

2023 SCCC Contest #1

Official Problem Set



Sponsored By:

HYUNDAI
MOBIS

대회 중 유의 사항

- 2023 SCCC 내부 대회 유의 사항입니다.
- 대회 진행 관련
 - 대회는 2시간 동안 8문제로 진행됩니다.
 - 대회 도중에는 대회 운영진을 제외한 타인과 대화할 수 없습니다.
 - 인터넷 검색을 허용하나, 제출하는 모든 코드는 대회 시간 도중에 본인이 직접 작성한 코드여야 합니다.
 - ChatGPT, Copilot, Bard 등 AI를 활용하여 생성한 코드도 사용할 수 없습니다.
 - 단, 외국어 문제 번역을 위해 GPT 등의 AI를 활용하는 것은 가능합니다.
 - 문제와 관련된 질문은 슬랙(21 나정희) 또는 디스코드(jhnah917) DM을 이용해야 합니다.
- 문제 관련
 - 문제는 운영진들이 생각하는 난이도순으로 정렬되어 있습니다.
 - 모든 문제는 C++17로 해결할 수 있음이 보장됩니다.
 - 제출한 프로그램은 문제에 명시된 제한 시간 내에 정답을 출력하고 정상적으로 종료되어야 합니다. 이는 return code가 0이어야 함을 의미하여, 이외의 exit code는 런타임 에러가 발생합니다.
 - 제출한 프로그램은 문제에 명시된 제한 메모리보다 많은 메모리를 사용할 수 없습니다.
 - 제출한 프로그램은 표준 입력(standard input)을 통해 입력받아서 표준 출력(standard output)을 통해 정답을 출력해야 합니다.
 - 표준 입출력을 제외한 파일 입출력, 네트워킹, 멀티 스레딩 등의 시스템 콜은 사용할 수 없습니다.
- 대회 상품
 - 1등: 5만원
 - $solve^3$ 가중치로 3명 추첨: BBQ 황금올리브치킨 + 콜라 1.25L
 - $solve^2$ 가중치로 10명 추첨: 스타벅스 아이스 카페 아메리카노 T 2잔 + 부드러운 생크림 카스텔라
 - $solve^1$ 가중치로 10명 추첨: 스타벅스 아이스 카페 아메리카노 T
- 등수는 다음과 같은 방법을 이용해 계산합니다.
 - 문제의 패널티 = (대회가 시작한 시점으로부터 처음으로 맞았습니다!!를 받기까지 걸린 분 단위 시간) + (제출 횟수 - 1) * 20분
 - 팀의 패널티 = 맞았습니다!!를 받은 모든 문제의 패널티의 합
 - 팀의 등수 = (더 많은 문제를 푼 팀의 수) + (푼 문제의 개수가 동일하면서 패널티가 더 작은 팀의 수) + 1

언어 가이드

- 채점은 Intel Xeon E5-2666v3 프로세서를 사용하는 AWS EC2 c4.large 인스턴스에서 진행합니다.
- 채점 서버의 운영체제는 Ubuntu 16.04.7 LTS입니다.
- BOJ에서 지원하는 모든 언어를 자유롭게 사용할 수 있습니다.
- 각 언어의 컴파일과 실행 옵션은 <https://help.acmicpc.net/language/info>에서 확인할 수 있습니다.
- C11/C++17에서 `scanf_s`와 `Windows.h`등의 비표준 함수를 사용할 수 없습니다.
- Java를 사용하는 경우, `main` 메소드를 포함하는 클래스의 이름은 `Main`이어야 합니다.
- Python에서 `numpy`와 같은 외부 모듈을 사용할 수 없습니다.
- 채점 사이트에서 컴파일 에러를 받은 경우, ‘컴파일 에러’ 글씨를 누르면 오류가 발생한 위치를 볼 수 있습니다.
- 아래 코드는 표준 입력(standard input)을 통해 공백으로 구분된 두 정수를 입력으로 받아서 표준 출력(standard output)을 통해 합을 출력하는 코드입니다.

