2023 SCCC Contest #3

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 나정휘) 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 LTS 입니다.
- 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

```c
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d", &a, &b);
    printf("%d\n", a + b);
    return 0;
}
```

  - C++17

```cpp
#include <iostream>
using namespace std;

int main() {
    int a, b;
    cin >> a >> b;
    cout << a + b << endl;
    return 0;
}
```

  - Java 15

```java
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        int b = sc.nextInt();
        System.out.println(a + b);
        sc.close();
    }
}
```

– Python 3 / PyPy3

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

- 입출력 양이 많을 때는 위 코드를 사용한 입출력이 너무 오래 걸리기 때문에 다른 방식으로 입출력해야 합니다.
- C11/C++17에서 `scanf`와 `printf`를 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 T를 입력받은 다음 T줄에 걸쳐 공백으로 구분된 두 정수를 입력으로 받아 표준 출력(standard output)을 통해 두 정수의 합을 총 T줄에 걸쳐 출력하는 코드입니다.

– C++17

```
#include <iostream>
using namespace std;

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int T;
    cin >> T;
    for(int i=1; i<=T; i++){
        int a, b;
        cin >> a >> b;
        cout << a + b << "\n"; // do not use endl
    }
```

```
    return 0;
}
```

– Java 15

```
import java.util.*;
import java.io.*;

public class Main{
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(System.out));

        int T = Integer.parseInt(br.readLine());
        for(int i=1; i<=T; i++){
            String[] temp = br.readLine().split(" ");
            int a = Integer.parseInt(temp[0]);
            int b = Integer.parseInt(temp[1]);
            bw.write(String.valueOf(a + b) + "\n");
        }
        br.close();
        bw.close();
    }
}
```

– Python 3 / PyPy3

```
import sys
input = lambda: sys.stdin.readline().rstrip("\n")

T = int(input())
for _ in range(T):
    a, b = map(int, input().split())
    print(a + b)
```

2023 SCCC Contest #3

Problem List

#	Problem Name	Time limit	Memory limit	Page
A	BOJ 7173 수업 참여도	1 second	1024MiB	7 – 7
B	BOJ 7239 톱니바퀴 수열	1 second	1024MiB	8 – 8
C	BOJ 28789 방 탈출	2 seconds	1024MiB	9 – 9
D	BOJ 7257 잔디깎기	1 second	1024MiB	10 – 10
E	BOJ 30168 격자 색칠	1 second	1024MiB	11 – 11
F	BOJ 7255 도로 건설하기	1 second	1024MiB	12 – 12
G	BOJ 28608 수열 만들기	1 second	1024MiB	13 – 13
H	BOJ 28753 블록 게임	2 seconds	1024MiB	14 – 14

문제지에 있는 문제가 총 8문제가 맞는지 확인하시길 바랍니다.

모든 문제는 C++17로 풀 수 있음을 보장합니다.

A. 수업 참여도 (BOJ 7173)

책상이 M행 N열로 배열되어 있는 교실에 $M \times N$명의 학생이 각자 자리에 앉아있다. 모든 학생은 컴퓨터와 스포츠 중 하나 이상에 관심이 있다. 컴퓨터에만 관심 있는 학생의 관심도는 9, 스포츠에만 관심 있는 학생의 관심도는 0, 그리고 둘 모두에 관심 있는 학생은 그 정도에 따라 1–8 중 하나의 정수로 관심도를 표현할 수 있다고 하자.

관심사가 비슷한 학생들이 인접한 위치에 있으면 서로 수다를 떨기 때문에 수업에 집중하지 않는다. 구체적으로, 각 학생의 수업 참여도는 그 학생과 앞, 뒤, 왼쪽, 오른쪽으로 인접한 학생의 관심도와의 차이의 평균으로 결정된다.

교실 전체의 수업 참여도는 모든 학생의 수업 참여도의 합과 같다. 교실 전체의 수업 참여도를 구하는 프로그램을 작성해 보자.

입력 형식

첫 번째 줄에 교실의 행의 개수와 열의 개수를 나타내는 두 정수 M과 N이 주어진다.

이후 M개의 줄에 걸쳐, i번째 줄에 i번째 행에 앉은 학생의 관심도 $A_{i,1}, A_{i,2}, \cdots, A_{i,N}$이 차례대로 공백 없이 주어진다.

