

대회 규칙

- 사용 가능한 언어는 C11, C++17, Java 11, Python 3.7.3입니다.
 - 모든 문제에 대해 제약 조건을 만족하며 정답을 출력하는 C++17 코드가 있음이 보장됩니다.
- 대회는 대회 전용 DomJudge 사이트에서 치뤄지며 문제, 채점 실시간 정보 등을 확인할 수 있습니다.
- 순위는 평균 문제가 많은 순서대로, 평균 문제 수가 같을 경우에는 패널티의 합이 낮은 순으로 정렬됩니다.
 - 문제별 패널티는 '(문제를 풀기까지 걸린 시간(분)) + (그 전까지 제출한 횟수) × 20'입니다.
 - 컴파일 에러는 제출 횟수에 포함되지 않습니다.

금지 / 제한 행위

- 대회가 진행되는 동안 화장실 등을 다녀오는 것은 자유이나, 충분히 이동은 제한됩니다.
- 대회 중도 퇴실은 불가합니다.
- 컴퓨터를 두 대 이상 사용하는 것을 금합니다.
- 운영진에게 질문하는 것 외에 다른 사람과 대화하는 것을 금합니다.
- 사전에 코드를 미리 작성해 와서 사용하는 것을 금합니다.
- 허용된 레퍼런스 페이지를 제외한 메신저, 인터넷 검색, 대화, 이동식 저장 매체를 통한 문제 풀이를 금합니다.
- 문제 제출을 비정상적으로 많이 시도하거나, 의도적으로 대회 웹 서버를 공격하는 행위를 금합니다.

대회 규칙을 어기거나, 운영진이 판단하기에 부정한 행위를 저지를 경우 경고 없이 대회 참가 자격이 박탈될 수 있습니다.

레퍼런스 사이트

다음 레퍼런스 사이트는 열람할 수 있습니다.

- C/C++ : <https://en.cppreference.com/w/>, <http://cplusplus.com>
- Java : <https://docs.oracle.com/en/java/javase/11/docs/api/index.html>
- Python : <https://docs.python.org/3/>

DOMjudge 채점

DOMjudge에 코드를 업로드할 때는 다음 조건을 지켜야 합니다.

- 파일 이름은 알파벳 및 숫자로 시작해야 하며, 알파벳 대소문자 / 숫자 / +.-만 사용 가능합니다.
- 확장자는 C는 .c, C++는 .cpp / cc / cxx / c++, Java는 .java, Python은 .py / .py3여야 합니다.
- 제출한 코드는 표준 입출력만으로 통신하여야 합니다 (파일 입출력은 금지됩니다).
- 제출한 소스코드의 크기는 256 MiB 이하여야 합니다.

DOMjudge 채점 결과

Submit을 한 다음에 Scoreboard 템에서 제출 결과를 확인할 수 있습니다.

- PENDING : 제출되었으며, 채점 대기중이거나 채점중입니다.
- CORRECT : 제출한 코드가 모든 테스트 케이스에 대해 시간 제한 / 메모리 제한 내에서 올바른 답을 내었고, 정상적으로 종료되었습니다. 이 경우 제출자는 해당 문제를 풀었습니다.
- COMPILER-ERROR : 컴파일 과정 중에 에러가 발생하여 채점이 진행되지 않았습니다.
- TIMELIMIT : 프로그램 수행 시간이 제한 시간을 초과하였습니다.
- RUN-ERROR : 프로그램 수행 중 에러가 발생하였습니다. (예시 : 0으로 나누기, 잘못된 주소 참조)
- WRONG-ANSWER : 프로그램이 오답을 출력하였습니다.
- OUTPUT-LIMIT : 프로그램이 지나치게 많은 출력을 하였습니다.

대회 중 ‘request clarification’ 템을 통해 주최진에게 질문을 물을 수 있습니다.

DOMjudge 언어별 시간 제한 설정

일부 언어는 실행 시간 제한에 보정이 있습니다.

- Java : ?배 (미정)
- Python 3 : 3배
- PyPy 3 : 3배

Python 3 코드는 PyPy 3에 제출하는 것을 권장드립니다.

컴파일 옵션

사용 가능한 언어와 컴파일 옵션은 다음과 같습니다. "\$@"는 업로드한 코드 및 생성된 프로그램 이름입니다.

