

2021 신촌지역  
대학생 프로그래밍 대회 동아리  
연합 겨울 대회

# SUAPC 2021 winter

## Official Problemset



ICPC Sinchon



Sogang ICPC Team



Algos



Morgorithm



EDOC



HI-ARC

Sponsors

kakao

NAVER D<sup>2</sup>

HYUNDAI  
MnSOFT

Tangunsoft

STARTLINK

SOLVED. AC

Youngjin.com  
Y.

한빛미디어  
hanbitmedia.com

Jpub  
Jpub.com



## 문제 목록

문제지에 있는 문제가 총 13문제가 맞는지 확인하시기 바랍니다.

- A 우선순위 계산기
- B 떡국
- C 반짝반짝
- D 달고나
- E 시철이가 사랑한 수식
- F 성실당
- G 신촌지역 초중고등학생 프로그래밍 대회 동아리 연합 대회
- H 카카오톡
- I 팰린드롬 척화비
- J 의자 게임
- K 합성인수분해
- L 습격받은 도시
- M Go와 함께하는 전화망 서비스

모든 문제의 메모리 제한은 1GB로 동일합니다.



## 문제 A. 우선순위 계산기

시간 제한 1.5 초  
메모리 제한 1024 MB

국렬이는 두 번씩이나 계산기 문제를 내놓고 또 계산기 문제를 냈다. 이대로라면 죽을 때까지 계산기를 우려먹을 생각이요, 당신은 귀찮지만 상금을 얻기 위해서 주어진 수식을 규칙에 맞게 계산해야 한다.

입력으로 주어지는 수식은 띄어쓰기 없이 수와 연산자가 번갈아 가면서 나온다. 수식의  $i$ 번째 수를  $X_i$ ,  $i$ 번째 연산자를  $Op_i$ 로 표시하면 수가  $n$ 개인 식을  $X_1 Op_1 X_2 Op_2 \dots Op_{n-1} X_n$ 로 표기할 수 있다. 연산자의 종류는  $+$ ,  $-$ ,  $*$ ,  $/$ 가 있다. 마지막에 연산자가 있는 경우와  $X_i$ 가 음수인 경우는 입력으로 주어지지 않는다. 즉,  $-1 - 1$ ,  $2 + -3$  같은 경우는 입력으로 주어지지 않는다. 그리고 불필요한 0이 수의 앞에 있을 수 있다. 즉,  $001 + 0002$  같은 수식이 입력으로 주어질 수 있다.

주어진 수식을 다음 규칙에 맞게 계산할 것이다.

1.  $X_i Op_i X_{i+1}$  중 가장 큰 값을 갖는  $i$ 를 선택한다. ( $1 \leq i \leq n-1$ )
2.  $X_i Op_i X_{i+1}$ 가 가장 큰 값을 갖는  $i$ 가 2개 이상인 경우,  $Op_i$ 가 곱하기 또는 나누기인  $i$ 를 선택한다.
3.  $X_i Op_i X_{i+1}$ 가 가장 큰 값을 갖고  $Op_i$ 가 곱하기 또는 나누기인  $i$ 가 2개 이상인 경우,  $i$ 가 가장 작은 것을 선택한다.
4.  $X_i Op_i X_{i+1}$ 를 먼저 계산하고, 위의 과정을 연산자의 개수만큼 반복한다.

예를 들어서 수식이  $3 \times 2 + 5 - 5 + 7$ 로 주어진다고 하면 다음과 같이 계산된다.

- $5 + 7 = 12$ 가 가장 크기에 먼저 계산한다. 이후 계산식은  $3 \times 2 + 5 - 12$ 이다.
- 그다음으로  $2 + 5 = 7$ 를 계산한다. 이후 계산식은  $3 \times 7 - 12$ 이다.
- 그 후  $3 \times 7 = 21$ 를 계산한다. 이후 계산식은  $21 - 12$ 이다.
- 마지막에 남은  $21 - 12$ 를 계산하면 최종 결과 값은 9가 된다.

이 문제에서의 나눗셈은 C++에서 정수 간에 정의된 나눗셈으로 생각한다. 즉, 나누어지는 수가 양수면 나머지가 0 이상, 음수면 나머지가 0 이하로 처리가 되는 식으로 진행했을 때 나오는 몫을 계산하는 방식으로 이루어진다. 예를 들어,  $3/2 = 1$ ,  $(-3)/2 = -1$ ,  $3/(-2) = -1$ ,  $(-3)/(-2) = 1$ 로 계산된다.

이와 같은 계산 과정에 따라 주어진 식을 계산하시오.

### 입력

다음과 같이 입력이 주어진다.

S

- S는 계산하고자 하는 수식으로 지문에서 언급된 수와 연산자가 다른 문자 없이 교대로 나오며, 길이는 200000 이하이다.



- 계산 과정 중의 모든 수는  $-2^{63}$  이상  $2^{63}$  미만이며, 0으로 나누는 경우는 없다. 수 앞에 불필요한 0이 있을 수 있다.
- 주어지는 연산자는  $+$ ,  $-$ ,  $*$ ,  $/$ 다.

## 출력

첫 번째 줄에 주어진 식을 계산한 결과 값을 출력한다. 불필요한 0은 제거해야 한다.

## 입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3*2+5-5+7	9
8/4/2	1
000009223372036854775807	9223372036854775807



## 문제 B. 떡국

시간 제한 1 초  
메모리 제한 1024 MB



FOR DEVELOPERS, BY DEVELOPERS

Naver D2를 아시나요? D2는 For Developers, By Developers의 약자로, 개발자들을 위해 개발자들이 직접 만들어 가고 있는 네이버 개발자 지원 프로그램입니다. 네이버가 축적한 기술과 지식을 공유하고, 외부 개발자들을 지원해 대한민국 개발자 역량 강화를 이끌고, 이를 통해 업계 전체와 네이버가 함께 성장하는 선순환 구조를 만들고자 합니다.

