대회 안내

1. 대회 관련

- 각 문제를 해결할 때마다 ({대회 후 경과시간} + {해당 문제 오답 횟수x20분})의 패널티가 가산됩니다.
- 순위는 문제를 많이 푼 순 > 패널티가 적은 순 > 마지막 정답이 빠른 순으로 정해집니다.
- 문의하기 기능을 사용해 문제와 직접적으로 관련된 사항에 대해 질문할 수 있습니다.

2. 제출관련

- 제출시 문제 번호와 사용 언어를 정확히 선택해주세요. 참가자의 실수에 대해 책임지지 않습니다.
- 제출한 코드는 채점 현황에서 확인할 수 있습니다.

3. 채점 관련

- 표준 입출력 이외의 파일을 읽거나 쓰려고 시도하는 경우 런타임 에러로 처리하여 오답으로 인정합니다.
- 병렬연산이나 스레드를 이용하면 안됩니다. 모든 프로그램은 단일 스레드 프로그램이어야 합니다.
- 시스템에 영향을 주는 기능(exit, sleep 등)을 사용하면 무조건 오답처리 합니다.
- 채점 입력 데이터 모두에 대해 올바른 답을 출력하여야 합니다. 부분 점수는 없습니다.
- 예제 데이터는 하나의 예제 일 뿐입니다. 실제 채점은 수 많은 비공개 데이터들로 이루어집니다.
- 각 채점 결과의 의미는 아래와 같습니다. 감독관은 오답의 이유를 절대 알려드리지 않습니다.
  - **기다리는 중** : 채점을 기다리는 중입니다. 조금 기다리면 채점 결과를 받을 수 있습니다.
  - 재채점을 기다리는 중 : 재채점을 기다리는 중입니다.
  - **채점 준비중** : 채점 프로그램이 채점을 하기 위해 여러가지 작업을 수행하는 중 입니다. 소스 코드 컴파일, 실행 환경 세팅 등이 포함됩니다.
  - **채점중** : 채점을 하는 중입니다.
  - **맞았습니다!!** : 제출한 프로그램이 모든 데이터를 맞은 경우에 받게 됩니다.
  - **출력 형식이 잘못되었습니다** : 답을 올바르게 구했으나, 문제에 나와있는 출력 형식과 다른 경우입니다. 다음은 이 결과를 받게 되는 일부 예시입니다.
    - 줄 뒤에 의미 없는 공백을 출력하는 경우. 단, 공백 한 칸은 무시하고 채점합니다. 예를 들어, "1 2"를 출력해야 하는 경우에 "1 2"와 "1 2 "는 를 받게 됩니다. "1 2 "는 공백 두 칸을 뒤에 출력해서, " 1 2"는 첫 문자가 공백이라서, "1 2"는 두 수를 공백 두 칸으로 구분했기 때문에, 를 받게 됩니다. 여기서 "는 공백을 보기 위해 편의상 삽입한 기호입니다.
    - 의미없는 빈 줄을 삽입하는 경우. 예를 들어, 1부터 N까지 한 줄에 하나씩 출력해야 하는 경우에 1부터 N을 한줄에 하나씩 공백으로 구분해서 출력하는 경우는 를 받게 됩니다. 이것은 사이트가 생겼을 때 부터 있던 채점 시스템의 버그로 수정될 예정입니다. 각 줄의 사이에 빈 줄을 출력하는 경우에는 를 받게 됩니다.
  - **틀렸습니다** : 출력 결과가 정답과 다른 경우입니다.
  - **시간 초과** : 프로그램이 제한된 시간 이내에 끝나지 않은 경우입니다. 시간 제한을 초과하면 실행을 즉시 중단합니다. 따라서, 정답이 맞았는지 틀렸는지는 알 수 없습니다.
  - **메모리 초과** : 프로그램이 허용된 메모리보다 많은 메모리를 사용한 경우입니다. 와 마찬가지로, 메모리 제한을 초과하면 실행을 즉시 중단합니다.
  - **출력 초과** : 프로그램이 너무 많은 출력을 하는 경우에 발생합니다. 이 결과는 와 같은 의미를 갖습니다. 정확하게는 미리 구해놓은 정답 파일 크기의 2배를 넘어가면 이 결과를 받게 됩니다. 스페셜 저지는 미리 구해놓은 정답 파일이 없을 수도 있기 때문에, 이 결과를 받을 수 없습니다. 만약, 출력하는 부분이 무한 루프에 빠진 경우, 시간을 초과하기 전에 를 받을 수 있습니다.