- **C11** (gcc 8.3.0)

컴파일 : gcc -x c -Wall -O2 -std=c11 -static -pipe -o "\$DEST" "\$@" -lm
실행 : exec "\$@"

- **C++17** (g++ 8.3.0)

컴파일 : g++ -x c++ -Wall -O2 -std=c++17 -static -pipe -o "\$DEST" "\$@" -lm
실행 : exec "\$@"

- **Java** (Java 11.0.4)

컴파일 : javac -encoding UTF-8 -sourcepath . -d . "\$@" 2> "\$TMPFILE"
실행 : java -Dfile.encoding=UTF-8 -XX:+UseSerialGC -Xss\${MEMSTACK}k -Xms\${MEMLIMITJAVA}k \ -Xmx\${MEMLIMITJAVA}k '\$MAINCLASS' "\\$@"

– MEMSTACK은 65536이며, MEMLIMIT은 문제의 제한에서 128MB를 뺀 값입니다.

- **Python 3** (Python 3.7.3, PyPy 추가 예정)

컴파일 : python3 -m py_compile "\$@"
실행 : python3 "\$@"

– 속도 측면에서 PyPy로 제출하는 것을 권장합니다.

문제 목록

대회는 3시간 동안 진행되며, 총 9문제로 구성되어 있습니다.

총 문제지가 표지를 제외하고 15쪽인지 확인하시길 바랍니다.

문제의 목록은 다음과 같으며, 출제진이 판단하기에 난이도 순으로 정렬되어 있습니다.

- A 평생 새내기가 되고 싶은 유신이 (first-year)
- B 피카츄 라이츄 (pika)
- C KCPC는 무엇의 약자일까? (acronym)
- D 초코칩 케이크 (cake)
- E 함정에 빠진 이동관 (trap)
- F 대자보 (poster)
- G 신앙 (faith)
- H 얼마나 예뻐? (pretty)
- I ○○ (oo)

문제 A. 평생 새내기가 되고 싶은 유신이 (first-year)

시간 제한: 1초

메모리 제한: 512 MB

매년 이 시기가 되면 새내기는 현내기가 되는 것에 두려움을 느낀다. 심지어 2020년 새내기는 2001년 생이다.

이것에 충격받은 16학번 화석 유신이는 다시 새내기가 되고자 대학교를 재입학하기로 마음먹었다. 유신이는 평생 새내기가 되고 싶지만, 고학년도 되고 싶은 마음이 있기에 다음과 같은 규칙을 가지고 학교 생활을 하기로 했다.

- 어떤 년도에 (재)입학한 사람의 학번은 그 년도의 마지막 2자리이며, 학년은 1학년이다.
- 한 해가 지나면 학년이 하나 증가한다. 단, 4학년이 된 해의 다음 해는 새내기(1학년)로 재입학을 한다.

예를 들면 유신이의 앞으로의 일정은 다음과 같다.

- 유신이는 2020년에 재입학하면 20학번 1학년이 된다.
- 2021년에는 20학번 2학년이 된다.
- 2022년에는 20학번 3학년이 된다.
- 2023년에는 20학번 4학년이 된다.
- 4학년이 된 유신이는 다음해에 다시 재입학한다.
- 즉 유신이는 2024년에 24학번 1학년이 된다.

유신이는 금수저기 때문에 이 과정으로 노년까지 하고자 한다. 유신이는 X 년에 몇 학번, 몇 학년일까? 단, 유신이는 2016년도에 1학년이라고 가정한다.

입력 형식

첫 번째 줄에는 유신이의 학번이 궁금한 년도 X 가 입력으로 들어온다. ($2016 \leq X \leq 2100$)

출력 형식

첫 번째 줄에 유신이의 학번 두 자리와 학년 한 자리를 공백으로 구분하여 출력한다.

예제

표준 입력	표준 출력
2016	16 1
2019	16 4
2100	00 1

문제 B. 피카츄 라이츄 (pika)

시간 제한: 1초

메모리 제한: 512 MB

포켓몬스터의 주인공은 지우와 피카츄이다. 피카츄는 N 번 대회를 치룰 예정인데, 대회를 치루면 필연적으로 경험치를 획득하게 된다. 그러나 피카츄는 지우에게 항상 피카츄로 남고 싶기 때문에 라이츄로 변신하지 않길 원한다.

