주최 아주대학교 AJOU UNIVERSITY | SW중심대학사업 National Center of Excellence in SW 주관 ANSi

후원 코딩엑스 AI dSPACE HYUNDAI MOBIS SOLVED.AC STARTLINK

APC

Ajou Univ. Programming Contest

Official 공식 문제지

Problemset

Division 2 — Challenge Round

- 본 문제지는 **Division 2**의 문제지입니다.
- 9문제가 정상적으로 인쇄되었는지 확인하십시오.
- 모든 문제의 메모리 제한은 1024MB로 동일 합니다.
- 참가자는 대회 중 사전에 승인받은 하드카피를 참고 할 수 있습니다.
- 문제의 난이도는 출제진이 판단한 난이도 순 입니다.
- 대회 중 문제에 관한 질문은 대회 페이지 내 질문 기능을 이용해주시기 바랍니다.
- 대회 중 퇴실은 불가합니다.

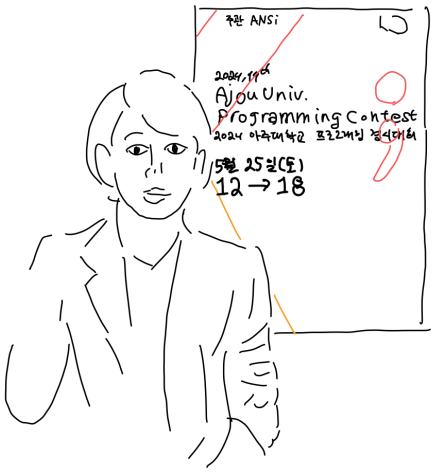
문항 목록

문제		시간 제한
A	이 대회는 이제 제 겁니다	1 초
B	마라탕후루	1 초
C	현대모비스 특별상의 주인공은? 2	1 초
D	APC2shake!	2 초
E	밤양갱	1 초
F	렬정! 렬정! 렬정!	1 초
G	너 재능있어	1 초
H	지금부터 서로 죽여라	1 초
I	나는 북극곰입니다	1.5 초

문제 A. 이 대회는 이제 제 겁니다

- 시간 제한 1초

문제



"제 마음대로 우승할 수 있는 겁니다."

2024년, 제14회 아주대학교 프로그래밍 경시대회(이하 2024 APC)가 개최된다! 2024 APC는 두 개의 Division으로 나누어 진행된다.

- Division 1은 누구나 참여할 수 있으며 수상 시 2024 경인지역 6개 대학 연합 프로그래밍 경시대회(이하 2024 shake!)의 진출권을 얻게 된다.
- Division 2는 소프트웨어융합대학이나 정보통신대학 소속이 아닌 모두가 참가할 수 있다. 하지만, 수상을 하더라도 2024 shake!에는 진출할 수 없다.

두 개의 Division을 같이 나갈 수 없다. 다시 말해, Division 1의 참가자와 Division 2의 참가자는 중복되지 않는다.

각 대회를 우승하게 되면 상금이 주어진다. 당신은 Division 1을 우승하게 되면 A만원, Division 2를 우승하게 되면 P만원, 2024 shake!를 우승하게 되면 C만원을 받게 된다. 소프트웨어융합대학이나 정보통신대학 소속이 아닌 당신은 어떤 대회라도 나가기만 하면 우승할 수 있다. 당신이 얻을 수 있는 최대 상금을 구해보자.

입력

첫 번째 줄에 세 정수 A, P, C가 공백으로 구분되어 주어진다. $(1 \leq A, P, C \leq 500)$

A는 Division 1, P는 Division 2, C는 shake!의 우승 상금을 의미한다.

출력

아주대학교 최고의 실력자인 당신이 문제의 조건에 맞게 대회에 출전했을 때 얻을 수 있는 최대 상금의 액수를 출력한다.

