Problem A. Apple Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	64 megabytes

사과나무란 각 노드 안에 사과가 담겨있는 트리이다. 각 노드에 담겨있는 사과의 개수는 다를 수 있으며, 0일 수도 있다.

한 마리의 벌레가 사과나무의 임의의 노드부터 움직이기 시작한다. 벌레가 어떤 노드에 위치해 있을 때, 이 벌레는 그 노드 안의 모든 사과를 먹을 수 있다. 사과를 다 먹은 후에 벌레는 나무의 가지를 따라 다른 노드로 이동한다. 만약 현재 노드에 연결된 노드들이 여러 개 있다면 그 중 하나를 임의로 택할 수 있으나 이미 방문했던 노드는 다시 방문할 수 없다. 더 이상 갈 수 있는 노드가 없을 때 벌레는 움직임을 종료한다.

사과나무가 주어졌을 때, 벌레가 먹을 수 있는 사과의 최대 개수와 벌레가 움직이기 시작하는 노드를 구해내는 프로그램을 작성하시오.

Input

입력은 여러 데이터로 구성된다. 첫째 줄에는 사과나무의 노드 수를 나타내는 자연수 $n(1 \le n \le 10,000)$이 주어진다. 다음 줄에는 1번부터 n번까지의 노드에 들어있는 사과의 개수가 차례대로 주어진다. 사과의 전체 개수는 signed 32-bit integer 범위를 넘지 않는다. 다음 $n-1$개의 줄에는 서로 다른 두 정수 $A, B(1 \le A, B \le n)$이 주어진다. 이는 A번 노드와 B번 노드를 연결하는 가지가 존재한다는 의미이다. 입력은 $n = 0$일 때 끝난다.

Output

각각의 데이터에 대해서, 벌레가 먹을 수 있는 사과의 최대 개수와 벌레가 시작해야 하는 노드의 번호를 출력한다. 만약 가능한 출발 노드가 여러개일 수 있다면 그 중 번호가 제일 작은 것을 출력한다.

Example

standard input	standard output
2 1 2 1 2 3 1 2 3 1 3 3 2 4 1 5 5 5 1 2 2 3 2 4 0	3 1 6 1 15 3

(by yuki)

Problem B. Bouncing Ball

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	64 megabytes

바닥에 n개의 스프링들이 같은 간격으로 놓여 있다. 첫 번째 스프링에서 출발하여, 당신은 스프링 위에서 공을 튀기면서 마지막(n번째) 스프링까지 도달하려 한다.

매번 공을 튀길 때, 공을 움직일 수 있는 범위는 스프링의 탄성력에 따라서 정해진다. 이 탄성력은 스프링마다 다를 수도 있다. i번째 스프링의 탄성력을 P_i라 할 때, i번째 스프링에서 공을 튀기면 $i - P_i$번째 스프링부터 $i + P_i$번째 스프링으로 이동할 수 있다. 예를 들어 $P_5 = 3$이고, 공을 5번 스프링에서 튀기면, 공은 2, 3, 4, 5, 6, 7, 8번 스프링 중 하나로 이동할 수 있다.

스프링에서 공을 튀길 때마다 스프링의 수명이 조금씩 닳기 때문에, 당신은 스프링을 최소 회수로 사용하여 마지막 스프링에 도달하려 한다. 각 스프링의 탄성력이 주어졌을 때, 첫 번째 스프링에서 시작하여 최소 몇 번 스프링에 공을 튀겨야 마지막 스프링에서 도달할 수 있는지 구하는 프로그램을 작성하시오.

Input

입력은 여러 데이터로 구성된다. 첫째 줄에는 스프링의 개수를 나타내는 자연수 $n(1 \le n \le 1,000,000)$이 주어진다. 두 번째 줄부터는 차례로 $P_1, P_2, \cdots, P_n$이 주어진다. 이 값들은 한 개 이상의 공백문자로 분리되어 있다. 각각의 $1 \le i \le n$에 대해서, P_i는 $1 \le P_i \le n$을 만족한다. 입력은 $n = 0$일 때 끝난다.