피카츄는 경험치가 X 이상이면 라이츄로 진화하며, 한 번 라이츄로 진화하면 다시 피카츄로 되돌아가지 않는다. 또 피카츄는 휴식 차원에서 매일 밤 A 만큼 경험치가 감소하나, 0보다 작아지지는 않는다. 앞으로 피카츄가 참여할 대회의 날짜와 획득하는 경험치가 주어질 때, 대회를 모두 마치고 난 후 피카츄가 진화했는지 아닌지의 유무를 판단해보자.

피카츄의 현재 경험치는 0이며, 모든 대회는 낮에 열린다.

입력 형식

첫 번째 줄에는 대회의 수 N , 라이츄로 진화하기 위해 필요한 경험치 X , 매일 밤 사라지는 경험치 A 가 공백으로 구분되어 주어진다. ($1 \leq N \leq 100, 1 \leq X \leq 1,000, 1 \leq A \leq 100$)

두 번째 줄부터 $N + 1$ 번째 줄까지 대회의 날짜 d_i 와 대회 이후 획득하는 경험치 e_i 가 공백으로 구분되어 주어진다. ($1 \leq d_i \leq 100, 1 \leq e_i \leq 100$) 대회 날짜는 모두 다르며, 오름차순으로 주어진다.

입력으로 들어오는 수는 모두 정수이다.

출력 형식

첫 번째 줄에 대회를 진행하며 피카츄가 라이츄로 진화했으면 'Raichu'를, 대회가 다 끝나도 피카츄로 남아있으면 'Pikachu'를 따옴표를 제외하고 출력한다.

예제

표준 입력	표준 출력
3 8 3	Raichu
1 6	
2 3	
3 5	
1 10 2	Pikachu
3 4	

설명

첫 번째 예제에 대한 설명은 다음과 같다.

- 1일차 낮에 보유한 경험치는 6이며, 밤에 보유한 경험치는 3이다.
- 2일차 낮에 보유한 경험치는 6이며, 밤에 보유한 경험치는 3이다.
- 3일차 낮에 보유한 경험치는 8이며, 진화한다.

두 번째 예제에 대한 설명은 다음과 같다.

- 2일차 밤까지 보유한 경험치는 0이다.
- 3일차 낮에 보유한 경험치는 4이다. 이 이후로는 경험치가 감소하기만 하므로 진화하지 못한다.

문제 C. KCPC는 무엇의 약자일까? (acronym)

시간 제한: 1초

메모리 제한: 512 MB

UCPC와는 다르게 KCPC는 무엇의 약자인지 이미 알려져 있다. Korea University Collegiate Programming Contest가 바로 그것인데, 이미 KCPC가 무엇의 약자인지 알고 있는 상황에서 'KCPC는 무엇의 약어일까'를 내려고 한 박홍빈은 김이 팍 빠졌다. 하지만 박홍빈은 그럼에도 문제를 내고 싶었기에 문제를 약간 수정했다. 문자열의 축약이란 문자열에서 임의의 문자들을 제거하고 남은 문자들을 순서를 유지하며 이어붙여 새로운 문자열을 만드는 과정으로 예시는 다음과 같다.

- "ABCDE" → "BD"
- "KoreaUniversityCollegiateProgrammingContest" → "KCPC"

따옴표 "는 문자열의 경계를 표현하기 위한 것이지, 문자열의 일부가 아니다.

어떤 문자열 S 를 축약해서 "KCPC"로 만들 수 있는지 확인하는 것은 이미 있는 문제이니 "KCPC"로도 주어진 문자열 A 로도 축약 가능한지 확인해 보기로 하였다.

문제에서 요구하는 것은 문자열 A 와 S 가 주어졌을 때, 문자열 S 를 축약해서 문자열 A 를 만들 수 있는지, 또 문자열 S 를 축약해서 "KCPC"로도 만들 수 있는지 확인하는 것이다.

대문자와 소문자는 다른 문자로 취급하며, 축약 과정은 독립적이다.

입력 형식

첫 번째 줄에 알파벳 대문자와 소문자로만 이루어진 문자열 A 가 주어진다.

두 번째 줄에 알파벳 대문자와 소문자로만 이루어진 문자열 S 가 주어진다.

