하기 귀찮다.

행렬과 그 곱셈이 익숙하지 않은 사람들을 위해 설명을 해 보자면, 먼저 행렬은 여러 수나 기호, 문자, 수식 같은 것을 직사각형 모양으로 적절히 배열한 후 이를 괄호로 묶은 것을 말한다. 편의상 이 문제에서는 행렬에 정수만 배열한다고 가정한다. 예를 들어 아래와 같은 것이 행렬의 한 예이다.

$$\begin{pmatrix} 2 & 0 & 1 & 6 \\ 0 & 5 & 2 & 8 \\ 1 & 4 & -1 & 7 \end{pmatrix}$$

행렬에 배열된 수를 *성분*이라고 한다. 행렬의 가로줄은 *행*이라고 부르며, 위에서부터 차례로 제1행, 제2행, 제3행, ... 식으로 이름을 붙인다. 또한 행렬의 세로줄은 *열*이라고 부르며, 왼쪽에서부터 차례로 제1열, 제2열, 제3열, ... 식으로 이름을 붙인다. m개의 행과 n개의 열로 이루어진 행렬은 $\boldsymbol{m \times n}$ **행렬**이라고 한다. 마지막으로 제i행 제j열에 위치한 성분을 행렬의 (i,j) 성분이라고 하며, 어떤 행렬 $\mathbf{A}$의 (i,j) 성분을 A_{ij}와 같이 표기한다고 한다.



그러면 행렬을 어떻게 곱한다는 것일까? 실수에서의 곱셈과 같이, 행렬의 곱셈도 두 개의 행렬을 가지고 할 수 있다. $\mathbf{A}$가 $m \times n$ 행렬이고, $\mathbf{B}$가 $n \times p$ 행렬

일 때, **A**와 **B**의 곱 **AB**는 아래와 같은 $m \times p$ 행렬로 정의된다고 한다.

$$\mathbf{AB} = \begin{pmatrix} \sum_{k=1}^{n} A_{1k}B_{k1} & \sum_{k=1}^{n} A_{1k}B_{k2} & \cdots & \sum_{k=1}^{n} A_{1k}B_{kp} \\ \sum_{k=1}^{n} A_{2k}B_{k1} & \sum_{k=1}^{n} A_{2k}B_{k1} & \cdots & \sum_{k=1}^{n} A_{2k}B_{kp} \\ \vdots & \vdots & & \vdots \\ \sum_{k=1}^{n} A_{mk}B_{k1} & \sum_{k=1}^{n} A_{mk}B_{kp} & \cdots & \sum_{k=1}^{n} A_{mk}B_{kp} \end{pmatrix}$$

AB의 각 성분을 계산하기 위해 n회의 정수 곱셈이 필요하고, **AB**이 $m \times p$ 행렬이므로, 모든 성분을 계산하기 위해 총 $(m \times p) \times n = m \times n \times p$회의 정수 곱셈이 필요함을 알 수 있다.

A의 열의 수와 **B**의 행의 수가 같을 때에만 행렬의 곱셈이 정의된다. 예를 들어 3×2 행렬과 4×5 행렬을 곱할 수는 없다는 것이다.

그렇다면 우리가 처음 생각했던 행렬 n개의 곱은 어떨까? 실수에서 곱셈을 할 때 곱셈의 여러 성질들을 활용하듯이, 행렬을 곱할 때에도 이러한 곱셈의 성질을 활용할 수 있다.

일반적으로 행렬의 곱셈은 교환법칙이 성립하지 않지만, 결합법칙은 성립하는 것으로 알려져 있다. 다시 말해, $m \times n$ 행렬 **A**, $n \times p$ 행렬 **B**, $p \times q$ 행렬 **C**에 대해 일반적으로

- $\mathbf{AB} \neq \mathbf{BA}$
- $\mathbf{ABC} = (\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$

임이 알려져 있다는 것이다. 즉 행렬 여러 개를 곱할 때 행렬이 나열된 순서를 바꿀 수는 없지만, 곱하는 순서는 상관이 없음을 알 수 있다. 그런데 행렬을 곱하는 순서를 바꾼다고 실제로 정수 곱셈의 수가 바뀔까? 실제 예시를 통해 이를 확인해보자.

A가 2×4 행렬이고, **B**가 4×3 행렬, **C**가 3×5 행렬이라고 하자. 행렬의 곱 **ABC**를 계산하기 위해 필요한 정수 곱셈의 수를 분석해 보면 아래와 같다.

- $(\mathbf{AB})\mathbf{C}$와 같이 계산한다면
  - 2×4 행렬 **A**와 4×3 행렬 **B**를 곱할 때 $2 \times 4 \times 3 = 24$회의 정수 곱셈이 필요하며, 그 결과 2×3 행렬이 만들어진다.
  - 2×3 행렬 **AB**와 3×5 행렬 **C**를 곱할 때 $2 \times 3 \times 5 = 30$회의 정수 곱셈이 필요하며, 그 결과 2×5 행렬이 만들어진다.
  - 따라서 총 $24 + 30 = 54$회의 정수 곱셈이 필요함을 알 수 있다.
- $\mathbf{A}(\mathbf{BC})$와 같이 계산한다면
  - 4×3 행렬 **B**와 3×5 행렬 **C**를 곱할 때 $4 \times 3 \times 5 = 60$회의 정수 곱셈이 필요하며, 그 결과 4×5 행렬이 만들어진다.
  - 2×4 행렬 **A**와 4×5 행렬 **BC**를 곱할 때 $2 \times 4 \times 5 = 40$회의 정수 곱셈이 필요하며, 그 결과 2×5 행렬이 만들어진다.
  - 따라서 총 $60 + 40 = 100$회의 정수 곱셈이 필요함을 알 수 있다.