예제

standard input	standard output
500 300 500	1000

문제 B. 마라탕후루

- 시간 제한 1초

문제



"그럼 내가 선배 맘에"

아주대의 마스코트인 치토는 요즘 유행한다는 마라탕과 탕후루를 합쳐서 마라탕후루 가게를 오픈했다. 그런데 첫날부터 발주를 실수하는 바람에 딸기와 샤인머스캣이 박스째로 쌓이게 되었다. 과일은 신선도가 중요하기 때문에 오늘은 탕후루를 손님들에게 무료로 제공하기로 했다.

치토는 사전에 탕후루 꼬치 N 개를 만들어 두었다. 그런데 바빠서 손에 잡히는 대로 과일을 꽂다 보니 꼬치마다 꽂혀있는 딸기와 샤인머스캣의 개수가 다를 수 있다는 사실을 뒤늦게 깨달았다. 맛의 밸런스를 유지하기 위해서는 꼬치마다 꽂혀있는 딸기와 샤인머스캣의 개수가 동일해야 한다. 단, 서로 다른 꼬치에 꽂혀있는 딸기와 샤인머스캣의 개수가 같을 필요는 없다. 치토는 가게를 운영하느라 매우 바빠서 딸기와 샤인머스캣의 개수를 동일하게 만들어줄 로봇을 고용했다.

현재 i 번 꼬치에는 A_i 개의 딸기와 B_i 개의 샤인머스캣이 꽂혀있다. 치토가 고용한 로봇은 한 가지 문제점이 존재하는데, 바로 딸기와 샤인머스캣을 매번 정해진 개수만큼 꽂는다는 것이었다. 로봇은 1 분마다 다음의 행동을 한 번 진행한다.

- i 번 꼬치를 선택한다. $(1 \le i \le N)$
- i 번 꼬치에 딸기를 P 개, 샤인머스캣을 Q 개 꽂는다.

치토는 곰곰이 생각해 보니 로봇이 꼬치마다 딸기와 샤인머스캣의 개수를 똑같이 만들지 못할 수도 있다는 사실을 깨달았다.

로봇이 꼬치마다 딸기와 샤인머스캣의 개수를 똑같이 만들 수 있는지 없는지 확인해 보자.

입력

첫 번째 줄에 탕후루 꼬치의 개수 N, 로봇이 1 분마다 꽂는 딸기의 개수 P, 샤인머스캣의 개수 Q 가 공백으로 구분되어 주어진다. $(1 \le N, P, Q \le 100)$

두 번째 줄에 꼬치마다 꽂혀있는 딸기의 개수 $A_1, A_2, \ldots, A_N$ 이 공백으로 구분되어 주어진다. $(1 \le A_i \le 100)$

세 번째 줄에 꼬치마다 꽂혀있는 샤인머스캣의 개수 $B_1, B_2, \ldots, B_N$ 이 공백으로 구분되어 주어진다. $(1 \le B_i \le 100)$

입력으로 주어지는 모든 수는 정수이다.

출력

첫 번째 줄에 로봇이 꼬치마다 딸기와 샤인머스캣의 개수를 똑같이 만들 수 있다면 YES, 아니라면 NO를 출력한다.

로봇이 꼬치마다 딸기와 샤인머스캣의 개수를 똑같이 만들 수 있는 경우, 두 번째 줄에 꼬치마다 로봇이 딸기와 샤인머스캣을 꽂은 횟수를 공백으로 구분하여 출력한다. 가능한 답이 여러 가지인 경우, 그중 하나를 출력한다.

단, 딸기와 샤인머스캣의 양이 무한하지 않기 때문에 로봇이 꼬치마다 딸기와 샤인머스캣을 꽂은 횟수는 각각 $10\,000$번 이내여야 한다.

예제

standard input	standard output
3 2 1 3 5 7 5 6 7	YES 2 1 0
3 2 1 3 5 7 6 5 4	NO
1 100 50 45 45	YES 0

문제 C. 현대모비스 특별상의 주인공은? 2

- 시간 제한 1초

문제



올해도 어김없이 현대모비스는 실력 있는 소프트웨어 인재를 양성하고 독려하기 위해 APC의 주최 동아리인 A.N.S.I.를 후원하고 있다. 올해 현대모비스는 2024 APC에 참가한 여러분들을 위해 특별상을 준비하였다.