사실 네이버의 개발자 지원은 오랜 기간 꾸준히 이어져 왔습니다. 개발자 컨퍼런스 DEVIEW를 비롯, 오픈 소스와 개발 도구 공개, 학회 및 커뮤니티 지원 등 여러 지원 프로그램이 있었습니다. 이런 다양한 프로그램을 하나로 통합한 것이 바로 Naver D2입니다.

함께 성장하는 개발자 지원 프로그램인 NAVER D2에서는 매년 개발자 컨퍼런스 DEVIEW를 개최한다.

2021년 DEVIEW에도 다양한 주제를 선보이기 위한 발표 준비 작업이 한창이다. 그런데 아주 큰 문제가 생겼다. 책상 위에 다 먹고 남은 떡국 그릇이 너무 많이 쌓여 작업을 진행할 수가 없다. 우연히 옆을 지나가던 당신이 이를 도와주기로 했다!

떡국 그릇 위에는 크기가 더 작은 떡국 그릇 하나를 쌓을 수 있다. 쌓은 떡국 그릇 위에 같은 방법으로 떡국 그릇을 또 쌓을 수 있다. 예를 들어, 크기가 4, 2, 3, 1인 떡국 그릇에 대해 4-3-2-1 순서로 쌓을 수 있지만 3-4-2-1 순서로는 쌓을 수 없다. 이렇게 쌓은 한 개 이상의 떡국 그릇들을 **떡국 그릇 탑**이라고 하자.

당신은 **떡국 그릇 탑**의 개수를 최소로 만들어 책상 위의 공간을 확보하려고 한다.

다음은 4, 2, 3, 1, 2인 떡국 그릇으로 쌓을 수 있는 **떡국 그릇 탑**의 개수의 예시이고, 최소 개수는 2개이다.

- 5개 : [4, 2, 3, 1, 2]
- 4개 : [4-2, 3, 1, 2] 또는 [4-3, 2, 1, 2] 또는 [4, 3-2, 1, 2] 또는 ...
- 3개 : [4-2, 3-1, 2] 또는 [4-1, 3-2, 2] 또는 [4-3, 2-1, 2] 또는 ...
- 2개 : [4-2, 3-2-1] 또는 [4-2-1, 3-2] 또는 [4-3-2, 2-1] 또는 ...
- 1개의 떡국 그릇 탑으로 만들 수 없다.

떡국 그릇들의 크기가 주어졌을 때, **떡국 그릇 탑**의 최소 개수를 구해주자. 당신에게 감사의 표시로 NAVER D2에서 후원하는 SUAPC 2021w의 한 문제를 정답 처리해줄 것이다.

### 입력

다음과 같이 입력이 주어진다.



$N$

$c_1 c_2 \dots c_N$

- $N$ 은 떡국 그릇의 개수이다. ( $1 \leq N \leq 500000$ )
- $c_i$ 는  $i$ 번째 떡국 그릇의 크기이다. ( $1 \leq c_i \leq 50000$ )
- 입력으로 주어지는 모든 수는 정수다.

## 출력

떡국 그릇 탑의 최소 개수를 출력한다.

## 입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 4 2 3 1 2	2



## 문제 C. 반짝반짝

시간 제한 2 초  
메모리 제한 1024 MB

이제 조금밖에 남지 않은 겨울 기분을 만끽하고 싶은 수현이는 지금부터라도 크리스마스 트리를 장식하려고 한다. 크리스마스 트리는 전구 스트립으로 두른다. 전구 스트립에는 전구  $N$ 개가 일(一)자로 설치되어 있고, 왼쪽에 전원을 넣는다. 특이하게도 이 전구 스트립은 전구 하나가 고장 나면 고장 난 전구를 시작으로 오른쪽에 설치되어 있는 모든 전구에 불이 들어오지 않게 된다.

수현이는 반짝반짝한 걸 좋아한다. 따라서 전구 스트립을 최대  $K$ 개의 토막으로 자르고, 왼쪽에 전원을 각각 다시 넣어서 트리를 장식할 것이다. 이렇게 해서 불이 들어온 전구의 개수의 기댓값이 최대가 되게 하고 싶다. 각각의 전구가 고장 날 확률이 주어질 때 불이 들어온 전구의 개수의 기댓값의 최댓값을 계산하라.

### 입력

다음과 같이 입력이 주어진다.

```
N K
p1 p2 ... pN
```

- $N$ 은 전구 스트립의 길이이다. ( $1 \leq N \leq 2500$ )
- $1 \leq K \leq \min\{N, 10\}$
- $p_i$ 는 전구가 고장 날 확률이며, 소수점 아래 두 자리까지 주어진다. 왼쪽 전구에서 오른쪽 전구의 순서로 주어진다. ( $0 \leq p_i \leq 1$ )
- $N$ 과  $K$ 는 정수이다.

### 출력

불이 들어온 전구의 개수의 기댓값의 최댓값을 출력한다.

출력한 값과 정답과의 절대 오차 또는 상대 오차가  $10^{-6}$  이하여야 한다.

### 입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 1 0.50 0.50 0.50 0.50 0.50	0.96875
5 2 0.50 0.50 0.50 0.50 0.50	1.625
5 2 0.10 0.20 0.30 0.40 0.50	3.024

2021 신춘지역  
대학생 프로그래밍 대회 동아리  
연합 겨울 대회

**SUAPC 2021** winter



Sogang ICPC Team



Algos



Morgorithm



EDOC



HI-ARC

이 페이지는 공백입니다



## 문제 D. 달고나

시간 제한 1 초  
메모리 제한 1024 MB

서강대학교 K관 512호 랩실에 달고나 기계가 생겼다. 달고나 기계 옆에는 뽑기를 위해 원 또는 단순 다각형 모양의 모형 틀이 준비되어 있다. 일반적으로 모형 틀의 모양에 따라 뽑기의 난이도가 결정된다. 하지만 그 어떤 모양도 뽑기 장인 상원이를 만족시킬 수는 없다. 뽑기가 너무 쉬워 고민이던 상원이는 더 어려운 뽑기를 만드는 방법을 떠올렸다. 여러 가지 모형틀을 한 뽑기 안에 찍는다면 뽑아야 할 조각의 수가 많아져 더 어려워질 것이다!