각각의 문자열 길이는 최대 1000자이다.

출력 형식

첫 번째 줄에 문자열 S 를 축약하여 문자열 A 와 "KCPC"를 만들 수 있다면 "KCPC!"를 그렇지 않다면 "KCPC?"를 따옴표를 제외하고 출력한다.

예제

표준 입력	표준 출력
LOVE ILOVEKCPC	KCPC!
Cat KcpCanDstRingS	KCPC?
CCCC KCPCKCPC	KCPC!

설명

첫 번째 예제의 경우 ILOVEKCPC와 ILOVEKCPC처럼 LOVE로도 KCPC로도 축약할 수 있다.

두 번째 예제의 경우 KcpCanDstRingS처럼 Cat으로 축약할 수는 있지만, KCPC로 축약할 수는 없다. 알파벳 대소문자를 구문함에 유의해야 한다.

세 번째 예제의 경우 KCPCKCPC와 KCPCKCPC처럼 CCCC로도 KCPC로도 축약할 수 있다.

문제 D. 초코칩 케이크 (cake)

시간 제한: 1초

메모리 제한: 512 MB

2019년도 어언 한 달밖에 남지 않았다. 다사다난했던 2019년을 기념하기 위하여 상현이는 고려대학교 프로그래밍 경진대회 운영진과 출제진과 나누어먹을 케이크를 장식하고자 한다.

균형과 일관성을 극도로 중시하는 상현이는 정사각형 케이크를 샀고, 이를 가로 n 줄 세로 n 줄이 되게 조각으로 만들었다. 미니멀리즘이 듬뿍 담겨있는 이 케이크 위에는 아무 장식도 뿐여져있지 않다. 너무 맛잇다고 생각한 상현이는 케이크에 초코칩을 다음과 같이 여러 번 뿐여 장식을 하려고 한다.

- 한 가로줄을 선택하여, 이 가로줄에 속한 모든 조각에 초코칩을 1개 추가한다.
- 한 세로줄을 선택하여, 이 세로줄에 속한 모든 조각에 초코칩을 1개 추가한다.

상현이는 가장 초코칩이 많이 뿐여져 있는 조각을 ‘가장 맛있는 조각’이라고 부른다.

매번 장식을 하는 상현이는 가장 맛있는 조각의 개수를 신경쓰지 않을 수 없다. 다행히 상현이는 프로그램을 작성하여 자신의 행동에 따른 가장 맛있는 조각의 개수를 성공적으로 계산해내었고, 이를 다음과 같이 문제로 만들었다. 케이크에 장식을 올리는 느낌으로 문제를 풀어보자.

입력 형식

첫 번째 줄에 두 정수 n, q 가 공백으로 구분되어 주어진다. ($1 \leq n, q \leq 30,000$)

n 은 가로줄과 세로줄의 개수이며, q 는 장식을 하는 횟수이다.

이후로 q 개의 줄에 두 정수 t, a 가 공백으로 구분되어 주어진다. ($1 \leq t \leq 2, 1 \leq a \leq n$)

t 가 1이면 a 번째 가로줄에, 2이면 a 번째 세로줄에 있는 조각들에 초코칩을 하나씩 더한다.

출력 형식

각 줄마다 매 번 장식을 한 이후 그 상태의 가장 맛있는 조각의 개수를 출력한다.

모든 초코칩은 장식이 끝난 후에도 유지된다.

예제

표준 입력	표준 출력
3 2	3
1 1	6
1 3	
1 3	1
1 1	1
2 1	1
1 1	
4 5	4
1 1	8
1 4	2
2 3	1
1 4	2
2 2	

설명

첫 번째 예제에 대한 설명은 다음과 같다.

- 첫 번째 장식은 1번째 가로줄에 초코칩을 뿌리는 것이므로, 3개의 조각에 초코칩이 1개, 6개의 조각에 초코칩이 0개 있어 가장 맛있는 조각은 3개이다..
- 두 번째 장식까지 하고 나면 6개의 조각에 초코칩이 3개, 나머지 3개의 조각에 초코칩이 0개 있어 가장 맛있는 조각은 6개이다.

두 번째 예제에선 조각이 1개밖에 없으므로, 장식 스타일과 상관없이 가장 맛있는 조각도 1개밖에 없다.