Output

각각의 데이터에 대해서, 스프링의 사용 회수의 최소값을 출력한다.

Example

standard input	standard output
4 1 1 1 1 10 3 2 1 2 1 2 1 2 3 1 0	3 4

Problem C. Classification of Surfaces

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	64 megabytes

곡면(surface)은 여러 가지 방법으로 정의할 수 있다. 그 중 하나는 위상적인 성질을 이용한 정의인데, 어떤 위상공간이 각 점에서 국소적으로 2차원 원과 위상적으로 닮아있으면 그러한 위상공간을 곡면이라 부른다. 어떤 한 공간을 부러뜨리지 않고 연속적으로 구부리거나 늘리면서 만들 수 있는 공간은 원래의 공간과 위상적으로 닮아있다고 한다.

수학자들은 어떤 수학적 대상을 연구할 때, 그 대상들을 비슷한 것들로 분류하는 작업을 한다. 곡면을 연구할 때에도 이러한 방법을 사용하는데, 특히 2차원 곡면의 경우에는 그 분류가 완전히 끝나 있다.

Rado는 이러한 곡면의 분류 작업을 수행하기 위해, 먼저 곡면을 삼각형들로 나누는 방법을 이용하였다. 즉, 곡면 위에 삼각형(곡면은 완전한 평면이 아니기 때문에, 실제로는 삼각형과 위상적으로 닮아 있는 도형이 그려지게 된다)을 가득 그리고, 삼각형의 변을 따라가면서 곡면을 뜯어낸 뒤, 이를 평면 상의 다각형으로 놓고 살펴보는 방법을 이용하였다. 이와 같이 얻어진 다각형에서 뜯어진 변들을 방향을 맞춰 붙여주면 원래의 곡면을 얻을 수 있다.

이해를 돕기 위해 위의 그림을 살펴보자. 오른쪽 그림은 뫼비우스의 띠라 불리는 곡면인데, 이는 왼쪽 그림과 같이 삼각형으로 표현할 수 있다. 이 때 위와 아래의 변을 화살표 방향을 따라 붙이면 뫼비우스의 띠가 된다.

Rado는 그의 증명을 완성하기 위해 분할된 삼각형들을 T_1, T_2, $\cdots$, T_n의 순서로 나열하되, T_i번 삼각형이 T_1, T_2, $\cdots$, T_{i-1}중에 적어도 하나와 붙어있도록 나열해야 했다. Rado가 그의 증명을 완성할 수 있도록, 삼각형들을 순서대로 나열해주는 프로그램을 작성하시오.

Input

입력은 여러 데이터로 구성된다. 첫째 줄에는 삼각형의 개수를 나타내는 자연수 $n(1 \le n \le 100,000)$과 자연수 m이 주어진다. 다음 m개의 줄에는 서로 다른 두 자연수 i, $j(1 \le i, j \le n)$이 주어진다. 이는 i번 삼각형과 j번 삼각형이 붙어 있다는 의미이다. 입력은 $n = m = 0$일 때 끝난다.

Output

각각의 데이터에 대해서, 주어진 조건대로 나열한 삼각형의 번호를 출력한다. 가능한 답이 여러 개인 경우에는 사전식으로 제일 먼저 오는 것을 출력한다. 답을 구성할 수 없는 경우는 입력으로 주어지지

않는다.