2차원 평면 상에 원 또는 단순 다각형 모양의 도형들이 주어졌을 때, 평면을 몇 개의 영역으로 나누는지 구하는 프로그램을 작성하자.

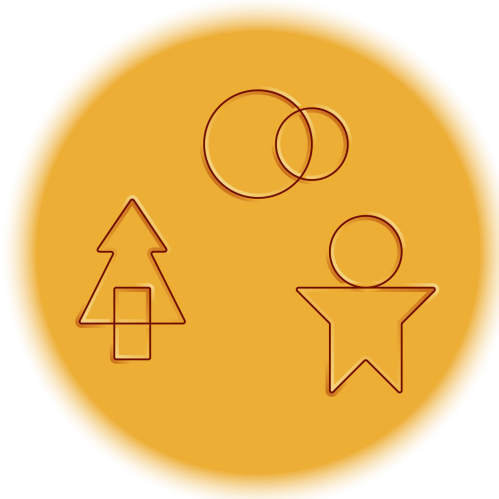


그림 1: 첫 번째 예제에 따른 뽑기 모양이다.

### 입력

다음과 같이 입력이 주어진다.

```
N  
poly1  
poly2  
...  
polyN
```

- $N$ 은 도형의 개수이다. ( $1 \leq N \leq 2000$ )
- $poly_i$ 는 각 도형의 정보이다.
  - 원인 경우, 다음과 같이  $poly_i$ 가 주어진다.



1  $x y r$

\* 중심이  $(x, y)$ 이고 반지름이  $r$ 인 원이다.  $(-10000 \leq x, y \leq 10000, 1 \leq r \leq 10000)$

- 단순 다각형인 경우, 다음과 같이  $poly_i$ 가 주어진다.

$m x_1 y_1 x_2 y_2 \dots x_m y_m$

\*  $m$ 은 단순 다각형을 이루는 꼭짓점의 개수이다.

\*  $(x_i, y_i)$ 는 각 꼭짓점의 좌표이다.  $(-10000 \leq x_i, y_i \leq 10000)$

\* 각 꼭짓점은 반시계방향 순서로 주어진다.

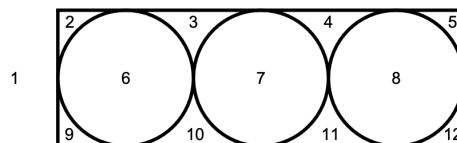
- 모든 단순 다각형의 선분 개수와 원의 개수의 합은 2000을 넘지 않는다.
- 세 개 이상의 원 또는 선분이 한 점을 지나는 경우는 주어지지 않는다.
- 입력으로 주어지는 모든 수는 정수다.

## 출력

도형들로 인해 나뉘지는 영역의 수를 출력한다.

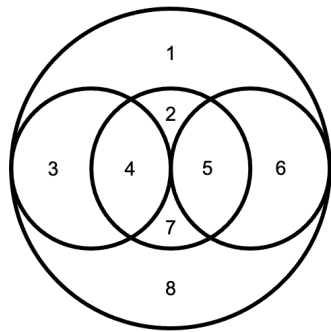
## 입출력 예시

표준 입력(stdin)	표준 출력(stdout)
<pre>6 7 3 5 1 2 2 2 0 -2 6 -2 4 2 5 2 4 2 0 2 -4 4 -4 4 0 1 16 2 2 7 20 0 12 0 14 -2 14 -6 16 -4 18 -6 18 -2 1 13 8 2 1 10 8 3</pre>	9
표준 입력(stdin)	표준 출력(stdout)
<pre>4 4 750 250 -750 250 -750 -250 750 -250 1 0 0 250 1 500 0 250 1 -500 0 250</pre>	12

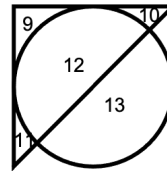




표준 입력(stdin)	표준 출력(stdout)
<pre>6 1 1 1 1 1 1 1 2 1 0 1 1 1 2 1 1 1 6 1 1 3 7 2 5 2 5 0</pre>	<pre>14</pre>



14



2021 신춘지역  
대학생 프로그래밍 대회 동아리  
연합 겨울 대회

**SUAPC 2021** winter



Sogang ICPC Team



Algos



Morgorithm



EDOC



HI-ARC

이 페이지는 공백입니다



## 문제 E. 시철이가 사랑한 수식

시간 제한 1 초  
메모리 제한 1024 MB

**Q1.** 코끼리를 냉장고에 넣는 방법은?

**A1.** 대학원생을 시킨다.

**Q2.** 수식의 값을 알아내는 방법은?

**A2.** 대학원생을 시킨다.

연희동 최고의 대학원생 시철이는 마치 한강이 흐르듯 수식을 전개하고, 그 끝마침이 대양과도 같았다.  
화려한 수식 전개로 전 세계인의 마음을 흠치던 시철이는, 어느덧 수식을 사랑하는 교수님의 눈에 띄게 되었다.

$$\sum_{n=1}^N \sum_{i=1}^n \sum_{j=1}^n \gcd(i, j) \times \text{lcm}(i, j)$$

$$\sum_{n=1}^N \sum_{i=1}^n \sum_{j=1}^n \gcd(i, j) \times \sum_{n=1}^N \sum_{i=1}^n \sum_{j=1}^n \text{lcm}(i, j)$$

교수님은 두 식의 값을 알고 싶었지만 쉽지 않았다. 그래서 교수님은 시철이에게 두 식의 값을 구해 달라고 부탁했다.  
처음에 시철이는 마지못해 수식을 전개하기 시작했지만, 결국 교수님 때문에 두 식을 사랑하게 되었다.  
하지만 시철이는 졸업논문 준비로 너무 바빴고, 본인의 대학원 동기 민수에게 나머지를 떠넘겼다. 민수는 두 식의 값이 클 수 있다고 생각해, 값을  $K$ 로 나눈 나머지를 교수님께 알려주려고 한다.  
민수와 함께 두 식의 값을 구해보자.