출력 형식

교실 전체의 수업 참여도를 소숫점 아래 4자리까지 출력한다. 정답을 `X`라고 하면, C언어는 `printf("%.4lf", X);`, C++은 `cout << fixed << setprecision(4) << X;`를 이용해 출력하면 된다.

제한

- $1 \le M, N \le 200$
- $M \times N \ge 2$
- $0 \le A_{i,j} \le 9$
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고

두 번째 행의 네 번째 열에 있는 학생의 수업 참여도는 $(|7-1| + |7-3| + |7-8|)/3 = 11/3 = 3.6666667$이다. 교실 전체의 수업 참여도는 46.5이다.

B. 톱니바퀴 수열 (BOJ 7239)

아래 두 부등식 중 하나 이상을 만족하는 수열 $B = \{B_1, B_2, B_3, \cdots\}$를 **톱니바퀴 수열**이라고 하자.

- $B_1 < B_2 > B_3 < B_4 > B_5 < B_6 > B_7 \cdots$
- $B_1 > B_2 < B_3 > B_4 < B_5 > B_6 < B_7 \cdots$

서로 다른 정수로 구성된 길이가 N인 수열 $A = \{A_1, A_2, \cdots, A_N\}$이 주어진다. A의 원소를 정확히 한 번씩 사용해서 만들 수 있는 톱니바퀴 수열을 구하는 프로그램을 작성하자.

입력 형식

첫 번째 줄에 수열의 길이 N이 주어진다.

두 번째 줄에 수열의 원소 $A_1, A_2, \cdots, A_N$이 공백으로 구분되어 주어진다.

출력 형식

입력으로 주어진 수열의 원소들을 정확히 한 번씩 사용해서 만든 톱니바퀴 수열을 공백으로 구분해서 출력한다. 만약 가능한 정답이 여러 가지이면 아무거나 출력해도 된다.

제한

- $1 \le N \le 1\,000$
- $1 \le A_i \le 10\,000$
- $i \neq j$이면 $A_i \neq A_j$
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고

첫 번째 예제에서 `15 1 5 3 9 4 8 7` 또는 `4 8 7 15 1 5 3 9` 등을 출력해도 정답으로 인정된다.

C. 방 탈출 (BOJ 28789)

방 탈출 게임을 하던 시루는 마지막 문제에 봉착했다. 책상 위에 놓인 쪽지에는 알파벳 소문자로 구성된 문자열 s와 함께 다음과 같은 힌트가 적혀 있다.

- 마지막 문의 비밀번호는 s의 서로 다른 두 위치를 선택한 다음, 두 위치에 있는 문자를 한 번 교환해서 얻을 수 있다.

시루는 가능한 모든 경우를 시도해 보려고 했지만, 경우의 수가 너무 많아 일일이 시도할 수 없다는 것을 깨달았다. 위 단서를 토대로, 비밀번호가 될 수 있는 문자열의 개수를 구하는 프로그램을 작성해 보자.

입력 형식

첫 번째 줄에 문자열 s가 주어진다.

출력 형식

비밀번호가 될 수 있는 문자열의 개수를 출력한다.

제한

- $1 \le |s| \le 10^5$
- s는 알파벳 소문자로 구성된 문자열이다.

예제

예제는 채점 사이트 참고

D. 잔디깎기 (BOJ 7257)

시루네 집 마당에는 N개의 잔디가 있다. i번째 잔디의 길이는 A_i이다.

시루는 마당에 자란 잔디를 깎기 위해 예초기를 구매했다. 시루는 예초기를 이용해 M일 동안 잔디를 깎으려고 하며, $j(1 \le j \le M)$번째 날의 계획은 다음과 같다.

1. 아침에는 아직 완전히 잘려 나가지 않은$(A_i > 0)$ 모든 잔디가 1 만큼 자라난다.
2. 낮 동안 시루는 예초기를 이용해 잔디를 총 B_j번 깎는다. 잔디를 한 번 깎을 때마다 완전히 잘려 나가지 않은 잔디의 높이가 1 만큼 감소한다.
3. 그 후 시루는 아직 잘리지 않은 잔디의 총길이가 얼마나 되는지 계산해서 노트에 적는다.

마당에 있는 잔디의 길이와 시루가 매일 잔디를 깎는 횟수가 주어지면, 시루가 노트에 적게 되는 M개의 정수를 구하는 프로그램을 작성해 보자.

입력 형식

첫 번째 줄에 잔디의 개수 N이 주어진다.