- **런타임 에러** : 프로그램이 비정상적으로 종료한 경우입니다. Exit code가 0이 아닌 경우, `segmentation fault`를 받은 경우가 대표적입니다.
- **컴파일 에러** : 컴파일에 실패한 경우입니다. 경고(Warning)가 있어도 컴파일에 성공하면 이 결과를 받지 않습니다. 를 클릭하면 컴파일 에러 메시지를 볼 수 있습니다.

4. C/C++ 관련

- 모든 입출력은 stdin, stdout을 통한 표준 입출력만 사용됩니다.
- GCC 컴파일러를 통해 채점됩니다. 비표준 MS C/C++ 라이브러리를 사용할 경우 컴파일 에러가 발생할 수 있습니다.
- C++의 cin/cout은 scanf/printf에 비해 속도가 현저히 느립니다. 채점 시간에 반영되므로 후자의 이용을 권장합니다.

5. Java 관련

- 소스코드에 Package 선언문을 절대로 넣지 마세요. 오답처리 됩니다.
- 모든 입출력은 system.in, System.out을 통한 표준 입출력만 사용됩니다.
- Java의 특성상 문자열 처리 시 StringBuilder를 활용하는 편이 속도면에서 매우 유리합니다.
- system.in 입력 시 Scanner보다 BufferedReader와 InputStreamReader를 활용하는 편이 훨씬 속도가 빠릅니다.

C/C++ 레퍼런스 사이트 : http://en.cppreference.com/w/
Java 레퍼런스 사이트 : http://docs.oracle.com/javase/7/docs/
Python 레퍼런스 사이트 : http://python.org/

Problem A
다항함수의 미분
시간 제한 : 1 Second

다항식(polynomial)은 문자의 거듭제곱의 상수 배들의 합을 표현하는 수식이다. 예를 들어 $x^2 - 2x + 3$ 같은 식을 의미한다. 그 중 변수가 하나인 것을 일변수 다항식이라고 하고 보통 다음과 같이 표현한다.

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$$

최대 일차 일변수 다항식이 주어졌을 때 그 함수를 미분한 결과를 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 내림차순으로 최대 일차 일변수 다항식이 주어진다. 항의 개수는 최대 2개이고, 변수는 항상 x로 주어지며, 각 항은 공백 문자로 구분되지 않는다. 주어지는 계수와 상수의 절댓값의 최댓값은 10,000을 넘지 않는 정수이다. 단, 계수의 절댓값이 1인 경우에는 계수를 생략한다.

출력

주어진 일변수 다항식을 미분한 결과를 출력한다.

예제 입력	예제 출력
6x-6	6

힌트

문제에서 다루는 함수인 멱함수의 미분법은 $(x^n)' = n\,x^{n-1}$ 를 사용한다.

또한 미분 가능 함수에 대하여 합의 법칙, $(f(x) + g(x))' = f'(x) + g'(x)$ 가 성립한다.

Problem B
이칙연산
시간 제한 : 1 Second

다음과 같이 세 수가 연속해서 주어졌을 때 한 번의 곱셈 기호와 한 번의 나눗셈 기호를 이용하여 만든 식 중 가장 큰 값을 출력하는 프로그램을 작성하시오. (세 수의 순서는 변하지 않는다.)

입력

첫째 줄에 세 개 정수 A, B, C(1 <= A, B, C <= 1,000,000)가 주어진다. 답은 int범위를 벗어나지 않는다.