### 입력

다음과 같이 입력이 주어진다.

$N K$

- $1 \leq N \leq 10^6$
- $10^6 \leq K \leq 10^9 + 7$ ,  $K$ 는 소수
- 입력으로 주어지는 모든 수는 정수다.

### 출력

첫째 줄에 첫 번째 식의 값을 출력한다.  
둘째 줄에 두 번째 식의 값을 출력한다.  
값이 클 수 있으니  $K$ 로 나눈 나머지를 출력한다.



## 입출력 예시

표준 입력(stdin)	표준 출력(stdout)
1 1000000007	1 1
3 1000000007	46 648

## 노트

교수님의 눈에 띈 민수는 박사 과정까지 함께했다.



## 문제 F. 성실통

시간 제한 3 초  
메모리 제한 1024 MB

최근 수현이는 빵가게 <성실통>을 차렸다. 같은 과자보다 바삭하고 속은 포근할 정도로 촉촉한 빵을 매일 구워 파는 성실통은 얼마 지나지 않아 인스타 빵지순례 명소가 될 정도로 인기 베이커리가 되었지만, 안타깝게도 강화된 거리두기로 손님이 매우 줄었다.

성실통은 시대의 흐름에 따라 배달 주문을 받기 시작했다. 빵 하나에는  $N$ 가지의 재료가 들어가는데, 각각의 재료는 두 가지 바リエ이션이 있어서 두 가지 중 하나를 고를 수 있다. 예를 들자면, 반죽은 밀가루나 통밀가루 중 하나를 고를 수 있고, 시럽은 메이플 시럽과 딸기 시럽 중 하나를 고를 수 있는 식이다. 따라서  $2^N$  종류의 빵이 만들어질 수 있다. 아니나 다를까 맵소사, 배달 첫날부터  $2^N$ 개의 주문이 들어왔는데 주문으로 들어온 빵의 종류가 모두 달랐다.

아무리 빵 굽기에 숙련된 수현이라도  $2^N$ 개의 주문을 처리하는 건 무리였다. 다행히도 성실통의 직원들은 각자 재료 한 종류씩을 빠르게 준비할 수 있어서, 여러 직원이 한 재료씩 각자 동시에 준비하면 시간을 절약할 수 있다. 따라서 수현이는 다음 식이 최소가 되도록 주문들의 순서를 바꾸려고 한다.

$$\sum_{i=1}^{2^N-1} (i \text{ 번째 주문과 } i+1 \text{ 번째 주문에서 겹치는 재료의 종류의 수})$$

단, 첫 번째로 주문한 손님의 빵은 첫 번째로 만들려고 한다.  $N$ 과 첫 번째 손님의 주문이 주어질 때, 위의 식이 최소가 되도록 하려면 주문들을 어떤 순서로 처리해야 하는지 알려주자.

### 입력

다음과 같이 입력이 주어진다.

$N$

$p_1$

- $N$ 은 성실통 빵에 들어가는 재료의 수다. ( $1 \leq N \leq 20$ )
- $p_1$ 은 첫 번째 주문을 의미하는 문자열이다.  $i$ 번째 문자는  $i$ 번째 재료 종류를 의미하며, 배달 주문 시스템의 대역폭을 아끼기 위해 한 바リエ이션을 0으로, 다른 바リエ이션을 1로 표현한다. ( $|c_0| = N$ )

### 출력

주문  $2^N$  개를  $p_1$ 으로 시작해 한 줄에 하나씩 출력한다. 각 주문은 입력 형식과 같은 방식으로,  $i$ 번째 문자가  $i$ 번째 재료의 종류가 되도록 길이  $N$ 의 문자열을 구성해 출력한다.

조건을 만족하는 주문 순서가 여러 개 있을 경우 그중 하나만 출력한다.



## 입출력 예시

표준 입력( <i>stdin</i> )	표준 출력( <i>stdout</i> )
1 0	0 1
2 00	00 11 10 01





## 문제 G. 신촌지역 초중고등학생 프로그래밍 대회 동아리 연합 대회

시간 제한 1 초  
메모리 제한 1024 MB

SUAPC의 성공에 힘입어, 연두는 “신촌지역 초중고등학생 프로그래밍 대회 동아리 연합 대회”의 개최를 기획하고 있다. 이 대회는 신촌의 초등학교/중학교/고등학교에 다니는 8세 ~ 19세의 학생이라면 누구든 참가할 수 있다.

연두는 컴퓨터 과학을 기반으로 연구한 자신만의 풍수지리설과 사주팔자를 굉장히 맹신한다. 따라서 이번 대회가 성공적으로 개최되기 위해서는, 참가자들의 나이에 따른 자리 배치가 매우 중요하다고 믿는다. 연두가 음양비트론에 의거하여 만든 데이터들에는,  $x$ 번째 자리와  $y$ 번째 자리에 앉은 참가자의 나이를 bitwise AND 또는 bitwise OR 했을 때 어떤 값이 나와야 하는지 적혀있다.

그런데 SUAPC 2021 Winter에 이어 이 대회에도 후원사로 참여한 카카오에서, 몇 개의 자리를 골라 각 자리마다 특정 나이의 참가자를 앉혀달라는 부탁을 해왔다. 후원사의 부탁을 거절하고 싶지는 않은지라, 연두는 나머지 빈자리에 적절한 나이의 참가자를 잘 배치하여 자신의 모든 데이터를 만족시킬 수 있을지 궁금해졌다. 연두는 이미 8세 이상 19세 이하의 많은 학생을 확보해놓았다. 대회 준비에 바쁜 연두를 대신하여 연두가 원하는 대로 자리를 배치해주자.