현대모비스는 $N \times N$ 크기의 격자판을 준비하였다. 격자판의 각 칸에는 알파벳 대문자가 한 글자씩 적혀있다. 특별상은 이 격자판에서 가장 많은 MOBIS를 찾은 사람에게 주어지는데, MOBIS를 찾는 것은 다음의 규칙을 따른다.

- 격자 위의 임의 위치에서 시작하여 상, 하, 좌, 우, 대각선의 8방향 중 한 방향으로 격자에 적힌 글자들을 차례로 5개 이어 붙였을 때, 이어 붙인 글자가 MOBIS여야 한다.

현빈이는 어떻게든 특별상의 주인공이 되고 싶다. 현빈이를 도와 주어진 격자판에서 MOBIS를 최대 몇 번 찾을 수 있는지 구해보자.

입력

첫 번째 줄에 정수 N이 주어진다. $(1 \leq N \leq 100)$ 두 번째 줄부터 $N+1$번째 줄까지, 현대모비스가 준비한 격자판의 정보가 주어진다. 각 줄은 N개의 알파벳 대문자로 이루어져 있다.

출력

주어진 격자판에서 찾을 수 있는 MOBIS의 개수를 출력한다.

예제

standard input	standard output
9 SXXXSXXXS XIXXIXXIX XXBXBXBXX XXXOOOXXX SIBOMOBIS XXXOOOXXX XXBXBXBXX XIXXIXXIX SXXXSXXXS	8
3 MOB AOI MUS	0

노트

아래는 두 번째 예제에 관한 설명이다.

MOBIS를 꺾어서 이어 붙일 수 없다.

문제 D. APC2shake!

- 시간 제한 2초

문제

2024년 아주대학교 프로그래밍 경시대회(이하 2024 APC) Division 1에서 훌륭한 성적을 거둔 참가자에게는 2024년 경인지역 6개 연합 프로그래밍 경시대회(이하 2024 shake!)에 진출할 수 있는 티켓이 주어진다. 2024 shake!의 진출자들은 아래의 규칙으로 선발된다.

- 아주대학교 학부 소속의 재학생
- 역대 국제 대학생 프로그래밍 경시대회(이하 ICPC)의 수상자가 아닌 자
- 역대 shake! 3위 이내의 수상자가 아닌 자

위 3가지 규칙을 만족하는 참가자 중 2024 APC에서 높은 등수를 가진 최대 10명의 참가자 모든 참가자들의 이름과 재학 여부, 역대 ICPC 수상 여부, 역대 shake! 최고 성적, 2024 APC의 성적이 주어질 때, 2024 shake!의 진출자들을 알아내 보자.

입력

첫 번째 줄에 참가자의 수 N이 주어진다. $(1 \le N \le 10^4)$

이후 N개 줄에 걸쳐 다음의 정보가 순서대로 공백으로 구분되어 주어진다.

- 이름 : 공백 없이 알파벳 소문자로만 이루어진 문자열이 주어진다. 이름의 길이는 10을 넘지 않는다.
- 재학 여부 : 재학 중이라면 `jaehak`이, 휴학 중이라면 `hewhak`가 주어진다.
- 역대 ICPC 수상 여부 : 역대 ICPC에서 수상하였다면 `winner`가, 아직 수상하지 못하였다면 `notyet`가 주어진다.
- 역대 shake! 최고 성적 : 참가자의 역대 shake! 최대 등수를 나타내는 정수 S가 주어진다. -1이라면 아직 한 번도 shake!에 진출하지 못하였단 뜻이다. $(1 \le S \le 10^4)$
- 2024 APC 성적 : 2024 APC의 결과 등수를 나타내는 정수 A가 주어진다. $(1 \le A \le N)$

같은 이름을 가지는 참가자들이 있을 수 있으며, 2024 APC에서 같은 등수를 가진 사람은 없다.