문제 E. 함정에 빠진 이동관 (trap)

시간 제한: 1초

메모리 제한: 512 MB

이동관은 $N \times M$ 격자 모양의 함정에 빠졌다. 모든 격자에는 숫자가 적혀있는 트램펄린이 존재하고, 트램펄린을 통해서만 다른 격자로 이동할 수 있다.

트램펄린의 이동 규칙은 다음과 같다.

- 함정 밖으로 이동할 수는 없다. 새로 도착하는 위치를 (a, b) 라고 하면, $1 \leq a \leq N$ 과 $1 \leq b \leq M$ 를 만족해야 한다.
- 트램펄린에 x 가 적혀 있다면, 상하좌우로 x 칸 떨어진 곳으로만 이동할 수 있다. 즉, 현재 위치가 (a, b) 라면, $(a + x, b)$, $(a - x, b)$, $(a, b + x)$, $(a, b - x)$ 로만 이동할 수 있다. 단, 위에 언급된 것처럼 함정 밖으로는 이동할 수 없다.

한 번 이동하는데 1만큼의 시간이 걸린다고 할 때, 탈출구까지 도달하는데 걸리는 최단 시간을 구하여라. 이동관의 출발 위치는 항상 $(1, 1)$ 이다.

입력 형식

첫 번째 줄에 함정의 행의 개수 N 과 열의 개수 M 이 공백으로 구분되어 주어진다. ($1 \leq N, M \leq 1,000$)

이후 N 개의 줄에 걸쳐 M 개의 자연수 a_1, a_2, \dots, a_m 이 공백으로 구분되어 주어진다. ($1 \leq a_j \leq \min(N, M)$)
 $i+1$ 번째 줄의 a_j 는 (i, j) 에 있는 트램펄린의 수이다.

마지막 줄에는 두 자연수 x, y 가 공백으로 구분되어 주어진다. ($1 \leq x \leq N, 1 \leq y \leq M$) 이는 탈출구가 (x, y) 에 존재함을 의미한다. 출발 위치가 탈출구일 수도 있다.

출력 형식

첫 번째 줄에 동관이가 탈출할 수 있다면 탈출하는데 걸리는 시간을 출력한다.

만약 탈출할 수 없으면, -1을 출력한다.

예제

표준 입력	표준 출력
3 3 2 1 2 1 1 1 2 1 2 2 2	-1
2 3 1 1 1 2 1 1 2 3	2

설명

첫 번째 예제에서, 동관이는 $(1, 1)$, $(1, 3)$, $(3, 1)$, $(3, 3)$ 이외의 격자에는 갈 수 없기 때문에 평생 트램펄린을 탈 것이고, 따라서 탈출구까지 갈 수 없다.

두 번째 예제는 $(1, 1) \rightarrow (2, 1) \rightarrow (2, 3)$ 로 간다면 2번만에 이동할 수 있다. 이보다 빠른 방법은 없다.

문제 F. 대자보 (poster)

시간 제한: 1초

메모리 제한: 512 MB

고려대학교에는 학생들의 소통을 위한 게시판이 설치되어 있다. 고려대학교 학생들은 게시판에 대자보를 붙여 자신의 신념을 밝히기도 하고, 비리나 부당한 상황을 고발하기도 한다.

그러던 어느날 게시판에 숫자로만 이루어진 긴 수열의 대자보가 붙었다. 사람들은 며칠간 수열에 담긴 의미를 찾아보려고 노력했지만, 그 누구도 해결하지 못했고 결국 해당 대자보는 철거되었다. 하지만 같은 형식의 수열로 이루어진 대자보가 몇차례 등장하였고, 사람들은 이를 누군가의 장난이라고 여기게 되었다.

하지만 ALPS의 회장 이세정은 그 정답을 예전부터 알고 있었다. 해당 수열은 알파벳 소문자로 이루어진 단어를 아래 방법을 따라 변환시킨 것이다.

1. 단어의 각 알파벳 소문자를 아래 표와 같이 1에서 26으로 변환한다.
2. 이웃한 두 숫자를 곱한 결과를 한 글자씩 나열한다. 예를 들어, `apple`의 각 알파벳은 1 16 16 12 5 으로 변환되고, 한 자씩 나열하면 1 6 2 5 6 1 9 2 6 0이 된다.