### 입력

다음과 같이 입력이 주어진다.

```
N
a1 a2 ... aN
M
t1 x1 y1 z1
...
tM xM yM zM
```

- $N$ 은 자리의 개수이다. ( $2 \leq N \leq 50000$ )
- $a_i = 0$  또는  $8 \leq a_i \leq 19$ 
  - $a_i = 0$ 이면, 현재  $i$ 번째 자리는 빈자리다.
  - $8 \leq a_i \leq 19$ 면,  $i$ 번째 자리에 앉을 참가자의 나이가  $a_i$ 다.
- $M$ 은 연두의 데이터의 개수다. ( $1 \leq M \leq 100000$ )
- $t_i = \&$  또는  $t_i = |$ 
  - $t_i = \&$ 면,  $x_i$ 번째 자리에 앉은 참가자의 나이와  $y_i$ 번째 자리에 앉은 참가자의 나이의 bitwise AND가  $z_i$ 여야 한다.
  - $t_i = |$ 면,  $x_i$ 번째 자리에 앉은 참가자의 나이와  $y_i$ 번째 자리에 앉은 참가자의 나이의 bitwise OR이  $z_i$ 여야 한다.
- $1 \leq x_i, y_i \leq N$



- $0 \leq z_i \leq 31$
- 입력으로 주어지는 모든 수는 정수다.

## 출력

빈자리에 8세 이상 19세 이하의 참가자를 배치하여 연두의 모든 데이터를 만족시킬 수 있다면, 첫 번째 줄에 1을 출력한다. 그다음 줄에, 1번째, 2번째, ...,  $N$ 번째 자리에 앉을 참가자의 나이를 공백으로 구분하여 출력한다. 그런 배치가 여러 가지면 그 중 아무거나 하나를 출력한다.

만약 그런 배치가 존재하지 않는다면, 대신에 첫 번째 줄에 0을 출력한다.

## 입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 15 0 0 0 3 & 1 2 1 & 1 3 12   3 4 31	1 15 17 12 19

첫 번째 자리에는 15세의 참가자가 이미 배치되어 있다. 두 번째, 세 번째, 네 번째 자리에 각각 17세, 12세, 19세의 참가자를 배치하면,  $15 \& 17 = 1$ ,  $15 \& 12 = 12$ ,  $12 | 19 = 31$  이므로 모든 데이터를 만족한다. 이 예제에서는 이 배치 이외에 가능한 다른 배치는 존재하지 않는다.

표준 입력(stdin)	표준 출력(stdout)
3 0 0 0 3   1 2 24   2 3 24   3 1 24	0

## 문제 H. 카카오톡

시간 제한 2 초  
메모리 제한 1024 MB



카카오톡은 주식회사 카카오가 2010년 3월 18일 서비스를 시작한 글로벌 모바일 인스턴트 메신저로, 2020년 기준 4000만 명의 사용자가 등록돼 있고 시장 점유율이 96%로 사실상 거의 모든 국민이 사용할 정도로 점유율이 매우 높다.

카카오는 이번에 Sinchon ICPC을 지원해 주는 대가로 출제진인 국렬이에게 카카오톡의 특이한 오픈톡방에 대한 실험 결과를 요구하였다. 실험에 대한 내용은 다음과 같다.

1.  $N$ 명의 유저가 모인 특이한 오픈톡방이 있다.
2. 특이한 오픈톡방은 하나의 좌표 평면으로 구성되어 있으며, 각각 유저들은 좌표 평면 상의 서로 다른 직선 1개를 할당받는다.
3. 각 유저들이 서로의 톡을 보기 위해서는 각 유저들의 직선이 서로 만나야 한다. 서로 만나지 않는다면 서로의 톡을 볼 수 없다.

이때, 국렬이는 특이한 오픈톡방 내에서 서로의 톡을 확인할 수 있는 유저의 쌍의 수를 구해야 한다. 국렬이는 너무 게을러서 이 실험을 대회에 떠넘겨버렸다. 당신은 상금을 위해 이 문제를 해결해야 한다.

### 입력

다음과 같이 입력이 주어진다.

```
N
a1 b1 c1
...
aN bN cN
```

- $N$ 은 오픈 톡방에 모인 사람의 수를 의미하는 양의 정수다. ( $1 \leq N \leq 500000$ )
- $a_i x + b_i y + c_i = 0$ 은  $i$ 번째 유저가 할당받은 직선이다. ( $1 \leq i \leq N$ )
- $-10^9 \leq a_i, b_i, c_i \leq 10^9$  ( $1 \leq i \leq N$ )
- $(a_i, b_i) \neq (0, 0)$  ( $1 \leq i \leq N$ )
- 다수의 유저들이 동일한 직선을 할당받는 경우는 존재하지 않는다.



- 입력으로 주어지는 모든 수는 정수다.

## 출력

서로의 특을 확인할 수 있는 유저의 쌍의 수를 출력하여라.

## 입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 1 2 3 2 4 5 1 3 6 3 9 7	4
3 1 2 3 -1 -2 3 1 -2 3	2



## 문제 I. 팰린드롬 척화비

시간 제한 1 초  
메모리 제한 1024 MB

흥선이는 팰린드롬을 싫어한다. 어느 날 지구를 정복한 흥선이는 팰린드롬 척화비를 세워, 전 지구의 팰린드롬을 없애버렸다. 그리고 수미상관 순수비를 만들어 수미상관을 널리 퍼뜨렸다.

팰린드롬과 수미상관의 정의는 다음과 같다.

- 팰린드롬 : (앞뒤가 똑같은 팰린드롬~) “u”, “xyx”, “krrk” 같이 뒤집어 읽어도 같은 문자열을 뜻한다.
- 수미상관 : (앞뒤가 똑같은 수미상관~) “z”, “ppqq”, “astoast” 같이 앞쪽 절반이 뒤쪽 절반과 같은 문자열을 뜻한다. 정확히는, 길이가  $N$ 인 문자열이면 길이가  $\lfloor \frac{N}{2} \rfloor$ 인 접두사와 접미사가 동일한 문자열을 뜻한다.