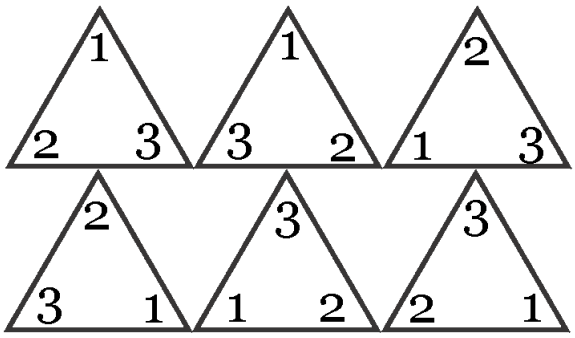
Example

standard input	standard output
2 2 1 2 1 2 3 3 3 2 3 1 0 0	1 2 1 3 2

Problem D. Dihedral Group

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	64 megabytes

정 n각형의 각 꼭지점에 1부터 n까지의 번호를 중복 없이 부여하는 것을 생각해 보자. 예를 들어 $n = 3$일 경우, 아래 그림과 같이 여섯 가지의 경우가 가능하다.



이와 같이 부여된 번호를 한 공통된 꼭지점을 기준으로 순서대로 나열하면 각각이 한 순열이 된다. 예를 들어 위의 그림에서 위의 꼭지점을 기준으로 삼고 시계 반대방향으로 점을 나열하면 각각 (1 2 3), (1 3 2), (2 1 3), (2 3 1), (3 1 2), (3 2 1)의 순열이 된다.

이 때 (1 2 $\cdots$ n)의 순열에 해당하는 정 n각형을 임의의 각도만큼 회전하거나, 혹은 임의의 축을 기준으로 뒤집었을 때 얻어지는 순열을 모아놓은 것을 Dihedral group이라 부른다. 예를 들어 (1 2 3)을 세로축을 기준으로 좌우로 뒤집으면 (1 3 2)가 되며, (1 2 3)을 가로축을 기준으로 상하로 뒤집은 후 반시계방향으로 60도 회전하면 (3 2 1)이 된다.

n이 주어졌을 때, Dihedral group의 원소에 해당하는 순열을 구하는 프로그램을 작성하시오.

Input

입력은 여러 데이터로 구성된다. 각 줄에는 자연수 $n(3 \le n \le 500)$이 주어진다. 입력은 $n = 0$일 때 끝난다.

Output

각각의 데이터에 대해서, 먼저 첫째 줄에 Dihedral group의 원소 수를 출력한다. 다음 줄들에는 한줄에 하나씩, 그리고 사전 순서대로 Dihedral group의 원소를 출력한다. 만약 Dihedral group의 원소가 1000개를 넘으면, 사전 순서대로 앞의 1000개까지만 출력하도록 한다.

Example

standard input	standard output
3 0	6 1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

Problem E. Elementary Operation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	64 megabytes

Gaussian Elemination은 다원 1차 연립방정식을 풀기 위한 방법의 하나이다. 이 방법은 행렬에 elemantary operation을 가하여 풀기 쉬운 꼴로 행렬을 변경하는 방법이다.

만일 주어진 계수들의 행렬이 모두 정수로 되어 있다면, Gaussian elemination을 적용한 결과가 유리수일 것이다. 이러한 점을 이용하여, 정수에서도 elementary operation을 생각할 수 있다. 정수로 된 $n \times m$ 행렬에서의 elemantary operation은 다음과 같이 정의된다.

- 서로 다른 두 행/열을 뒤바꾼다(행끼리, 혹은 열끼리만).
- 어떤 행/열에 -1을 곱한다.
- 어떤 행/열에 다른 행/열의 정수배를 더한다(행끼리, 혹은 열끼리만).

위와 같은 elementary operation을 잘 가하면, 행렬의 대각선에 d_1, d_2, $\cdots$, d_l만 남도록 할 수 있다($l = \min\{n, m\}$). 이 때 특별히, d_i가 d_{i+1}의 약수가 되도록 할 수 있다($1 \le i < l$). 정수 a가 b의 약수라는 말은, 물론 $b = ac$를 만족하는 정수 c가 존재한다는 의미이다.

정수로 된 행렬이 주어졌을 때, 가능한 d_1값을 모두 구하는 프로그램을 작성하시오.