알파벳	숫자								
a	1	b	2	c	3	d	4	e	5
g	7	h	8	i	9	j	10	k	11
m	13	n	14	o	15	p	16	q	17
s	19	t	20	u	21	v	22	w	23
y	25	z	26					x	24

세정이는 해당 대자보가 올라올 때마다 대자보를 해석하면서, 대자보를 붙인 사람에 대한 증거를 모으고 있다. 하지만 한 대자보의 수열이 수많은 알파벳 문자열로 변환이 될 수 있다. 일단 세정이는 가능한 원본 문자열의 개수를 구해보고자 한다. 값이 매우 커질 수 있으므로, 1,000,000,007로 나눈 나머지를 구하자.

입력 형식

첫 번째 줄에 수열의 길이인 정수 N 이 주어진다. ($3 \leq N \leq 300,000$)

두 번째 줄에 N 개의 한 자리 정수 a_1, a_2, \dots, a_N 이 공백으로 구분되어 주어진다. ($0 \leq a_i \leq 9$)

a_i 는 수열의 i 번째 숫자를 의미한다. 첫 번째 숫자가 0인 경우는 주어지지 않는다.

출력 형식

주어진 수열에 대해 세정이가 확인해야 하는 단어의 수를 1,000,000,007로 나눈 나머지를 출력한다.

예제

표준 입력	표준 출력
3 4 4 4	7
6 3 6 3 6 3 6	12
10 2 8 2 1 9 0 9 2 2 4	0

문제 G. 신앙 (faith)

시간 제한: 1 초

메모리 제한: 512 MB

새내기 시절 안수빈은 중앙광장에서 코딩하던 도중 노트북 배터리를 모두 소모한 적이 있다. 이에 수빈이는 화를 참지 못하고 발을 세게 굴렸고, 그 때의 여파로 3층 건물 하나스퀘어는 지하에 박히게 되었다. 그 때의 여파로 노벨 광장 한복판에 박힌 석조물은 아직까지도 뽑히지 않아 지금은 랜드마크로 자리잡게 되었다.

하지만 그런 비화를 모르는 수많은 사람들은 안암에 토착 신이 존재한다고 믿게 되었다. 토착 신을 믿는 신도들은 토착 신에게 기도를 드리며 신앙을 증명하며, 한편으로 하나스퀘어 사태 복구를 위한 성금을 주기적으로 낸다. 각자의 신앙심 f_i 나 현금 금액 m_i 는 양의 정수로 표현된다.

이들을 이끄는 사제는 신도들의 마음을 조종할 수 있다. 사제는 빨간 약과 파란 약을 이용해서 신앙심이나 내고자 하는 성금을 조작할 수 있다.

- 빨간 약: 먹은 사람의 신앙심의 값을 2배로 증가시킨다. ($f_i := 2f_i$)
- 파란 약: 먹은 사람의 현금 금액을 신앙심의 값으로 바꾼다. ($m_i := f_i$)

사제가 빨간 약과 파란 약을 신도들에게 적절히 먹였을 때 얻을 수 있는 현금 총합을 최댓값을 구해라. 약을 모두 사용할 필요는 없다.

입력 형식

첫째 줄에 신도의 수 N , 빨간 약의 수 R , 파란 약의 수 B 가 공백으로 구분되어 주어진다. ($1 \leq N \leq 1,000$, $0 \leq R \leq 20$, $0 \leq B \leq 1,000$)

이후 N 개의 줄에 걸쳐 두 정수 f_i 와 m_i 가 공백으로 구분되어 주어진다. ($0 \leq f_i, m_i \leq 100,000$) 이는 i 번째 신도의 신앙심과 현금 금액을 의미한다.

출력 형식

첫 번째 줄에 사제가 빨간 약과 파란 약을 신도들에게 적절히 먹였을 때 얻을 수 있는 현금 총합을 최댓값을 출력한다.

예제

표준 입력	표준 출력
3 1 2 3 5 3 1 2 2	13
5 2 2 8 7 5 2 17 100 10 12 5 5	157

문제 H. 얼마나 예뻐? (pretty)

시간 제한: 1초
메모리 제한: 512 MB

문제 조건 변경됨.

