

평생 새내기가 되고 싶은 유신이 (first-year)

시간 제한: 1 초
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 생활을하기로 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 줄에 유신이의 학번과 학년이 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

피카츄 라이츄 (pika)

시간 제한: 1 초
메모리 제한: 512 MB

포켓몬스터의 주인공은 지우와 피카츄이다. 피카츄는 N 번 대회를 치를 예정인데, 대회를 치르면 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 1 6 2 3 3 5	Raichu
1 10 2 3 4	Pikachu

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.

함정에 빠진 이동관 (trap)

시간 제한: 2 초
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. 동관이의 위치는 (a, b) 는 항상 $1 \leq a \leq N, 1 \leq b \leq M$ 를 만족해야 하며, 이 조건을 만족하는 위치로만 이동할 수 있다.
- 현재 위치의 트램펄린에 x 가 적혀 있다면, 상하좌우로 x 칸 떨어진 곳으로만 이동할 수 있다. 즉, 현재 위치가 (a, b) 라면, $(a+x, b)$, $(a-x, b)$, $(a, b+x)$, $(a, b-x)$ 이며 위의 규칙을 만족할 때 이동할 수 있다.
- 만약 어느 칸으로도 이동할 수 없으면 영원히 그 위치에 있게 된다.

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 트램펄린에 적혀 있는 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, 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.

대자보 (poster)

시간 제한: 2 초
메모리 제한: 512 MB

고려대학교에는 학생들의 소통을 위한 게시판이 설치되어있다. 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 알고 있었다. 대자보의 수열은 알파벳 소문자로 이루어진 단어를 아래 방법을 따라 변환시킨 것이다.

- 단어의 각 알파벳 소문자를 알파벳 순으로 1에서 26으로 변환한다. a는 1로, b는 2로, c는 3으로, ..., y는 25로, z은 26으로 변환된다.
- 이웃한 두 숫자를 곱한 결과를 한 글자씩 나열한다. 예를 들어, apple의 각 알파벳은 1 16 16 12 5으로 변환된다. 이웃한 두 숫자를 곱하면 16 256 192 60이 되며, 한 자씩 나열하면 1 6 2 5 6 1 9 2 6 0이 된다.

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

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 굴렀고, 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 헌금 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 헌금 금액을 신앙심의 값으로 바꾼다. ($m_i := f_i$)

사제가 빨간 약과 파란 약을 신도들에게 적절히 먹였을 때 얻을 수 있는 헌금 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 헌금 금액을 의미한다.

출력 형식

첫 번째 줄에 사제가 빨간 약과 파란 약을 신도들에게 적절히 먹였을 때 얻을 수 있는 헌금 총합의 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

○○ (○○)

시간 제한: 1 초
메모리 제한: 512 MB

영수의 이름에 ○이 2개 들어가있다는 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 점의 수 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 읽어들이 답을 판정하기 때문에 16 자리부터는 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
7 0.0000000000000000 0.0000000000000000 0.0000000000000000 2.0000000000000000 2.0000000000000000 2.0000000000000000 2.0000000000000000 0.0000000000000000 50.0000000000000000 11.0000000000000000 51.0000000000000000 10.0000000000000000 49.0000000000000000 10.0000000000000000	1.0 1.0 1.4142135624 50.0 10.0 1.0
3 -4.0000000000000000 0.0000000000000000 0.0000000000000000 1.012345678901234 5.0000000000000000 0.0000000000000000	0.0 0.0 1.012345678901234 0.5 0.0 4.50000

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 $-9 \times 10^5 \leq x, y \leq 9 \times 10^5, 0.51 \leq r \leq 9 \times 10^6$ 조건을 만족하는 답이 존재함이 보장된다.

임의의 세 점이 이루는 각이 0.95π 이하임이 보장된다.

Sum without carries (nocarry)

시간 제한: 3 초
메모리 제한: 1024 MB

정수 x, y 에 대해 $x @ y$ 라는 연산을 정의하자. $x @ y$ 는 x 와 y 를 d 진법 상에서 더하되, carry(자리 올림)가 없는 연산이다. 즉, 각 자리수를 따로 더하고 각 자리마다 d 로 나눈 나머지를 취하는 연산이다.