3개의 행렬을 곱할 때에도 행렬을 곱하는 순서에 따라 정수 곱셈의 횟수가 달라질 수 있으니, n개의 행렬을 곱할 때 역시 마찬가지임을 쉽게 짐작할 수 있을 것이다. 행렬의 수가 많아지면 많아질수록 행렬을 곱하는 방법의 수는 다양하게 존재한다. 예를 들어 $n = 4$일 때, 4개의 행렬 $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_4$을 곱하는 방법에는 아래 5가지가 있다.

- $\big((\mathbf{M}_1\mathbf{M}_2)\mathbf{M}_3\big)\mathbf{M}_4$
- $\big(\mathbf{M}_1(\mathbf{M}_2\mathbf{M}_3)\big)\mathbf{M}_4$
- $(\mathbf{M}_1\mathbf{M}_2)(\mathbf{M}_3\mathbf{M}_4)$
- $\mathbf{M}_1\big((\mathbf{M}_2\mathbf{M}_3)\mathbf{M}_4\big)$
- $\mathbf{M}_1\big(\mathbf{M}_2(\mathbf{M}_3\mathbf{M}_4)\big)$

이렇게 많은 방법들 가운데 어떤 것을 택하더라도 그 결과가 같으므로, 이 방법들 가운데 필요한 정수 곱셈의 수가 가장 적은 것을 택하면 행렬을 곱하는 데 필요한 시간이 최소화될 것이다.

계산을 좋아하는 승현이는 최근 이와 같은 n개의 행렬의 곱에 대해 배웠는데, 보통 사람은 이해할 수 없는 일이지만 몇몇 예제를 계산해 보더니 곱셈의 매력에 푹 빠지고 말았다. 성분끼리 곱한 것을 모두 더하는 것이 참 아름답다고 하는데, 이해하긴 어렵지만 일단 그렇다고 하자.

승현이가 정수 곱셈을 하는 데에는 0의 시간이 걸리기 때문에, 승현이는 '행렬을 곱할 때 필요한 정수 곱셈 횟수를 굳이 최소화할 필요가 있을까?' 하는 생

각을 하게 되었다. 승현이는 이러한 의문을 품고 선생님께 질문을 하였고, 이에 선생님은 최악의 정수 곱셈 횟수와 최적의 정수 곱셈 횟수 사이의 차가 상당히 커지는 경우가 있기 때문에, 정수 곱셈에 상당한 시간이 걸리는 일반인들에게는 곱셈 횟수를 최소화하는 것이 중요하다고 대답해 주셨다. "최악/최적의 정수 곱셈 횟수"라는 것은 행렬을 곱하는 모든 방법들 가운데 필요한 정수 곱셈 횟수가 가장 많은/적은 방법의 정수 곱셈 횟수를 의미한다.

일반인들의 심정에 전혀 공감할 수 없었던 승현이는 질문을 이어나갔고, 승현이의 질문 공세에 지친 선생님은 결국 모든 자연수 K에 대해, 행렬을 곱하기 위해 필요한 최악의 정수 곱셈 횟수와 최적의 정수 곱셈 횟수의 차이가 정확히 K가 되는 어떤 행렬들 $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_n$이 항상 존재한다고 대답해 주셨다.

충격을 받은 승현이는 당신에게 혹시 임의의 K를 줄 때 그런 행렬들을 아무 것이나 하나 찾아줄 수 있는지 부탁해왔다. 최악의 곱셈 횟수와 최적의 곱셈 횟수의 차가 K인 어떤 행렬들을 찾아내는 프로그램을 작성하자.

입력

첫 줄에 정수 K ($1 \le K \le 10^9$)가 주어진다.

출력

첫 줄에 K를 만족시킬 수 있는 데이터의 행렬 개수 정수 N($1 \le N \le 100$)을 출력한다. 둘째 줄에는 해당 행렬의 정보를 (N+1)개의 정수 a_0, a_1, .., a_n로 나타내어 출력한다. 행렬의 크기는 a_0 x a_1, a_1 x a_2, .., a_{n-1} x a_n 이다.

예제 입력 1 복사

```
20
```

예제 출력 1 복사

```
3
3 2 4 2
```

예제 입력 2 복사

```
20160528
```

예제 출력 2 복사

```
8
177 42 12 193 193 201 124 47 36
```

출처

Contest > Coder's High > Coder's high 2016 Round 1: Online G번

- 문제를 만든 사람: tncks0121

메모

메모 작성하기

기부하기
기능 추가 요청
스페셜 저지 제작 프로젝트

채점 현황

채점 현황

최근 풀린 문제
재채점 및 문제 수정

유저 대회 / 고등학교 대회

FunctionCup kriiicon 구데기컵
꼬마컵 네블컵 소프트콘 웰노운컵
HYEA Cup 경기과학고등학교
대구과학고등학교 부산일과학고
서울과학고등학교 선린인터넷고등학교

Coder's High

대학교 대회

KAIST POSTECH 고려대학교
광주과학기술원 국민대학교 서강대학교
서울대학교 숭실대학교 아주대학교
연세대학교 인하대학교 전북대학교
중앙대학교 충남대학교 한양대 ERICA
홍익대학교
경인지역 6개대학 연합 프로그래밍 경시 대회

사업자 등록 번호: 541-88-00682
대표자명: 최백준
주소: 서울시 서초구 강남대로 359 대우도씨에빛2 5층 502호
전화번호: 02-521-0487 (이메일로 연락 주세요)
이메일: contacts@startlink.io
통신판매신고번호: 제 2017-서울서초-2193 호