– C11

```
1 #include <stdio.h>
2
3 int main() {
4     int a, b;
5     scanf("%d %d", &a, &b);
6     printf("%d\n", a + b);
7     return 0;
8 }
```

– C++17

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int a, b;
6     cin >> a >> b;
7     cout << a + b << endl;
8     return 0;
9 }
```

– Java 15

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
```

```
5     Scanner sc = new Scanner(System.in);
6     int a = sc.nextInt();
7     int b = sc.nextInt();
8     System.out.println(a + b);
9     sc.close();
10    }
11 }
```

– Python 3 / PyPy3

```
1 a, b = map(int, input().split())
2 print(a + b)
```

- 입출력 양이 많을 때는 위 코드를 사용한 입출력이 너무 오래 걸리기 때문에 다른 방식으로 입출력해야 합니다.
- C11/C++17에서 `scanf`과 `printf`을 사용하는 경우, 입출력 속도는 문제를 해결할 수 있을 정도로 충분히 빠릅니다.
- C++17에서 `cin`과 `cout`을 사용하는 경우, 입출력 전에 `ios_base::sync_with_stdio(false);`와 `cin.tie(nullptr);`를 사용하여야 합니다. 단, 이 이후에는 `cin`, `cout` 계열 함수와 `scanf`, `printf` 계열 함수를 섞어서 사용하면 안 됩니다. 또한, 개행문자로 `std::endl` 대신 `\n`을 사용해 주세요.
- Java 15에서는 `BufferedReader`와 `BufferedWriter`를 사용하여야 합니다.
- Python 3 및 PyPy3 에서는 `input()` 대신 `sys.stdin.readline().rstrip("\n")`을 사용하여야 합니다. 코드의 가장 위 부분에 `import sys` 와 `input = lambda: sys.stdin.readline().rstrip("\n")`을 사용하여야 합니다.
- 아래 코드는 표준 입력(standard input)을 통해 문제의 개수 T 를 입력받은 다음 T 줄에 걸쳐 공백으로 구분된 두 정수를 입력으로 받아 표준 출력(standard output)을 통해 두 정수의 합을 총 T 줄에 걸쳐 출력하는 코드입니다.

– C++17

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     ios_base::sync_with_stdio(false);
6     cin.tie(nullptr);
7     int T;
8     cin >> T;
9     for(int i=1; i<=T; i++){
10         int a, b;
11         cin >> a >> b;
12         cout << a + b << "\n"; // do not use endl
13     }
}
```

```
14     return 0;  
15 }
```

– Java 15

```
1 import java.util.*;  
2 import java.io.*;  
3  
4 public class Main{  
5     public static void main(String[] args) throws IOException {  
6         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
7         BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(System.out));  
8  
9         int T = Integer.parseInt(br.readLine());  
10        for(int i=1; i<=T; i++){  
11            String[] temp = br.readLine().split(" ");  
12            int a = Integer.parseInt(temp[0]);  
13            int b = Integer.parseInt(temp[1]);  
14            bw.write(String.valueOf(a + b) + "\n");  
15        }  
16        br.close();  
17        bw.close();  
18    }  
19 }
```

– Python 3 / PyPy3