민수는 팰린드롬의 유구한 역사를 살리고 싶었지만, 여기저기 박힌 팰린드롬 척화비 때문에 그 꿈을 이룰 수 없었다. 그래도 하늘이 무너져도 솟아날 구멍이 있으랴! 민수는 길이가  $N$ 인 수미상관이면서 팰린드롬인 문자열을 발견한 것이다! 이것은 흥선이의 팰린드롬 척화비에 위배되지 않았고, 민수는 팰린드롬의 유구한 역사를 후대에 남기기 위해 노트 한 권에 이 사실을 적었다.

미래에 후손들은 민수의 노트를 발견했지만, 시간이 너무 오래 지나버렸다. 민수의 노트는 얼룩과 먼지 때문에 많은 부분을 알아볼 수가 없었다. 오로지 민수가 길이가  $N$ 인 수미상관이면서 팰린드롬인 문자열을 발견했다는 사실만을 알 수 있었을 뿐이다.

민수의 후손인 시철이는 민수의 노트가 사실이라 생각했고, 길이가  $N$ 인 수미상관이면서 팰린드롬인 문자열을 찾기 위해 공학관에서 밤새 연구를 하고 있다.

민수가 찾은 것은 무엇이었을까?

### 입력

다음과 같이 입력이 주어진다.

$N$

- $1 \leq N \leq 1000000$
- $N$ 은 정수다.

### 출력

길이가  $N$ 이고 알파벳 소문자로 이루어진, 수미상관이면서 팰린드롬인 문자열을 출력한다.

항상 그러한 문자열은 존재하고, 만약 여러 가지가 있다면 그중 하나를 출력한다.

### 입출력 예시

표준 입력(stdin)	표준 출력(stdout)
1	a
3	sts

2021 신촌지역  
대학생 프로그래밍 대회 동아리  
연합 겨울 대회

**SUAPC 2021** winter



Sogang ICPC Team



Algos



Morgorithm



EDOC



HI-ARC

## 노트

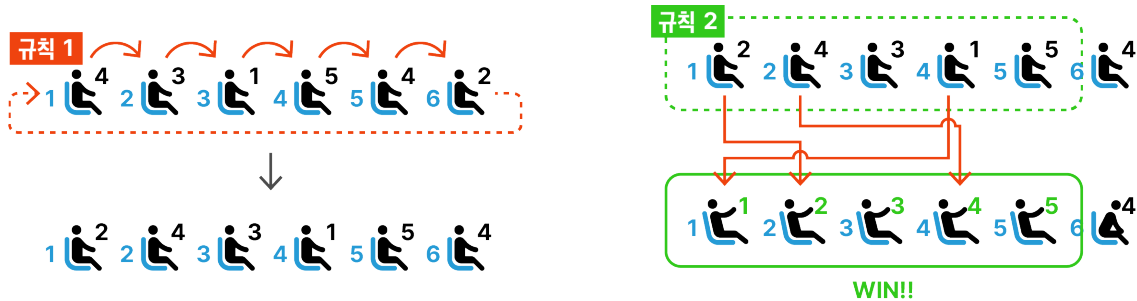
민수는 노트의 여백이 부족해, 발견했다는 사실만을 적었다고 전해진다.

## 문제 J. 의자 게임

시간 제한 3 초  
메모리 제한 1024 MB

드디어 HI-ARC의 첫 MT가 시작되었다. HI-ARC의 운영진들은 재밌는 의자 게임을 준비하였다. 방 안에는 1번, 2번, ...,  $N$ 번 의자가 순서대로 놓여있다. 즉, 가장 오른쪽에 있는  $N$ 번 의자를 제외하면,  $i$ 번 의자의 오른쪽에는  $i+1$ 번 의자가 있다. 각 의자에는  $N$ 명의 참가자가 앉아있으며, 모든 참가자는 1 이상  $N$  이하의 정수인 등번호를 부여받았다. 등번호는 같을 수도 있다.

게임은 우승자가 나오기 전까지 다음의 순서대로 규칙에 따라 진행된다.



1. 각 참가자는 자신의 오른쪽에 있는 의자로 이동한다. 단, 가장 오른쪽 의자에 앉아있는 참가자는 1번 의자로 이동한다.
2. 연속되게 앉아있는  $K$ 명 이상의 참가자들이 다음 조건을 만족하면, 그 참가자들이 게임에서 공동 우승한다: 그 참가자들끼리 자리를 재배열해, 자신의 등번호와 의자의 번호를 똑같이 만들 수 있다. 우승자가 없다면, 다시 규칙 1번으로 돌아간다.

하지만 이럴 수가! 게임이 진행되던 도중, 운영진들은 이 게임이 영원히 끝나지 않을 수도 있다는 것을 깨달았다. 운영진들은 참가자를 슬쩍 추가하여 이 문제를 해결하려 한다. 규칙 2번에서 규칙 1번으로 돌아가기 전, 원한다면 아래의 방식대로 참가자를 한 명 추가할 수 있다.

- 현재  $x$ 개의 의자가 있다면,  $x+1$ 번 의자를  $x$ 번 의자의 오른쪽에 추가하고, 거기에 새로운 참가자가 앉는다. 이 참가자의 등번호는 운영진이 임의로 정할 수 있다.

운영진들은 게임의 흥을 깨지 않기 위해 최소한의 참가자만을 추가하고 싶다. 게임이 언젠가는 끝나게 하기 위해서, 운영진들은 최소 몇 명의 참가자를 추가해야 할까?

### 입력

다음과 같이 입력이 주어진다.

$N$   $K$   
 $a_1$   $a_2$  ...  $a_N$

- $N$ 은 의자와 참가자의 수,  $K$ 는 우승자의 최소 인원수이다. ( $1 \leq K \leq N \leq 100000$ )



- $a_i$ 는 초기에  $i$ 번 의자에 앉은 참가자의 등번호다. ( $1 \leq a_i \leq N$ )
- 입력으로 주어지는 모든 수는 정수다.