출력

첫 번째 줄에 2024 shake!에 참가할 수 있는 진출자들의 수 M을 출력한다.

이후 M개 줄에 걸쳐 참가자의 이름을 사전 순서대로 출력한다.

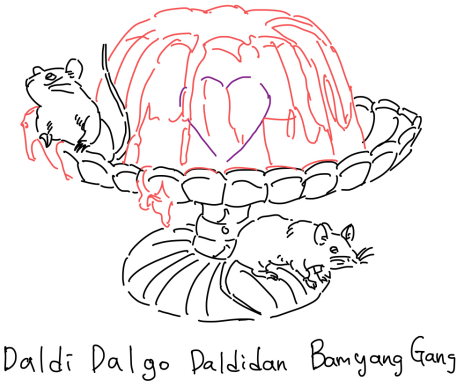
예제

standard input	standard output
20 kim jaehak winner 5 16 park jaehak notyet -1 18 choi hewhak winner 10 17 lee jaehak notyet 20 19 kimyu hewhak notyet 30 20 parkyu jaehak winner -1 15 choiyu hewhak notyet 1 14 leeyu jaehak notyet 9 12 baek jaehak notyet 8 11 shin jaehak notyet 4 13 han jaehak notyet 4 8 oh jaehak notyet 3 9 jang hewhak notyet -1 7 pang jaehak winner 2 10 jaegal jaehak notyet 14 6 sunwoo jaehak notyet 11 5 yang hewhak notyet -1 3 yoon jaehak winner 12 4 moon hewhak notyet -1 2 no hewhak winner 30 1	8 baek han jaegal lee leeyu park shin sunwoo

문제 E. 밤양갱

- 시간 제한 1초

문제



달디달고, 달디달고, 달디단, 밤양갱, 밤양갱

<장기하, 밤양갱, 2024>

민우는 비비의 신곡 <밤양갱>에 꽂혀 하루 종일 "달디달고 달디달고 달디달고... 달디단"이 머릿속을 맴돌고 있다.

민우의 머릿속에선 daldidalgo가 총 N번 반복된 후, 반복이 완료되었다면 daldidan으로 끝나게 된다. 예를 들어 $N = 3$이라면 민우의 머릿속엔 daldidalgodaldidalgodaldidalgodaldidan이 재생된다.

민우는 N이 주어지면 얼마나 빨리 daldidalgodaldidalgo...daldidan을 컴퓨터에 입력할 수 있는지 궁금하다. 매초 민우는 두 개의 작업 중 하나를 선택하여 시행할 수 있다.

알파벳 소문자 a부터 z 중에서 민우가 원하는 알파벳을 하나 골라서 지금까지 입력한 내용의 맨 뒤에 입력한다. 지금까지 입력한 문자열의 연속된 부분 문자열을 복사 후 입력한 내용의 맨 뒤에 붙여넣는다. 예를 들어 지금까지 작성한 문자열이 ajouapcshake라면, ajouapcshake를 복사할 수도, apc를 복사할 수도 있지만, aashake를 복사하여 붙여넣을 수는 없다. 민우는 몇 초 만에 머릿속에 떠오른 가사를 컴퓨터에 입력할 수 있을까?

입력

첫 번째 줄에 민우의 머릿속에 떠오른 daldidalgo의 횟수 N이 주어진다. $(1 \leq N \leq 10^9)$

출력

민우가 문제에 언급된 시행 중 하나를 선택하여 매초 시행했을 때, N번의 daldidalgo를 입력한 후 1번의 daldidan을 입력할 수 있는 최소 시간을 출력한다.

예제

standard input	standard output
2	11

문제 F. 렬정! 렬정! 렬정!