```
1 import sys  
2 input = lambda: sys.stdin.readline().rstrip("\n")  
3  
4 T = int(input())  
5 for _ in range(T):  
6     a, b = map(int, input().split())  
7     print(a + b)
```

2023 SCCC Contest #1

Problem List

#	Problem Name	Time limit	Memory limit	Page
A	BOJ 28938 컨베이어 벨트	2 seconds	1024MiB	7 – 7
B	BOJ 29558 축제 부스 기획 1	2 seconds	1024MiB	8 – 8
C	BOJ 28813 축제 부스 기획 2	2 seconds	1024MiB	9 – 10
D	BOJ 28601 축제 부스 기획 3	1 second	1024MiB	11 – 11
E	BOJ 28841 축제 부스 기획 4	2 seconds	1024MiB	12 – 12
F	BOJ 28744 단순한 문제	2 seconds	1024MiB	13 – 13
G	BOJ 28944 간단한 문제	2 seconds	1024MiB	14 – 14
H	BOJ 28730 바둑알 배치하기	2 seconds	1024MiB	15 – 15

문제지에 있는 문제가 총 8문제가 맞는지 확인하시길 바랍니다.

모든 문제는 C++17로 풀 수 있음을 보장합니다.

A. 컨베이어 벨트 (BOJ 28938)

승실이는 공장에서 일자 형태의 길이가 무한한 컨베이어 벨트를 조작하는 업무를 담당하고 있다. 컨베이어 벨트의 제어판에는 각각 $-1, 0, 1$ 이 적혀 있는 세 개의 버튼이 있다. -1 이 적혀 있는 버튼을 누르면 컨베이어 벨트가 왼쪽으로 1미터 이동하고, 1 이 적혀 있는 버튼을 누르면 오른쪽으로 1미터 이동하며, 0 이 적혀 있는 버튼을 누르면 아무런 변화가 발생하지 않는다.

승실이가 버튼을 누른 횟수와 누른 버튼에 적혀 있는 수가 주어지면, 컨베이어 벨트가 초기 상태와 비교했을 때 어느 방향으로 이동했는지 구해보자.

입력 형식

첫 번째 줄에 승실이가 버튼을 누른 횟수 N 이 주어진다.

두 번째 줄에 승실이가 누른 버튼에 적힌 N 개의 수 A_1, A_2, \dots, A_N 이 공백으로 구분되어 주어진다.

출력 형식

버튼을 N 번 누른 이후에 컨베이어 벨트가 왼쪽으로 이동했으면 “Left”, 오른쪽으로 이동했으면 “Right”, 초기 상태와 동일하다면 “Stay”를 출력한다.

제한

- $1 \leq N \leq 1000$
- $A_i \in \{-1, 0, 1\}$
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고

B. 축제 부스 기획하기 1 (BOJ 29558)

SCCC 운영진은 내년 봄 축제에서 SCCC 부스를 열기 위해 부스에서 진행할 게임을 만드는 데 물두하고 있다.

정휘, 찬솔, 성서, 주원은 각자 게임을 하나씩 준비해 왔다. 4명 모두 자기가 준비한 게임이 제일 재미있다고 주장하는 바람에 어떤 게임을 축제에서 사용할지 결정하지 못하고 있다. 한 달 동안 의견 차이를 좁히지 못한 운영진들은 결국 소모임 내부 대회를 통해 가장 재미있는 게임을 정하기로 했다.

여러분들은 앞으로 4개의 문제를 통해 게임들을 직접 플레이해 보면서, 어떤 게임이 가장 재미 있는지 판단해야 한다.

정휘가 준비한 게임의 규칙은 다음과 같다.

- 게임은 1명의 진행자와 여러 명의 참가자로 구성되어 있다.
- 진행자는 양의 정수 N 과 정수 D 를 외친다.
- 길이가 N 이고 평균이 D 이면서, 모든 원소의 값이 서로 다른 정수인 수열을 가장 먼저 외치는 사람이 승리한다.

N 과 D 가 주어지면 모든 원소가 서로 다른 정수이면서 평균이 D 인 수열 A_1, A_2, \dots, A_N 을 구하는 프로그램을 작성하자.

입력 형식

첫 번째 줄에 두 정수 N, D 가 공백으로 구분되어 주어진다.

출력 형식