출력

나올 수 있는 가장 큰 값을 출력한다. (단 소수점 아래는 버린다. 1e-9 오차 내에서의 버림은 허용한다.)

예제 입력	예제 출력
32 16 8	64

Problem C
조별과제를 하려는데 조장이 사라졌다
시간 제한 : 1 Second

3학년 1학기를 재학중인 성우는 '빨간눈 초파리의 뒷다리 털의 개수와 파인애플 껍질의 이해'라는 과목을 수강중이다. 기말고사를 맞이하여 교수님은 수강생들에게 조별과제를 내주었고, 그 내용은 다음과 같다.

> 중간고사 이전에 배운 빨간눈 초파리의 뒷다리 털의 개수를 구하는 방법을 이용하여,
> 파인애플 껍질의 두께를 구하는 공식을 과학적인 근거와 함께 A4용지 10장 이상으로 제출하시오. [30%]

하지만 성우의 조 조장인 민건은 과제 내용을 듣자마자 집으로 도망쳐 버렸고, 성우는 민건이를 찾기위해 떠난다. 성우는 1분에 1에서 5까지의 거리를 이동할 수 있다. 성우가 있는 곳으로부터 민건이의 집까지 거리가 주어졌을 때, 정확히 몇 분만에 민건이를 찾을 수 있는지 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 성우의 현재위치와 민건이의 집까지의 거리 L(1 ≤ L ≤ 1,000,000)가 주어진다.

출력

성우가 몇 분만에 민건이를 찾을 수 있는지 출력한다.

예제 입력	예제 출력
12	3

Problem D
에리 - 카드
시간 제한 : 1 Second

2468년 개최된 해왕성 올림픽, '에리 - 카드'는 드디어 올림픽 정식 종목으로 채택된다. '에리 - 카드'는 N 장의 '공유 숫자카드'와 N 장의 '팀 숫자카드'를 받고 시작한다. 상대 팀은 우리 팀의 '팀 숫자카드' K 장을 견제할 수 있는데, 견제된 카드는 낼 수 없게 된다. 모든 견제가 마친 후 우리 팀은 '공유 숫자카드'에서 한 장, '팀 숫자카드' 중 한 장씩을 골라 내게 되는데, 두 카드의 곱이 우리 팀의 점수가 되며 이후 같은 방식으로 상대 팀도 진행하여 상대 팀의 점수보다 높을 경우 이기게 된다.

상대팀은 항상 최선의 방법으로 N장의 우리 팀의 '팀 숫자카드' 중 K장을 견제한다고 했을 때, 우리 팀이 얻을 수 있는 최대 점수를 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 N, K(0 ≤ K < N ≤ 100)가 주어지고 둘째 줄에는 N개의 '공유 숫자카드'에 적혀 있는 숫자, 셋째 줄에는 N개의 '팀 숫자카드'에 적혀 있는 숫자가 주어진다. 이 수는 -10,000보다 크거나 같고, 10,000보다 작은 정수이다.

출력

우리 팀이 얻을 수 있는 최대 점수를 출력한다.

예제 입력	예제 출력
5 2 -1 2 3 4 5 -1 0 2 3 4	10

힌트

상대팀은 '팀 숫자카드' 2장의 카드를 견제 할 수 있고 가장 큰 점수가 나올 수 있는 카드 4와 카드 3을 견제할 것이다. 이후 남은 카드로 이 팀은 '공유 숫자카드'의 카드 5와 '팀 숫자카드'의 카드 2로 최대 10점을 얻을 수 있다.

Problem E
방탈출
시간 제한 : 1 Second

방탈출 게임을 하던 혜민이는 마지막 문제에 봉착했다. 단서는 다음과 같다.