- 시간 제한 1초

문제



원소의 순서만 바꾸던 기존 정렬 알고리즘을 공부하던 민석이는 모든 게 부질없음을 깨닫고 원소의 값을 바꿔서 정렬해 버리기로 했다.

길이 N의 배열 A가 주어진다. 아래의 연산을 $\lfloor \frac{N}{2} \rfloor$번 이하로 사용하여 배열 A의 모든 원소가 내림차순이 되도록 만들어 보자. 여기서 내림차순이란, 1 이상 $N-1$ 이하의 모든 정수 i에 대하여 $A_i \geq A_{i+1}$을 만족하는 상태를 말한다.

- $1 \leq i, j \leq N$를 만족하는 서로 다른 두 정수 i와 j에 대하여 $0 < x \leq 10^6$를 만족하는 정수 x를 선택하여 A_i에 x를 더하고, A_j에 x를 뺀다.

입력

첫 번째 줄에 배열 A의 길이를 나타내는 정수 N이 주어진다. $(1 \leq N \leq 100)$

두 번째 줄에 A의 원소 N개가 공백으로 구분되어 주어진다. A의 모든 원소는 1 이상 $5\,000$ 이하의 정수이다.

출력

첫 번째 줄에 연산을 사용한 횟수 K를 출력한다. K가 최솟값일 필요가 없음에 유의하자.

이후 K개의 줄에 걸쳐 각 연산이 완료된 배열 A의 각 원소를 한 칸의 공백으로 구분하여 출력한다.

가능한 답이 여러 가지라면 그중 아무거나 출력한다.

만약 연산을 1번 이상 $\lfloor \frac{N}{2} \rfloor$번 이하로 사용하여 배열 A를 내림차순으로 만들 수 없다면, 첫 번째 줄에 `-1`을 출력한다.

예제

standard input	standard output
5 30 70 50 60 10	1 80 70 50 10 10
4 1 1 1 1	2 2 1 1 0 3 1 1 -1
4 1 2 3 4	2 6 2 -2 4 6 2 2 0

노트

$\lfloor x \rfloor$는 x를 초과하지 않는 가장 큰 정수를 말한다. 예를 들어$\lfloor 2.5 \rfloor = 2$이고, $\lfloor 3 \rfloor = 3$이다.

아래는 첫 번째 예제에 대한 설명이다.

4번째 원소에서 50을 빼서 1번째 원소에 더한다.

문제 G. 너 재능있어

- 시간 제한 1초

문제

리그 오브 레전드(이하 롤)를 즐기는 민우는 자기보다 롤을 못하는 민석이를 도발했다.

"너? 재능 있어. 열심히 해."

민석이는 너무 분하여 어떻게 하면 점수를 효율적으로 올릴 수 있는지 연구하기 시작했다. 철저한 자기 객관화가 가능한 민석이는 앞으로의 게임에서 N번의 승리와 M번의 패배를 하게 될 것이란 사실을 알고 있다. 민석이는 i번째 승리를 하게 된다면 W_i만큼의 점수를 얻게 되고, j번째 패배를 하게 된다면 L_j만큼의 점수를 잃게 된다. 민석이의 점수는 0점에서 시작하고, 점수는 음수가 될 수 있다.

롤에는 민석이와 같은 하수들이 한 번에 너무 큰 점수를 잃지 않도록 매 K점마다 점수 보호권이 존재한다. 정수 a와 $0 < b < K$를 만족하는 정수 b에 대하여, 민석이의 현재 점수가 $a \times K + b$라고 하자. 만약 민석이가 이 상태에서 j번째 패배를 당했다면, 현재 점수에서 b점을 초과하여 잃지 않는다. 다시 말해, 민석이는 $\min(L_j, b)$만큼의 점수를 잃게 된다. 만약 현재 점수가 $a \times K$라면, L_j만큼의 점수를 잃게 됨에 유의하자.

당연하게도 민석이는 $N + M$번의 게임 후 가장 높은 점수를 얻고 싶다. 민석이를 도와주자.