## 출력

게임이 유한한 시간 내에 끝나기 위해서 추가해야 하는 최소 인원수를 출력한다.

## 입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 3 3 1 2	0

$[3, 1, 2] \xrightarrow{\text{규칙 1}} [2, 3, 1] \xrightarrow{\text{규칙 1}} [1, 2, 3]$

규칙 2로 인해,  $[1, 2, 3]$  이 승리한다.

표준 입력(stdin)	표준 출력(stdout)
4 3 1 4 4 4	2

$[1, 4, 4, 4] \xrightarrow{\text{규칙 1}} [4, 1, 4, 4] \xrightarrow{2 \text{ 추가}} [4, 1, 4, 4, 2] \xrightarrow{\text{규칙 1}} [2, 4, 1, 4, 4] \xrightarrow{3 \text{ 추가}} [2, 4, 1, 4, 4, 3] \xrightarrow{\text{규칙 1}} [3, 2, 4, 1, 4, 4]$

규칙 2로 인해,  $[3, 2, 4, 1, 4, 4]$  이 승리한다.





## 문제 K. 합성인수분해

시간 제한 1 초  
메모리 제한 1024 MB

소인수분해란 어떤 자연수를 소수의 곱으로 나타내는 것이다. 정수론을 끔찍하게 싫어하는 연두는 소수만 보면 치가 떨려, 대신에 자연수를 합성수의 곱으로 나타내는 “합성인수분해”라는 것을 만들었다.

자연수  $N$ 의 합성인수분해는 다음의 조건을 모두 만족하는 수열  $A$ 로 정의한다.

- $A$ 의 모든 원소는 합성수이다. (합성수란 1과 자기 자신 이외의 다른 약수를 가지는 정수이다.)
- $A$ 의 모든 원소의 곱은  $N$ 이다.

하지만 연두는  $N$ 의 합성인수분해가 여러 개이거나 존재하지 않을 수도 있다는 것을 깨달았다. 연두를 대신해  $N$ 을 합성인수분해 해주는 프로그램을 만들어보자. 만약 가능한 결과가 여러 개일 경우, 사전 순으로 가장 앞서는 것을 선택해야 한다.

### 입력

다음과 같이 입력이 주어진다.

$N$

- $2 \leq N \leq 10^{12}$
- $N$ 은 정수다.

### 출력

$N$ 의 합성인수분해 중 사전순으로 가장 앞서는 수열의 원소들을 순서대로 공백으로 구분하여 출력한다.

합성인수분해가 불가능하다면 대신에  $-1$ 을 출력한다.

### 입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3	-1
24	4 6

### 노트

수열  $A = a_1, a_2, \dots, a_n$ 가 수열  $B = b_1, b_2, \dots, b_m$ 보다 사전 순으로 앞선다는 것의 엄밀한 정의는, 다음 중 하나를 만족한다는 것이다.

- $a_1 = b_1, a_2 = b_2, \dots, a_{i-1} = b_{i-1}$  이고  $a_i < b_i$  인  $i$ 가 존재한다.
- $a_1 = b_1, a_2 = b_2, \dots, a_n = b_n$  이고  $n < m$ 이다.

2021 신춘지역  
대학생 프로그래밍 대회 동아리  
연합 겨울 대회

**SUAPC 2021** winter



Sogang ICPC Team



Algos



Morgorithm



EDOC



HI-ARC

이 페이지는 공백입니다

## 문제 L. 습격받은 도시

시간 제한 3 초  
메모리 제한 1024 MB

극악무도한 테러리스트 주현이가 도시를 습격했다. 습격받은 도시는 세로  $N$ 칸, 가로  $N$ 칸으로 이뤄진 격자 모양이며, 각 칸은 빈칸이거나 건물이 존재한다. 주현이는 자신이 만든 수제 폭탄을 건물이 없는 곳에 설치한다. 폭탄은 터질 때 상하좌우 각 방향에 대해 충격파가 퍼져나가는데, 충격파가 닿은 건물은 파괴되어 건물 잔해가 된다. 충격파는 건물 또는 건물 잔해에 닿고 난 뒤 소멸한다.



이번 테러 사건 수사를 맡은 향빈이는 현장을 재구성하는 중이다. 건물 잔해의 위치를 통해 어떤 위치에서 폭탄이 터졌는지 알아내고자 한다. 아무리 생각해도 폭탄의 위치를 알아낼 수 없는 향빈이는 문제 해결의 대가인 당신을 찾아왔다. 습격받은 도시의 정보가 주어졌을 때, 주현이가 설치한 폭탄의 위치를 구해주자.

### 입력

다음과 같이 입력이 주어진다.

$$\begin{matrix}
 N \\
 a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\
 a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\
 \vdots & \vdots & \ddots & \vdots \\
 a_{N,1} & a_{N,2} & \cdots & a_{N,N}
 \end{matrix}$$

- $N$ 은 도시의 크기이다. ( $1 \leq N \leq 2000$ )
- $a_{i,j} = .$  또는  $a_{i,j} = 0$  또는  $a_{i,j} = X$ 
  - $a_{i,j} = .$  면, 도시의 세로  $i$ 번째 가로  $j$ 번째 칸은 빈칸이다.
  - $a_{i,j} = 0$  면, 도시의 세로  $i$ 번째 가로  $j$ 번째 칸은 건물이다.
  - $a_{i,j} = X$  면, 도시의 세로  $i$ 번째 가로  $j$ 번째 칸은 건물 잔해이다.
- 항상 답이 존재하는 경우만 주어진다.

### 출력

$N$ 개의 줄에 도시의 정보를 출력한다. 각 줄은  $N$ 개의 문자를 포함하며  $i$ 번째 줄  $j$ 번째 문자는 도시의 세로  $i$ 번째 가로  $j$ 번째 칸에 대한 정보이다. 빈칸일 경우  $.$ , 건물일 경우  $0$ , 건물 잔해일 경우  $X$ , 폭탄이 설치된 칸인 경우  $B$ 이다. 답이 여러 가지인 경우, 아무거나 출력한다.