1. 앞에는 일렬로 놓여진 N개의 버튼이 모두 불이 꺼진 상태로 있다.
2. 0 또는 1로 구성되어 있는 N자리 숫자가 적힌 쪽지가 있다.
3. 0은 불이 꺼진 버튼, 1은 불이 켜진 버튼을 뜻한다.
4. 불이 켜져 있는 버튼을 누르면 불이 꺼지고, 불이 꺼져 있는 버튼을 누르면 불이 켜진다.
5. 버튼을 누르면 그 버튼 뿐만이 아닌 오른쪽 두 개의 버튼도 같이 눌린다.

혜민이는 현재 모두 불이 꺼진 상태에서 버튼을 최소로 눌러서 쪽지와 똑같은 상태로 만들어야 한다는 것을 알아냈다! 혜민이를 도와줘서 방탈출 게임에 성공하자.

입력

첫째 줄에 N(1 ≤ N ≤ 1,000,000)가 주어지고 둘째 줄에는 쪽지에 적혀 있는 N자리의 숫자가 빈 칸을 사이에 두고 주어진다.

출력

눌러야하는 버튼의 최소값을 출력한다.

예제 입력	예제 출력
7 0 0 1 0 0 1 0	2

힌트

다음과 같이 2개의 버튼을 눌러 쪽지에 적혀 있는 상태를 만들 수 있다.

모든 버튼이 꺼진 처음 상태 → 0 0 0 0 0 0 0
세 번째 버튼을 누른 상태 → 0 0 1 1 1 0 0
네 번째 버튼을 누른 상태 → 0 0 1 0 0 1 0

Problem F
수영장 사장님
시간 제한 : 2 Second

수영장 사업을 시작하려는 수형이는 산의 자연을 훼손하지 않고 지형을 그대로 이용한 수영장을 만들기로 한다. 그래서 물이 고일 수 있는 부분에만 물을 채워넣는 방법을 사용하기로 한다. 이 때 수형이는 여기서 얼마만큼의 물을 채울 수 있는지 궁금해 하는데, 땅의 정보가 주어졌을 때 얼마만큼의 물을 채울 수 있는지 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 N, M(1 ≤ N, M ≤ 100)가 주어진다. 다음 N 줄동안 매 줄마다 M개의 H(1 ≤ H ≤ 10,000)가 주어진다. 여기서 i 번째 줄의 j 번째 숫자를 H[i][j] 라고 할 때, H[i][j]는 해당하는 땅의 높이이다.

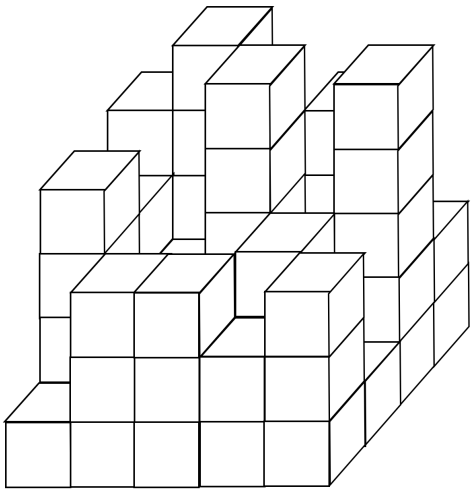
출력

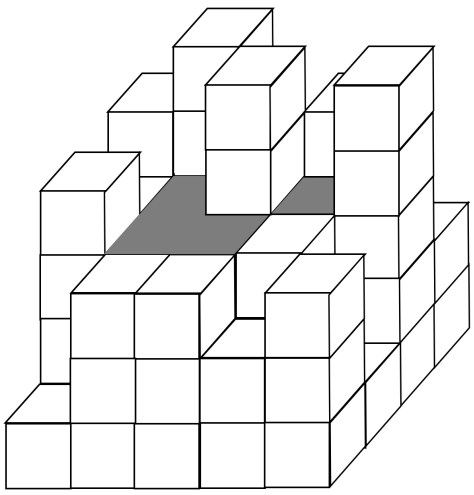
최대한 물을 채웠을 때 얼마만큼의 물을 채울 수 있는지를 출력한다.

예제 입력	예제 출력
4 5 4 5 3 4 2 3 2 5 2 5 4 1 2 3 1 1 3 3 2 3	5

힌트

다음과 같이 총 5 만큼의 물을 채울 수 있다.