Input

입력은 여러 데이터로 구성된다. 첫째 줄에는 두 자연수 n, $m(1 \le n, m \le 300)$이 주어진다. 다음 n개의 줄에는 m개의 정수로 행렬이 주어진다. 행렬을 이루는 각각의 값들은 절댓값이 $2^{63} - 1$을 넘지 않는 정수이다. 입력은 $n = m = 0$일 때 끝난다.

Output

각각의 데이터에 대해서, 가능한 d_1값을 오름차순으로 출력한다.

Example

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

Problem F. Free Group of Alphabets

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	64 megabytes

어떤 집합 G와 G의 두 원소들간에 정의된 이항 연산 $*$가 다음 조건들을 만족하면 group이라 부른다.

- G는 $*$에 의해 닫혀 있다. 즉, $x, y \in G$이면 $x * y \in G$.
- $*$는 결합법칙을 만족한다. 즉, $x, y, z \in G$이면 $x * (y * z) = (x * y) * z$.
- G에는 항등원이 존재한다. 즉, [모든 $x \in G$에 대하여 $x * e = e = e * x$]인 $e \in G$가 존재.
- G에는 $*$의 역원이 존재한다. 즉, 각 $x \in G$에 대하여 [$x * x' = e = x' * x$인 $x' \in G$가 존재].

또, 집합 S가 공집합이 아닐 때, S의 원소들과 S의 역원들로 만들어진 문자열들의 집합을 S가 만드는 free group이라 한다.

이 문제에서는 알파벳들이 만드는 free group을 생각하자. 이를 위해서는 먼저 문자열들간의 이항 연산이 필요한데, 이는 물론 두 문자열을 붙여 쓰는 연산이다. 다음으로는 항등원이 필요한데, 이는 물론 빈 문자열(null string)이다. 마지막으로 알파벳의 역원이 필요한데, 이는 대소문자를 뒤바꾼 것으로 한다. 즉 `a`는 `A`의 역원이며, 따라서 `A`는 `a`의 역원이다. 즉, 알파벳들이 만드는 free group은 대소문자가 섞여 있는 문자열들의 집합이다.

그런데 역원 관계에 있는 두 알파벳이 문자열에서 연달아 나오는 경우, 둘 사이에 연산을 적용하여 두 문자를 제거할 수 있다. 예를 들어 `xyyyXxYY`와 같은 문자열이 주어진 경우, 중간의 `Xx`를 제거하여 `xyyyYY`로 만들 수 있고, `yY`를 제거하여 `xyyY`로 만들 수 있으며, 다시 `yY`를 제거하여 `xy`로 만들 수 있다.

Free group의 한 원소가 주어졌을 때, 위와 같은 연산을 적용하여 만들 수 있는 문자열의 최소 길이를 구하는 프로그램을 작성하시오.

Input

입력은 여러 데이터로 구성된다. 각 데이터는 길이가 100,000을 넘지 않는 문자열로 구성된다. 입력되는 문자열은 압축된 문자열일 수 있다. 입력 중간에 괄호로 둘러싸인 자연수 `(n)`이 나올 경우, 이는 괄호 바로 앞의 문자가 n번 반복된다는 의미이다. 압축을 풀었을 때의 문자열의 길이는 10^{18}을 넘지 않는다.

Output

각각의 데이터에 대해서, 가능한 최소 길이를 출력한다. 이 때에는 압축을 고려하지 않는다.

Example

standard input	standard output
`x(61)` `xy(3)XxY(2)`	`61` `2`

Problem G. Gene and Subgene

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	64 megabytes

생명체의 유전자는 보통 A, C, G, T로 구성된 길이 n의 문자열로 표현된다. 이러한 문자열에서 나타나는 문자들의 순서나 패턴이 생명체의 특징을 결정하기 때문에 중요하다. 특히 이러한 문자열에서 특정 패턴이 반복적으로 나타날 경우가 중요한 연구의 대상이 된다.