모든 원소가 서로 다른 정수이면서 평균이 D 인 수열 A_1, A_2, \dots, A_N 을 공백으로 구분하여 출력한다.

만약 가능한 수열이 여러 가지인 경우에는 아무거나 출력해도 된다.

제한

- $1 \leq N \leq 1\,000$
- $|D| \leq 1\,000$
- $|A_i| \leq 10\,000$
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고

C. 축제 부스 기획하기 2 (BOJ 28813)

SCCC 운영진은 내년 봄 축제에서 SCCC 부스를 열기 위해 부스에서 진행할 게임을 만드는 데 몰두하고 있다.

정휘, 찬솔, 성서, 주원은 각자 게임을 하나씩 준비해 왔다. 4명 모두 자기가 준비한 게임이 제일 재미있다고 주장하는 바람에 어떤 게임을 축제에서 사용할지 결정하지 못하고 있다. 한 달 동안 의견 차이를 좁히지 못한 운영진들은 결국 소모임 내부 대회를 통해 가장 재미있는 게임을 정하기로 했다.

여러분들은 앞으로 4개의 문제를 통해 게임들을 직접 플레이해 보면서, 어떤 게임이 가장 재미 있는지 판단해야 한다.

찬솔이가 준비한 게임의 규칙은 다음과 같다.

- 게임은 1명의 참가자로 구성되어 있다.
- 게임은 이차원 좌표 평면상에서 진행되고, 참가자는 원점에 있다.
- 총 N 개의 드론이 참가자가 있는 곳을 향해 이동한다.
- i 번째 드론은 처음에 (x_i, y_i) 에 위치해 있다.
- 각 드론은 1초마다 자신의 x좌표와 y좌표를 0 또는 1 만큼 변화시킬 수 있다. 즉, 드론은 매초마다 가만히 있거나 상하좌우 또는 대각선 방향으로 한 칸 이동할 수 있다.
- 참가자는 1초마다 원하는 드론 하나를 제거할 수 있다.
- 같은 위치에 두 개 이상의 드론이 위치할 수 있지만, 드론은 매번 하나만 제거할 수 있다.
- 참가자가 드론을 제거한 다음에 드론이 이동한다. 따라서 게임은 (드론 제거) \rightarrow (드론 이동) \rightarrow (드론 제거) $\rightarrow \dots$ 순으로 진행된다.
- 드론이 참가자가 있는 곳에 도달하기 전에 모든 드론을 없앨 수 있으면 참가자가 승리한다.

드론의 초기 위치가 주어지면 모든 드론을 제거할 수 있는지 판별하고, 만약 가능하다면 드론을 제거하는 순서를 구하는 프로그램을 작성하자.

입력 형식

첫 번째 줄에 드론의 개수 N 이 주어진다.

두 번째 줄부터 N 개의 줄에 걸쳐, 한 줄에 하나씩 드론의 초기 위치 x_i, y_i 가 공백으로 구분되어 주어진다.

출력 형식

만약 모든 드론을 없애는 방법이 존재하지 않는다면 -1 을 출력한다.

그렇지 않은 경우, 드론을 제거하는 순서를 공백으로 구분하여 출력한다. i 번째로 출력하는 수는 i 번째로 없애는 드론의 번호를 의미한다.

만약 가능한 제거 순서가 여러 가지인 경우에는 아무거나 출력해도 된다.

제한

- $1 \leq N \leq 100\,000$
- $|x_i|, |y_i| \leq 100\,000$
- 드론의 초기 위치는 원점이 아니다.
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고

D. 축제 부스 기획하기 3 (BOJ 28601)

SCCC 운영진은 내년 봄 축제에서 SCCC 부스를 열기 위해 부스에서 진행할 게임을 만드는 데 몰두하고 있다.

정휘, 찬솔, 성서, 주원은 각자 게임을 하나씩 준비해 왔다. 4명 모두 자기가 준비한 게임이 제일 재미있다고 주장하는 바람에 어떤 게임을 축제에서 사용할지 결정하지 못하고 있다. 한 달 동안 의견 차이를 좁히지 못한 운영진들은 결국 소모임 내부 대회를 통해 가장 재미있는 게임을 정하기로 했다.

여러분들은 앞으로 4개의 문제를 통해 게임들을 직접 플레이해 보면서, 어떤 게임이 가장 재미 있는지 판단해야 한다.

성서가 준비한 게임의 규칙은 다음과 같다.

- 게임은 1명의 진행자와 각각 3명으로 구성된 팀 여러 개로 구성되어 있다.
- 게임은 직사각형 형태의 운동장에서 진행하며, 모든 사람은 한 꼭짓점에 모여 시작한다.