두 번째 줄에 각 잔디의 길이 $A_1, A_2, \cdots, A_N$이 공백으로 구분되어 주어진다.

세 번째 줄에 잔디를 깎는 일수 M이 주어진다.

네 번째 줄에 M개의 양의 정수 $B_1, B_2, \cdots, B_M$이 주어진다. B_j는 j번째 날 잔디를 깎는 횟수를 의미한다.

출력 형식

M개의 줄에 걸쳐 정답을 출력한다. i번째 줄에는 i일째가 끝났을 때 아직 잘리지 않은 잔디의 총길이를 출력한다.

제한

- $1 \le N, M \le 10^5$
- $1 \le A_i, B_j \le 10^6$
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고

첫 번째 예시의 진행 과정은 문제 지문의 이미지에서 확인할 수 있다.

E. 격자 색칠 (BOJ 30168)

H개의 행과 W개의 열로 구성된 격자가 주어진다. 격자의 각 칸은 흰색 또는 검은색으로 칠할 수 있다. 이때, R_i와 C_j를 다음과 같이 정의하자.

- R_i는 i번째 행의 가장 왼쪽 칸부터 시작해서 오른쪽으로 연속되어 검은색으로 칠해져 있는 칸의 개수로 정의한다. 만약 i번째 행의 가장 왼쪽 칸이 흰색이면 $R_i = 0$이다.
- C_j는 j번째 열의 가장 위쪽 칸부터 시작해서 아래쪽으로 연속되어 검은색으로 칠해져 있는 칸의 개수로 정의한다. 만약 j번째 열의 가장 위쪽 칸이 흰색이면 $C_j = 0$이다.

다시 말해, i번째 행은 정확히 R_i개의 검은색 칸으로 시작하고 $R_i + 1$번째 칸은 존재하지 않거나 흰색으로 칠해져 있다는 것을 의미한다. 비슷하게, j번째 열은 정확히 C_j개의 검은색 칸으로 시작하고 $C_j + 1$번째 칸은 존재하지 않거나 흰색으로 칠해져 있다는 것을 의미한다.

문제 그림 참고

격자의 크기를 나타내는 H, W와 R_i, C_j가 주어졌을 때, 조건을 만족하도록 격자의 각 칸을 색칠하는 경우의 수를 구하는 프로그램을 작성해 보자. 단, 정답이 너무 클 수 있으므로 $10^9 + 7$로 나눈 나머지를 출력한다.

입력 형식

첫 번째 줄에 격자의 행 개수와 열 개수를 나타내는 H, W가 공백으로 구분되어 주어진다.

두 번째 줄에 $R_1, R_1, \cdots, R_H$가 공백으로 구분되어 주어진다.

세 번째 줄에 $C_1, C_2, \cdots, C_W$가 공백으로 구분되어 주어진다.

출력 형식

조건을 만족하면서 격자를 칠하는 경우의 수를 $1\,000\,000\,007 = (10^9 + 7)$로 나눈 나머지를 출력한다.

제한

- $1 \le H, W \le 1\,000$
- $0 \le R_i \le W$
- $0 \le C_j \le H$
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고

문제에 있는 두 개의 그림이 첫 번째 예시에서 가능한 격자 상태이다.

두 번째 예시에서 조건을 만족하는 격자는 존재하지 않는다.

세 번째 예시에서 격자를 칠하는 경우의 수는 797922655보다 더 많지만, $10^9 + 7$로 나눈 나머지를 출력한 것에 주의하라.

F. 도로 건설하기 (BOJ 7255)

SCCC국(國)은 N개의 도시로 구성되어 있는 나라이다. SCCC국의 대통령 시루는 국민들의 편의를 위해 모든 도시가 도로를 통해 서로 연결되도록 도로를 건설하려고 한다.

시루는 먼저 각 도시에 거주하는 주민의 수 $S_1, S_2, \cdots, S_N$을 파악했다. 도로를 이용하는 주민이 많을수록 도로를 더 튼튼하게 지어야 하므로, 도로가 연결하는 두 도시의 주민이 많을수록 도로를 건설하는 데 필요한 비용이 많이 든다. 구체적으로, u번 도시와 v번 도시를 연결하는 도로를 건설하기 위해서는 $S_u \times S_v$ 만큼의 비용이 필요하다.

SCCC국에 이미 건설되어 있는 도로들의 정보가 주어지면, 도로를 통해 모든 도시가 연결되도록 만들기 위해 필요한 도로 건설 비용의 최솟값을 구하는 프로그램을 작성해 보자.