## 입출력 예시

표준 입력( <i>stdin</i> )	표준 출력( <i>stdout</i> )
5	.B.X0
...X0	..X00
..X00	..BX0
...X0	O....
O....	OXX..
OXX..	



## 문제 M. Go와 함께하는 전화망 서비스

시간 제한 1 초  
메모리 제한 1024 MB

국렬이는 NAVER D2의 “Go와 함께하는 전화망 서비스 구축” 글을 보고 Go를 통해서 전화망 서비스를 구축하려고 한다. 국렬이는 양방향으로 소통이 가능하게 만들기 위해서 전화망을 무방향 완전 그래프  $G = (V, E)$  로 구성할 것이다. 그래프의 정점은 접속한 사람을 의미하고, 간선은 접속한 사람들을 연결하는 역할을 가지고 있다. 각 간선  $(i, j) \in E$ 에 대한 접속 속도는  $x_{i,j}$ 로 표기하며 각 값은 0 이상 1 이하의 실수 값을 가질 수 있다.

NAVER는 적절하게 전화망 서비스의 접속 속도를 지정할 것이다. 전화망을 구축할 때 접속 속도의 합에 비례하는 구축 비용이 들기에 전화망에 있는 모든 간선의 속도의 합을  $|V| - 1$ 로 설정하였다. 또한, 한 사람에 너무 많은 통신이 몰리는 것을 막기 위해 정점  $v$ 와 인접한 모든 간선의 속도의 합은  $b_v$  이하로 설정하였다. 전화망의 접속을 끊는 사람이 발생할 수 있으며 끊은 사람에 해당하는 노드와 인접한 모든 간선이 삭제된다. 이때, 어떤 사람들이 접속을 끊더라도 항상 전화망에 남은 모든 간선의 속도의 합이 (남은 사람의 수 - 1) 이하가 되도록 설정하였다. NAVER는 이 조건을 만족하는 전화망을 **안정적인 전화망**이라고 정의하였다.

국렬이는 안정적인 전화망을 수학적으로 명확히 하기 위해서 몇 가지 집합 연산을 정의했다.  $\delta(v)$ 는 정점  $v$ 에 이웃한 간선의 집합이고, 정점의 부분 집합  $S$ 에 대해  $E(S)$ 는 양 끝점이 모두  $S$ 에 포함되는 간선의 집합이다. 이를 수학적으로 정의하면 다음과 같다.

- $\delta(v) = \{(u, v) : u \in V\}$
- $E(S) = \{(u, v) : u \in S, v \in S\}$

NAVER가 정의한 **안정적인 전화망**은 다음 조건을 만족하는 무방향 완전 그래프  $G = (V, E)$ 에 대한 접속 속도  $x_{i,j}$ 를 의미한다.

- $\sum_{e \in E} x_e = |V| - 1$ 를 만족해야 한다.
- 크기가 1 이상인 임의의  $S \subseteq V$ 에 대해서  $\sum_{e \in E(S)} x_e \leq |S| - 1$ 를 만족해야 한다.
- 임의의 정점  $v \in V$ 에 대해서  $\sum_{e \in \delta(v)} x_e \leq b_v$ 를 만족해야 한다.

국렬이는 전화망 서비스를 임의로 하나 만들었다. 그러나 국렬이는 귀찮아서 전화망 서비스가 안정적인지 확인하는 것을 대회를 치르는 여러분에게 맡기려고 한다. 여러분은 상금을 원한다면 국렬이의 전화망 서비스가 안정적인지 판단해야 한다.

### 입력

다음과 같이 입력이 주어진다.

```
N
b1 b2 ... bN
x1,1 x1,2 ... x1,N
x2,1 x2,2 ... x2,N
⋮ ⋮ ⋱ ⋮
xN,1 xN,2 ... xN,N
```



- $N$ 은 무방향 완전 그래프  $G$ 의 정점 개수를 의미한다. ( $2 \leq N \leq 100$ )
- $1 \leq b_i \leq N - 1$
- $b_i$ 는 양의 정수다.
- $x_{i,j}$ 는 간선  $(i, j)$ 의 접속 속도를 의미하며, 소수 첫째 자리까지 주어진다.
- $x_{i,j} = x_{j,i}$  ( $1 \leq i, j \leq N$ )
- $0.0 \leq x_{i,j} \leq 1.0$  ( $1 \leq i, j \leq N$ )
- $x_{i,i} = 0.0$  ( $1 \leq i \leq N$ )

## 출력

전화망 서비스가 안정적이면 **YES**를 출력하여야라. 그렇지 않다면 **NO**를 출력하여야라.

## 입출력 예시

표준 입력(stdin)	표준 출력(stdout)
<pre>3 2 1 2 0.0 0.5 1.0 0.5 0.0 0.5 1.0 0.5 0.0</pre>	YES
<pre>3 2 1 1 0.0 0.5 1.0 0.5 0.0 0.5 1.0 0.5 0.0</pre>	NO
<pre>4 3 3 3 3 0.0 1.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0</pre>	NO
<pre>4 3 3 3 3 0.0 1.0 1.0 0.0 1.0 0.0 1.0 0.0 1.0 1.0 0.0 1.0 0.0 0.0 1.0 0.0</pre>	NO



표준 입력(stdin)	표준 출력(stdout)
4 3 3 3 3 0.0 1.0 0.0 0.0 1.0 0.0 1.0 0.0 0.0 1.0 0.0 1.0 0.0 0.0 1.0 0.0	YES

2021 신춘지역  
대학생 프로그래밍 대회 동아리  
연합 겨울 대회

**SUAPC 2021** winter



Sogang ICPC Team



Algos



Morgorithm



EDOC



HI-ARC

이 페이지는 공백입니다