- 각 사람은 매번 직사각형의 변에 평행한 방향으로 1씩 이동할 수 있다.
- 진행자는 음이 아닌 정수 S 를 외친다.
- 각 팀은 팀원들의 위치를 꼭짓점으로 하는 삼각형의 넓이가 $S/2$ 이상이 되도록 적절히 이동해야 한다.
- 팀원들의 이동 거리의 합이 최소인 팀이 승리한다.

예를 들어 $S = 3$ 이면 첫 번째 팀원은 원점에 가만히 있고, 두 번째 팀원은 가로 방향으로 1칸, 세로 방향으로 2칸, 세 번째 팀원은 가로 방향으로 2칸, 세로 방향으로 1칸 이동해서 넓이가 $3/2$ 인 삼각형을 만들 수 있다. 이때 팀원들의 이동 거리의 합은 6이다.

S 가 주어지면 세 팀원의 위치를 꼭짓점으로 하는 삼각형의 넓이를 $S/2$ 이상으로 만들기 위해 필요한 이동 거리의 최솟값을 구하는 프로그램을 작성하자.

입력 형식

첫 번째 줄에 음이 아닌 정수 S 가 주어진다.

출력 형식

세 팀원의 위치를 꼭짓점으로 하는 삼각형의 넓이를 $S/2$ 이상으로 만들기 위해 필요한 이동 거리의 최솟값을 구하는 프로그램을 작성하자.

제한

- $0 \leq S \leq 10^{18}$
- 입력으로 주어지는 수는 모두 정수이다.
- 입력과 출력이 32비트 정수 범위를 넘을 수 있음에 주의하자.

예제

예제는 채점 사이트 참고

E. 축제 부스 기획하기 4 (BOJ 28841)

SCCC 운영진은 내년 봄 축제에서 SCCC 부스를 열기 위해 부스에서 진행할 게임을 만드는 데 몰두하고 있다.

정휘, 찬솔, 성서, 주원은 각자 게임을 하나씩 준비해 왔다. 4명 모두 자기가 준비한 게임이 제일 재미있다고 주장하는 바람에 어떤 게임을 축제에서 사용할지 결정하지 못하고 있다. 한 달 동안 의견 차이를 좁히지 못한 운영진들은 결국 소모임 내부 대회를 통해 가장 재미있는 게임을 정하기로 했다.

여러분들은 앞으로 4개의 문제를 통해 게임들을 직접 플레이해 보면서, 어떤 게임이 가장 재미 있는지 판단해야 한다.

주원이가 준비한 게임의 규칙은 다음과 같다.

- 게임은 1명의 진행자와 여러 명의 참가자로 구성되어 있다.
- 진행자는 참가자들에게 0과 1로만 구성된 문자열 S 를 제공한다.
- 참가자는 아래 조건을 만족하는 두 구간 $[l_1, r_1]$, $[l_2, r_2]$ 를 찾아야 한다.
 - $l_1 \leq r_1$
 - $l_2 \leq r_2$
 - $[l_1, r_1]$ 과 $[l_2, r_2]$ 는 서로 다른 구간이다. 즉, $l_1 \neq l_2$ 또는 $r_1 \neq r_2$ 를 만족해야 한다.
 - $r_1 - l_1 = r_2 - l_2$
 - $\sum_{i=l_1}^{r_1} S[i] = \sum_{j=l_2}^{r_2} S[j]$
- $r_1 - l_1$ 가장 큰 구간을 찾는 참가자가 승리한다.

문자열 S 가 주어지면, 위 조건을 만족하는 가장 긴 두 구간 $[l_1, r_1]$, $[l_2, r_2]$ 를 구하는 프로그램을 작성하자.

입력 형식

첫 번째 줄에 0과 1로만 구성되어 있는 문자열 S 가 주어진다.

출력 형식

조건을 만족하는 두 구간이 존재하지 않으면 -1을 출력한다.

그렇지 않은 경우, l_1, r_1, l_2, r_2 를 순서대로 공백으로 구분하여 출력한다.

만약 가능한 구간이 여러 가지인 경우에는 아무거나 출력해도 된다.

제한

- $1 \leq |S| \leq 1\,000\,000 = 10^6$
- S 는 0과 1로만 구성된 문자열이다.

예제

예제는 채점 사이트 참고

F. 단순한 문제 (BOJ 28744)

길이가 N 인 수열 A_1, A_2, \dots, A_N 이 주어진다. 아래 식을 계산하는 프로그램을 작성하자.

$$\sum_{1 \leq i < j < k \leq N} A_i \cdot A_j \cdot A_k = \sum_{i=1}^{N-2} \sum_{j=i+1}^{N-1} \sum_{k=j+1}^N A_i \cdot A_j \cdot A_k$$

단, 결과가 너무 클 수 있으므로 $1\,000\,000\,007 = 10^9 + 7$ 로 나눈 나머지를 출력하라.