입력 형식

첫 번째 줄에 도시의 개수와 이미 건설되어 있는 도로의 개수를 나타내는 N, M이 공백으로 구분되어 주어진다.

두 번째 줄에는 각 도시에 거주하는 주민의 수를 나타내는 $S_1, S_2, \cdots, S_N$이 공백으로 구분되어 주어진다.

세 번째 줄부터 $M+2$번째 줄까지, $i+2$번째 줄에 이미 건설되어 있는 i번째 도로가 연결하는 두 도시의 번호 u_i, v_i가 공백으로 구분되어 주어진다.

출력 형식

모든 도시를 연결하기 위해 도로를 추가로 건설할 때, 도로를 건설하는 비용의 최솟값을 출력한다.

제한

- $1 \le N \le 10^5$
- $0 \le M \le 10^5$
- $1 \le S_i \le 100$
- $1 \le u_i, v_i \le N$
- 모든 (u_i, v_i)쌍은 서로 다르며, $u_i \neq v_i$를 만족한다.
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고

첫 번째 예시는 첫 번째 도시 또는 두 번째 도시를 세 번째 도시와 연결하면 $2 \times 3 = 6$ 만큼의 비용으로 연결할 수 있다.

두 번째 예시는 이미 모든 도시가 서로 연결되어 있으므로 도로를 건설할 필요가 없다.

G. 수열 만들기 (BOJ 28608)

정수 N이 주어지면, $A_1 + A_2 + \cdots + A_x = N$이고 $B_1 \times B_2 \times \cdots \times B_y = N$이면서, $A_1 \times A_2 \times \cdots A_x = B_1 + B_2 + \cdots + B_y$를 만족하는 양의 정수로 구성된 두 수열 A, B를 구하는 프로그램을 작성하라. 단, A는 두 개 이상의 원소로 구성된 수열이어야 하며, B의 모든 원소는 서로 달라야 한다.

입력 형식

첫 번째 줄에 정수 N이 주어진다.

출력 형식

조건을 만족하는 두 수열 A, B가 존재하지 않으면 `-1 -1` 을 출력한다.

그렇지 않은 경우, 첫 번째 줄에 수열 A, B의 길이를 나타내는 두 정수 x, y를 출력한다. 두 번째 줄에는 A의 원소, 세 번째 줄에는 B의 원소를 공백으로 구분해서 차례대로 출력한다.

만약 가능한 정답이 여러 가지이면 아무거나 출력해도 된다.

제한

- $1 \le N \le 10^5$
- $x \ge 2$
- $y \ge 1$
- $1 \le A_i, B_i \le N$
- $i \ne j$ 이면 $B_i \ne B_j$
- $\sum_{i=1}^{x} A_i = \prod_{j=1}^{y} B_j = N$
- $\prod_{i=1}^{x} A_i = \sum_{j=1}^{y} B_j$
- 입력으로 주어지는 수는 모두 정수이다.
- 출력하는 수는 모두 정수여야 한다.

예제

예제는 채점 사이트 참고

H. 블록 게임 (BOJ 28753)

두 사람이 일렬로 놓인 N개의 블록을 이용해 턴을 주고받으며 게임을 한다.

각 사람은 자신의 턴에 앞에서부터 몇 개의 블록을 파괴한 다음, 파괴되지 않은 가장 앞에 있는 블록을 가져간다. 각 사람은 (자신이 가져간 블록에 적힌 수의 합) - (상대방이 가져간 블록에 적힌 수의 합)을 최대화하려고 한다.

두 사람이 모두 최선을 다해 게임을 했을 때, 블록을 먼저 가져가는 사람이 얻는 점수를 구하는 프로그램을 작성해 보자.

입력 형식

첫 번째 줄에 블록의 개수 N이 주어진다.

두 번째 줄에 N개의 블록에 적힌 수 $A_1, A_2, \cdots, A_N$이 앞에 있는 블록부터 차례대로 공백으로 구분되어 주어진다.

출력 형식

두 사람이 모두 최선을 다해 게임을 했을 때, 턴을 먼저 갖는 사람이 얻게 되는 점수를 출력한다.

제한

- $1 \le N \le 200\,000$
- $1 \le A_i \le 10^9$
- 입력으로 주어지는 수는 모두 정수이다.

예제

예제는 채점 사이트 참고