입력

첫 번째 줄에 승리 시 얻을 수 있는 점수를 나타내는 배열 W의 길이 N이 주어진다. $(1 \le N \le 1\,000)$

두 번째 줄에 W의 원소 N개가 공백으로 구분되어 주어진다. W의 모든 원소는 1 이상 100 이하의 정수다.

세 번째 줄에 패배 시 잃게 되는 점수를 나타내는 배열 L의 길이 M이 주어진다. $(1 \le M \le 1\,000)$

네 번째 줄에 L의 원소 M개가 공백으로 구분되어 주어진다. L의 모든 원소는 1 이상 100 이하의 정수다.

다섯 번째 줄에 점수 보호를 위한 정수 K가 주어진다. $(100 \le K \le 1\,000)$

출력

N번의 승리와 M번의 패배를 한 후 민석이가 얻을 수 있는 점수의 최댓값을 출력한다.

예제

standard input	standard output
4 50 30 70 90 2 100 100 110	160
2 27 51 4 44 62 7 16 101	-23
1 100 1 49 100	51

노트

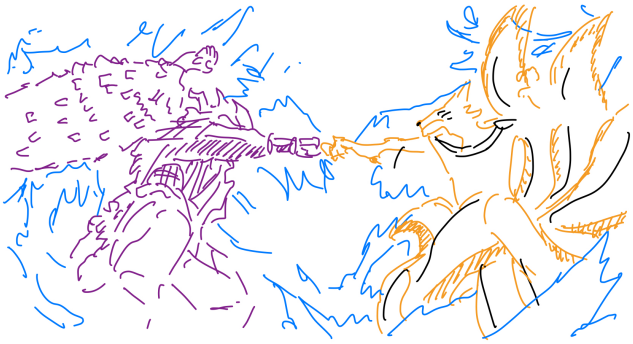
아래는 첫 번째 예시에 대한 설명이다.

민석이의 경기 결과가 순서대로 [승,패,승,패,승,승]일 경우 160점을 얻을 수 있다. 이보다 점수를 더 얻을 수 있는 방법은 존재하지 않는다.

문제 H. 지금부터 서로 죽여라

- 시간 제한 1초

문제



나루토와 사스케는 서로를 죽이기 위한 혈투를 시작했다. 혈투가 시작되기 전, 나루토는 X, 사스케는 Y 의 체력을 가지고 있다. 혈투는 나루토와 사스케의 차례가 번갈아 가면서 진행된다. 차례는 나루토부터 시작되며, 각자 자신의 차례일 때 공격 또는 회복 중 하나를 선택하여 시행한다.

- 공격을 선택하면 상대방의 체력이 자신의 공격력만큼 감소한다.
- 회복을 선택하면 자신의 체력이 자신의 회복력만큼 증가한다.

회복을 선택한 경우, 최대 체력의 상한선은 존재하지 않는다. 즉, 초기의 체력을 넘어갈 수도 있다.

혈투가 진행되는 도중 나루토와 사스케 중 한 명의 체력이 0 이하가 되면, 그 순간 혈투가 종료되고 체력이 0 이하가 된 사람이 패배한다. 만약 10^{100} 번의 차례가 진행된 이후에도 두 명의 체력이 모두 0보다 크다면, 무승부로 혈투가 종료된다.

나루토와 사스케는 혈투에서 이길 수 있다면 이기는 전략을 취하고, 이길 수 없다면 비기기 위해 최선을 다한다. 혈투의 정보가 주어질 때, 결과를 알아내 보자.

입력

첫 번째 줄에 테스트 케이스의 개수 T가 주어진다. 이후 다음과 같은 형태의 입력이 T개 주어진다. $(1 \le T \le 10^3)$

각 테스트 케이스의 첫 번째 줄에 나루토의 체력 X와 사스케의 체력 Y가 공백으로 구분되어 주어진다. $(1 \le X, Y \le 10^{12})$

각 테스트 케이스의 두 번째 줄에 나루토의 공격력 A와 회복력 B, 사스케의 공격력 C와 회복력 D가 공백으로 구분되어 주어진다. $(1 \le A, B, C, D \le 10^{12})$

입력으로 주어지는 모든 수는 정수이다.