입력 형식

첫 번째 줄에 수열의 길이 N 이 주어진다.

두 번째 줄에 N 개의 정수 A_1, A_2, \dots, A_N 이 공백으로 구분되어 주어진다.

출력 형식

주어진 수식을 계산한 결과를 $10^9 + 7$ 로 나눈 나머지를 출력한다.

제한

- $3 \leq N \leq 1\,000\,000 = 10^6$
- $0 \leq A_i \leq 1\,000\,000 = 10^6$
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고

G. 간단한 문제 (BOJ 28944)

길이가 N 인 수열 A_0, A_1, \dots, A_{N-1} 이 주어진다. 모든 $i < j$ 에 대해 $A_i - A_j < j - i$ 를 만족하도록 몇 개의 수만 선택하려고 할 때, 선택할 수 있는 수의 최대 개수를 구하는 프로그램을 작성하라. 이때, i, j 는 입력으로 주어진 수열의 인덱스임에 주의하라.

입력 형식

첫 번째 줄에 수열의 길이 N 이 주어진다.

두 번째 줄에 N 개의 정수 A_0, A_1, \dots, A_{N-1} 이 공백으로 구분되어 주어진다.

출력 형식

모든 $i < j$ 에 대해, $A_i - A_j < j - i$ 를 만족하도록 선택할 수 있는 원소의 최대 개수를 출력한다.

제한

- $1 \leq N \leq 1\,000\,000 = 10^6$
- $1 \leq A_i \leq 10^9$
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고

참고

첫 번째 예제는 $A_0 = 1, A_1 = 2, A_2 = 2$ 인 입력이 주어진다. A_0, A_1, A_2 를 모두 선택해도 부등식이 성립함을 알 수 있다.

- $A_0 - A_1 = 1 - 2 < 1 - 0$
- $A_1 - A_2 = 2 - 2 < 2 - 1$
- $A_0 - A_2 = 1 - 2 < 2 - 0$

두 번째 예제는 A_0, A_2, A_3, A_5 를 선택하는 것이 최적이다.

- $A_0 - A_2 = 5 - 5 < 2 - 0$
- $A_0 - A_3 = 5 - 6 < 3 - 0$
- $A_0 - A_5 = 5 - 5 < 5 - 0$
- $A_2 - A_3 = 5 - 6 < 3 - 2$
- $A_2 - A_5 = 5 - 5 < 5 - 2$
- $A_3 - A_5 = 6 - 5 < 5 - 3$

H. 바둑알 배치하기 (BOJ 28730)

1×1 크기의 정사각형 칸들이 체크무늬로 칠해져 있는 $N \times N$ 크기의 체스판이 주어진다. 여러분은 아래 조건을 충족하도록 체스판에 바둑알을 배치해야 한다.

- 체스판의 각 칸에는 최대 한 개의 바둑알만 배치할 수 있다.
- 바둑알은 체스판의 한 칸에 정확히 포함되도록 배치해야 한다.
- 1×1 크기의 도미노는 최대 N 개 배치할 수 있다.
- 바둑알이 없는 칸들은 모두 1×2 크기의 도미노를 이용해 덮을 수 있어야 한다.
- 각 도미노는 체스판 상에서 인접하는(변을 공유하는) 두 칸을 덮도록 배치해야 한다.
- 도미노는 바둑알이 있는 칸을 덮을 수 없다.
- 두 개 이상의 도미노가 같은 칸을 덮을 수 없다.
- 도미노를 이용해 바둑알이 없는 칸을 덮는 방법이 유일해야 한다.

N 이 주어지면 조건을 만족하도록 체스판에 바둑알을 배치하는 방법을 찾는 프로그램을 작성하자. 조건을 만족하는 배치 방법은 항상 존재한다.

입력 형식

첫 번째 줄에 체스판의 한 변의 길이를 나타내는 정수 N 이 주어진다.

출력 형식

N 개의 줄에 걸쳐 정답을 출력한다.

i 번째 줄에는 체스판의 i 번째 행의 상태를 출력한다. 바둑알이 배치되어 있는 칸은 #, 배치되어 있지 않은 칸은 .을 출력한다.

만약 가능한 배치가 여러 가지인 경우에는 아무거나 출력해도 된다.

제한

- $2 \leq N \leq 100$
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고