12월 25일은 크리스마스이다. 크리스마스를 맞은 피카츄는 지우를 위해 트리를 만들기로 했다. 금전적인 문제로 살아있는 나무를 구할 수 없게 된 피카츄는 나무 대신 그래프의 일종인 트리를 꾸미기로 했다.

트리란 모든 정점이 연결되어 있으며, 한 정점에서 다른 정점으로 가는 경로가 유일한 그래프이다. 트리에서 간선으로 연결된 두 정점 u 와 v 에 대해 u 가 루트 정점에 더 가까운 정점이라고 할 때, v 의 부모는 u 이며 u 는 v 를 자식으로 가진다. 트리에 루트 정점은 단 하나 존재하며, 정의에 의해 루트 정점은 부모 정점이 없다. 정점 x 의 서브트리란 x 와 x 의 자식들의 서브트리로 구성된 트리를 의미한다.

정점 x 에서의 전위 순회는 트리를 특정한 규칙에 의해 방문하는 순서로, 다음과 같이 정의된다.

- 정점 x 를 방문한다.
- 자식 정점에서 정확히 한 번 전위 순회를 하되, 그 순서는 조절할 수 있다.

전위 순회의 정의에 의해, 한 자식 정점에서의 전위 순회를 하는 도중에는 다른 자식의 서브트리로 이동할 수 없다. 추가적으로, 트리의 전위 순회는 루트 정점에서의 전위 순회로 정의한다.

한 정점에서의 전위 순회는 그 서브트리의 모든 정점을 정확히 한 번만 방문하기 때문에 의미가 있다. 다만 정점의 자식이 많을 경우, 전위 순회가 유일하지 않을 수 있다.

수학적 정의가 귀찮았지만 트리를 그리는 건 좋았던 피카츄는 인터넷 검색을 통해 1번 정점이 루트인 트리를 그렸다. 트리를 더 장식하기 위해 각 정점에 괄호(여는 괄호 '(' 또는 닫는 괄호 ')')를 한 개씩 그렸는데, 트리를 본 지우는 정점이 다음의 성질을 만족하면 아름답다고 정의했다.

- 정점 x 에서의 전위 순회를 하며, 방문한 순서대로 각 정점에 부여된 괄호를 추가하가며 문자열을 얻을 수 있다.
- 이렇게 만들 수 있는 괄호 문자열 중 올바른 괄호 문자열이 존재한다면 x 는 아름답다.

올바른 괄호 문자열은 다음과 같이 정의된다.

1. 빈 문자열은 올바른 괄호 문자열이다.
2. A가 올바른 괄호 문자열이면 괄호를 씌운 (A)도 올바른 괄호 문자열이다.
3. A와 B가 올바른 괄호 문자열이면 둘을 이어붙인 AB도 올바른 괄호 문자열이다.

지우는 트리의 아름다움을 트리의 정점 중 아름다운 정점의 수라고 정의했다. 피카츄를 도와 피카츄가 꾸민 트리가 얼마나 아름다운지 구하자.

입력 형식

첫 번째 줄에 노드의 개수 N ($2 \leq N \leq 10^5$)이 주어진다.

두 번째 줄에는 0과 1로만 이루어진 N 개의 정수 a_1, a_2, \dots, a_N 가 공백으로 구분되어 주어진다. a_i 가 0이면 정점 i 에 여는 괄호 '('가 적혀 있으며, 1이면 ')'이 적혀 있다.

이후 $N - 1$ 개 줄에 걸쳐, 두 정수 a 와 b 가 공백으로 구분되어 주어진다. 이는 정점 a 와 정점 b 가 간선으로 연결되어 있음을 의미한다.

입력으로 들어오는 그래프는 트리이다.

출력 형식

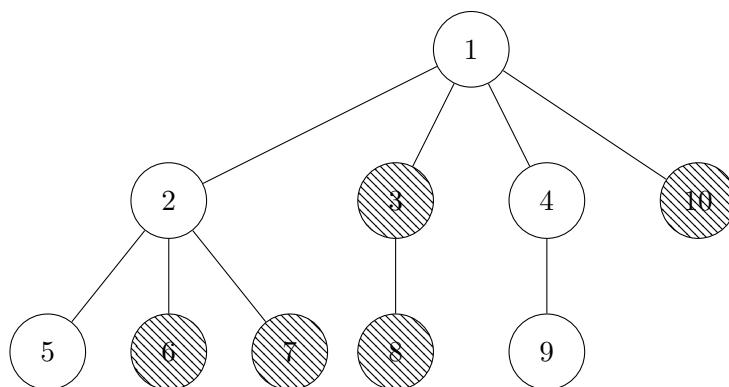
첫 번째 줄에 피카츄가 만든 트리의 아름다움을 출력한다.