주어진 유전자의 부분 유전자는, 유전자를 표현하는 문자열의 부분 문자열을 의미한다. 이러한 부분 문자열은 주어진 문자열에서 이어서 나타나는 것이여야 한다. 예를 들어, "AC"는 C가 A다음에 오지 않기 때문에 "AGCT"의 부분 문자열이 아니다. "AGCT"의 부분 문자열로는 "AG", "GC", "AGC", "AGCT" 등이 있다.

이와 같은 부분 유전자 중에서 m번 이상 나타나는 부분 유전자들이 있다. 예를 들어 "AGAG"가 있을 때, "A", "G", "AG"는 각각 두 번씩 나타나는 부분 유전자이다. 이와 같이 m번 이상 나타나는 부분 유전자의 개수와, k번째 부분 유전자를 구하는 프로그램을 작성하시오.

유전자들의 순서는 길이가 더 짧은 것이 더 앞에 오며, 같은 길이일 때는 사전 식으로 앞서는 것이 앞에 온다.

Input

입력은 여러 데이터로 구성된다. 첫재 줄에는 세 정수 $n(1 \le n \le 1,000), m(1 \le m \le n), k$가 주어진다. k는 가능한 범위를 넘지 않는다. 입력은 $n = m = k = 0$일 때 끝난다.

Output

첫째 줄에 m번 이상 나타나는 부분 유전자의 개수를 출력한다. 그 다음 줄에는 k번째의 부분 유전자를 출력한다.

Example

standard input	standard output
6 1 3 GGGGGG 4 2 3 AGAG 0	6 GGG 3 AG

Problem H. Height of a Tree

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	64 megabytes

각각의 간선에 음 아닌 가중치가 있고, 루트가 있는 트리가 있다. 이러한 트리의 높이는 루트에서 다른 노드까지의 거리들 중 최대값으로 정의한다.

이 트리의 간선의 가중치를 1씩 줄일 때마다 1만큼의 비용이 든다. 쉽게 생각하면, 가중치가 x인 간선의 가중치를 $y(0 \le y \le x)$로 줄일 때 $x - y$만큼의 비용이 든다는 의미이다. 문제를 쉽게 하기 위해서 음 아닌 정수에서 음 아닌 정수로 줄이는 경우만 생각하기로 하자.

이처럼 몇 개의 간선의 가중치를 잘 줄이면 트리의 높이가 $H(0 \le H)$가 되도록 할 수 있다. 이 때 어느 간선을 선택하여 얼마로 줄이느냐에 따라서 드는 비용이 달라질 수 있다.

트리의 높이를 H로 만들 수 있는 최소의 비용을 구하는 프로그램을 작성하시오.

Input

입력은 여러 데이터로 구성된다. 첫째 줄에는 정점의 개수 $n(2 \le n \le 100,000)$과 $H(0 \le H \le 10^9)$가 주어진다. 다음 n개의 줄에는 1번 정점부터 n번 정점까지 차례로 각 정점의 부모 정점과 그 정점에서 부모 정점까지의 거리(간선의 가중치)가 주어진다. 루트의 경우에는 부모 정점과 비용 대신에 0 두 개가 주어진다. 간선의 가중치의 총 합은 10^9을 넘지 않는다. 입력은 $n = H = 0$일 때 끝난다.

Output

각각의 데이터에 대해, 가능한 경우에는 최소 비용을, 불가능한 경우에는 -1을 출력한다.

Example

standard input	standard output
2 1 0 0 1 0 5 0 0 0 1 1 2 1 2 1 1 2	-1 5

Problem I. Internet Packet

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	64 megabytes

Input

Output

Example

`standard input`	`standard output`

Problem J. JM Playing with Domino

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	64 megabytes

Input

Output

Example

standard input	standard output