예를 들어, $d = 4$ 이면, $7_{10} @ 10_{10} = 13_4 @ 22_4 = 31_4 = 13_{10}$ 이 된다.

당신에게 q 개의 쿼리가 주어진다. 각 쿼리는 2개의 정수 l, R 을 담고 있다. 당신은 각 쿼리마다 다음을 만족하는 r 중에서 가장 큰 r 값을 구해야 한다.

- $l < r \leq R$
- $a_l @ a_{l+1} @ \dots @ a_r$ 이 가능한 한 제일 높아야 한다.

입력 형식

첫 번째 줄에 세 정수 n, d, q 가 공백으로 구분되어 주어진다. ($2 \leq n \leq 10^5, 2 \leq d \leq 10, 1 \leq q \leq 10^5$) n 은 수열의 길이, d 는 진법, q 는 쿼리의 개수이다.

두 번째 줄에 수열의 값을 의미하는 n 개의 정수 a_1, a_2, \dots, a_n 이 공백으로 구분되어 주어진다. ($0 \leq a_i \leq 10^{18}$)

세 번째 줄부터 q 개의 줄에 걸쳐 쿼리를 의미하는 2개의 정수 l, R 이 공백으로 구분되어 주어진다. ($1 \leq l < R \leq n$)

출력 형식

q 개의 줄에 걸쳐 각 쿼리에 대한 정답 r 을 출력한다.

예제

표준 입력	표준 출력
7 10 6	5
16 27 90 0 73 55 4	4
1 6	5
2 4	6
4 6	4
5 6	2
3 5	
1 3	

서버 증축 (server)

시간 제한: 1 초
메모리 제한: 512 MB

“지난 밤, 선량한 서버가 죽었습니다. 관리자들은 모두 고개를 들어주세요.”

서버 관리자에게 서버가 다운되는 것만큼 무섭고 고된 일은 없을 것이다. 상현이도 그 마수에서 벗어날 수 없었다. 상현이는 자신이 도맡아 관리하는 서버들이 뻥는 걸 원하지 않기에 서버 용량을 늘리는 방향을 선택했다.

상현이는 회사로부터 $2^0, 2^1, \dots, 2^{k-1}$ MB의 용량을 지니고 있는 서로 다른 디스크를 각 용량별로 a 개씩 지원받았다. 딥러닝을 열심히 돌려본 결과 서버에 정확히 n MB의 용량을 추가로 늘리면 최적의 성능을 발휘할 것으로 분석되었다. 문제는, n MB의 용량을 만드는 방법이 너무 다양했다는 것이다! 상현이는 서버 증축을 할 수 있는 방법의 수를 구해보고자 한다. 그 수가 너무 커질 걸 우려해, 1048573으로 나눈 나머지를 계산하려고 한다. $1048573 = 2^{20} - 3$ 은 소수이다.

입력 형식

첫 번째 줄에는 세 정수 k, a, n 이 공백으로 구분되어 주어진다. ($1 \leq k \leq 40, 1 \leq a \leq 10^6, 1 \leq n \leq 10^{15}$)

k 는 제공되는 디스크의 용량의 가짓수, a 는 각 용량별 서로 다른 디스크의 개수, n 은 증축해야 하는 용량을 의미한다.

출력 형식

첫 번째 줄에 정확히 n MB를 증축할 수 있도록 디스크를 선택하는 경우의 수를 1048573으로 나눈 나머지를 출력한다.

예제

표준 입력	표준 출력
2 3 5	12
5 10 1000	0
11 23 58	182185

참고 사항

이 문제를 풀 때 뫼비우스의 정리(Lucas' Theorem)를 사용하면 편리할 것이다.