Problem G
Python 문법
시간 제한 : 1 Second

프로그래밍 언어 Python에는 C나 Java와는 다르게 중괄호를 여닫아서 코드를 구분하지 않는다. 대신에, indendation(들여쓰기 - Tab)으로 구분한다. 이 문제에서는 Python의 구문을 두 가지만 다룰 것이고, 그것들은 다음과 같다.

1) **For statement** : 반복문에 쓰이는 구문으로써 여러 구문들을 포함할 수 있다. For 문이 시작되면 반복시킬 구문들은 For 문보다 한 번 더 탭으로 띄워야 하고, For 문 안에 블록은 비어있을 수 없다.
2) **Execute statement** : 실행문이다. 한 줄에 하나씩 쓸 수 있다.

여러 구문들이 indentation 없이 주어졌을 때, indentation을 만족시키는 경우의 수가 총 몇개인지 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 N(1 ≤ N ≤ 5000)개의 문자열이 주어진다. 각 문자는 'f'(본문의 For statement) 또는 'e'(본문의 Execute statement)로 이루어져 있으며, 만들 수 있는 모든 경우가 Python 문법에 어긋나는 입력은 주어지지 않는다.

출력

구문들이 주어졌을 때, Python indentation을 만족시키는 모든 경우의 수를 1,000,000,007로 나눈 나머지를 출력한다.

예제 입력	예제 출력
fefe	2

힌트

예제 입력과 출력은 다음의 경우가 존재한다.

```
1)
for statement
     execute statement
for statement
     execute statement
```

```
2)
for statement
     execute statement
     for statement
          execute statement
```

Problem H
도토리 숨기기
시간 제한 : 1 Second

HEPC 1등 상금으로 도토리 D개를 받은 욕심많은 다람쥐 수형이는 자신의 모든 도토리를 뺏기지 않게 보관하려고 한다. 수형이는 1~N까지의 번호가 붙여있는 N개의 상자를 가지고 있고 이 안에 도토리를 넣어 다른 다람쥐들이 찾지 못하게 전부 숨기려고 한다. 방이 너무 많아 도토리가 있는 상자를 모두 외울 수 없는 수형이는 A번 상자부터 B번 상자 전까지 C개 간격으로 도토리를 하나씩 더 넣는 규칙을 만들었다! 다른 다람쥐들이 규칙을 눈치채고 모든 도토리를 잃는 것이 무서운 나머지 이러한 규칙들을 K개를 만들어 도토리를 최대한 안전하게 저장해 놓으려고 한다. 예를 들어 100번 상자부터 150번상자까지 10개 간격으로, 110번 상자부터 150번 상자까지 15개 간격으로 넣는다면 100, 110, 120, 125, 130, 140, 150번 상자에 도토리가 있으며 110번 상자와 140번 상자에는 2개의 도토리가 들어가 있게 된다. 상자 하나에 들어갈 수 있는 도토리의 개수는 제한이 없으며 앞의 상자부터 최대한 꽉꽉 채워 나간다고 했을 때 마지막 도토리가 들어가 있는 상자의 번호를 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 상자의 개수 N(1 ≤ N ≤ 1,000,000)과 규칙의 개수 K(1 ≤ K ≤ 10,000), 도토리의 개수 D(1 ≤ D ≤ 1,000,000,000)가 주어진다. 그 후 K개 줄에는 A, B, C(1 ≤ C ≤ A ≤ B ≤ N)가 주어지며 A번 상자부터 B번 상자까지 C개 간격으로 도토리를 하나씩 넣는 규칙을 뜻한다. D는 모든 규칙으로 넣을 수 있는 도토리의 수보다 같거나 작다.

출력

D개의 도토리를 규칙에 맞게 상자 앞에서부터 넣었을 때 마지막 도토리가 들어가는 상자의 번호를 출력하시오.

예제 입력	예제 출력
200 2 5 100 150 10 110 150 15	125