출력

각 테스트 케이스에 대해 나루토가 이긴다면 NARUTO, 사스케가 이긴다면 SASUKE, 둘이 비기게 된다면 DRAW를 한 줄에 하나씩 출력한다.

예제

standard input	standard output
2 10 20 15 15 5 5 10 20 10 10 20 5	NARUTO SASUKE

문제 I. 나는 북극곰입니다

- 시간 제한 1.5초

문제



북극은 총 N개의 빙하와 서로 다른 두 빙하를 잇는 M개의 얼음 다리로 이루어져 있다. 각 얼음 다리는 양방향으로 자유롭게 왕복할 수 있으며, i번 얼음 다리의 길이는 d_i이다. 초기에 서로 다른 두 빙하 사이를 왕복할 수 있는 경로가 반드시 존재한다.

북극곰의 사냥터는 1번 빙하이고, 북극곰의 집은 N번 빙하이다. 북극곰은 현재 1번 빙하에서 신나게 연어를 잡아먹고 있다. 그러나 북극곰은 이내 지구 온난화로 인해 얼음 다리가 무너지고 있어 서둘러 N번 빙하에 있는 자신의 집으로 돌아가야 한다는 사실을 깨닫게 되었다. 북극곰은 매초 1마리의 연어를 먹을 수 있고, 매초 길이 1만큼 움직일 수 있다.

북극곰은 이미 무너졌거나 건너는 와중 무너질 얼음 다리를 건널 수 없다. 북극곰은 꽤 민첩하므로 얼음 다리가 무너지는 시점과 얼음 다리를 건너는 걸 완료하는 시점이 일치하는 경우, 얼음 다리를 건널 수 있다. 다시 말해, 현재 k초가 지났고 북극곰이 건너고자 하는 얼음 다리의 길이가 d, 얼음 다리가 t초에 무너질 예정이라면 $k + d \leq t$를 만족해야만 북극곰이 해당 얼음 다리를 건널 수 있다.

연어가 너무 맛있는 나머지, 북극곰은 최대한 늦게까지 사냥터에 남아 연어를 잡아먹고 싶다. 얼음 다리에 대한 정보가 주어질 때, 배고픈 북극곰을 위해 북극곰이 잡아먹을 수 있는 최대 연어의 수를 구해주자.

입력

첫 번째 줄에 빙하의 수 N과 빙하를 잇는 얼음 다리의 수 M이 공백으로 구분되어 주어진다. $(2 \leq N \leq 100\,000; 1 \leq M \leq 100\,000)$

두 번째 줄부터 M개의 줄에 걸쳐 얼음 다리에 대한 정보가 주어진다. 각 줄은 네 개의 정수 u, v, d, t가 공백으로 구분되어 주어지며, 이는 u번 빙하와 v번 빙하를 잇는 다리의 길이는 d이고 t초에 다리가 무너짐을 나타낸다. $(1 \leq u, v \leq N; u \neq v; 1 \leq d \leq 10^9; 1 \leq t \leq 10^9)$

출력

북극곰이 집이 위치한 N번 빙하로 돌아갈 수 있을 때, 먹을 수 있는 최대 연어의 수를 출력한다. 만약, 집으로 돌아갈 수 없는 경우 -1을 출력한다.

예제

standard input	standard output
5 5 1 2 3 3 1 3 3 6 3 4 3 6 2 4 3 9 3 5 3 9	3
5 5 1 2 3 3 1 3 3 3 3 4 3 3 2 4 3 3 3 5 3 3	-1
5 4 1 2 1 4 2 3 1 4 3 4 1 4 4 5 1 4	0

마지막 페이지입니다.