뫼비우스의 정리란, 양의 정수 m, n 이 소수 p 에 대해 p 진법으로 각각 $m_k m_{k-1} \dots m_0, n_k n_{k-1} \dots n_0$ 으로 표현될 때,

$$\binom{m}{n} \equiv \binom{m_k}{n_k} \times \binom{m_{k-1}}{n_{k-1}} \times \dots \times \binom{m_1}{n_1} \times \binom{m_0}{n_0} \pmod{p}$$

이 성립한다는 것이다. $\binom{n}{r} = \frac{n!}{r!(n-r)!}$ 이며, $n < r$ 일 때 $\binom{n}{r} = 0$ 으로 정의한다.

예를 들어 $\binom{7}{4}$ 를 5로 나눈 나머지를 뫼비우스의 정리를 이용해 구해보자. $7 = 12_5$ 이고 $4 = 04_5$ 이기에 $\binom{7}{4} \equiv \binom{1}{0} \cdot \binom{2}{4} \equiv 1 \cdot 0 \equiv 0 \pmod{5}$ 이다. 실제로 $\binom{7}{4} = 35$ 로, 5로 나눈 나머지는 0이다.

다른 예시로 $\binom{111}{11}$ 를 7로 나눈 나머지는, $111 = 216_7$ 이고 $11 = 014_7$ 이므로 $\binom{111}{11} \equiv \binom{2}{0} \cdot \binom{1}{1} \cdot \binom{6}{4} \equiv 1 \cdot 1 \cdot 1 \equiv 1 \pmod{7}$ 이다. 실제로 $\binom{111}{11} = 473239787751081 = 7 \times 67605683964440 + 1$ 이다.

기묘한 여행계획 2 (trip)

시간 제한: 1 초
메모리 제한: 512 MB

기묘한 세계의 이상한 도시에 사는 엘리스는 다른 도시로 여행을 떠날 계획을 세웠다. 기묘한 세계에는 총 N 개의 도시가 있으며, M 개의 양방향 도로로 연결되어 있다. 각 도시는 순서대로 1번부터 N 번까지 번호가 붙어 있으며 엘리스가 사는 도시의 번호는 1번이다. 각 도로는 이용하는데 필요한 비용 v 가 있고, 도로를 이용하기 위해서는 돈을 내야 한다. 기묘한 세계에서 도로 이용 비용을 부과하는 방법은 다음과 같다.

목적지에 도달할 때 까지 이용한 도로의 비용을 순서대로 v_1, v_2, \dots, v_k 라고 했을 때 $v_1 \oplus v_2 \oplus \dots \oplus v_k$ 가 부과된다. (\oplus 는 bitwise xor이다.)

엘리스는 가난하기 때문에 여행을 떠나기 전에 가장 효율적인 여행 방법을 구하려고 한다. 계산에 서툰 엘리스를 대신하여 엘리스가 사는 도시를 제외한 모든 도시까지 가는 최소 비용을 구하는 프로그램을 작성하시오.

엘리스가 사는 도시의 번호는 1번이다. 원하는 목적지에 도달할 때까지 이용하는 도로의 개수에 제한은 없으며, 1번 도시에서 출발해 도로들을 이용하여 다른 모든 도시에 도달할 수 있음이 보장된다.

입력 형식

첫 번째 줄에 도시의 수 N 과 도로의 수 M 이 공백으로 구분되어 주어진다.

($2 \leq N \leq 2 \times 10^5$, $N - 1 \leq M \leq 2 \times 10^5$)

두 번째 줄부터 M 개의 줄에 걸쳐 도로의 정보를 의미하는 세 정수 a, b, c 가 공백으로 구분되어 주어진다. ($1 \leq a, b \leq N$, $a \neq b$, $0 \leq c \leq 10^9$) 이는 도시 a 와 도시 b 사이에 c 의 비용을 가지는 도로가 존재한다는 뜻이다.

두 도시를 직접적으로 연결하는 도로는 최대 1개 있다.

출력 형식

첫 번째 줄에 $N - 1$ 개의 수 C_2, C_3, \dots, C_N 를 공백으로 구분하여 출력한다. C_i 는 1번 도시에서 i 번 도시로 가는 최소 비용이다.

예제

표준 입력	표준 출력
3 3 1 2 1 2 3 5 1 3 6	1 4
4 5 1 2 5 2 3 7 1 3 1 4 1 0 3 4 4	0 1 0