예제

표준 입력	표준 출력
10 0 0 1 0 0 1 1 1 0 1 1 2 1 3 1 4 1 10 2 5 2 6 2 7 3 8 4 9	2

설명

첫 번째 예제에서 닫는 괄호가 있는 칸에 빗금을 쳐서 그리면 다음과 같다.



정점 1에서의 전위 순회 중 $1 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 10 \rightarrow 4 \rightarrow 9 \rightarrow 3 \rightarrow 8$ 은 올바른 팔호 문자열 $((())(())$ 을 생성하므로, 아름다운 정점이다.

$1 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 9 \rightarrow 3 \rightarrow 8 \rightarrow 6 \rightarrow 7 \rightarrow 10$ 은 겉보기에는 올바른 팔호 문자열 $((((())($)을 생성하는 것으로 보일 수 있다. 정점 2에서의 전위 순회 중 정점 2의 서브트리에 속하지 않은 정점 4로 이동하므로, 전위 순회가 아니다. 전위 순회를 통해 생성된 팔호 문자열만 고려해야 한다.

정점 2에서의 전위 순회 중 $2 \rightarrow 6 \rightarrow 5 \rightarrow 7$ 은 올바른 팔호 문자열 $(()()$ 을 생성하므로, 아름다운 정점이다. 그 외에 아름다운 정점은 존재하지 않는다.

문제 I. ○○ (oo)

시간 제한: 1초

메모리 제한: 512 MB

영수의 이름에 ○이 2개 들어가 있다는 점에서 알 수 있듯 영수는 2개의 원을 좋아한다.

영수는 좌표평면에 N 개의 점을 찍었다.

이 N 개의 점은 적절한 2개의 원을 정해서 모든 점이 적어도 어느 하나의 원 위에 존재하게끔 할 수 있음이 보장된다.

N 개의 점이 주어질 때 모든 점이 적어도 어느 하나의 원 위에 존재하게끔 하는 2개의 원을 구해보자.

입력 형식

첫번째 줄에 N 이 주어진다. ($1 \leq N \leq 20$)

둘째 줄부터 N 개의 줄에 걸쳐 실수 x, y 가 소숫점 아래 15자리까지 공백으로 구분되어 주어진다. (x, y) 는 점의 좌표를 의미한다. ($-10^6 \leq x, y \leq 10^6$)

임의의 두 점 사이의 거리가 1.1 이상임이 보장된다.

출력할 때 x, y, r 의 소숫점 아래 자리수는 제한이 없지만 C++ double로 읽어들여 답을 판정하기 때문에 16자리부터는 의미가 없다.

출력 형식

두 줄에 걸쳐 원의 중심의 좌표 x, y 와 반지름 r 을 공백으로 구분하여 출력한다. 이 때 x, y, r 은 실수이고 $-10^6 \leq x, y \leq 10^6$, $0.5 \leq r \leq 10^7$ 을 만족해야 한다.

답이 여러 개일 경우 그 중에서 아무거나 출력하면 된다.

예제

표준 입력	표준 출력
3 3 2 1 2 1 1 1 2 1 2 2 2	-1
2 3 1 1 1 2 1 1 2 3	2

참고 사항

원의 중심과 점의 거리를 d 라고 할 때, d 와 반지름 r 의 절대 혹은 상대 오차가 10^{-3} 이하일 때 점은 원 위에 있다고 판단된다.

N 개의 점 모두를 서로 다른 임의의 방향으로 최대 10^{-6} 만큼 움직여도 여전히 답이 존재함이 보장된다.

두 원이 모두 $-10^6 + 10^{-2} \leq x, y \leq 10^6 - 10^{-2}$, $0.51 \leq r \leq 10^7 - 10^{-2}$ 조건을 만족하는 답이 존재함이 보장된다.