

2025 SCON

숭실대학교 프로그래밍 대회

문제지

Official Problemset

언어 가이드

- 채점은 Intel Xeon E5-2666v3 프로세서를 사용하는 AWS EC2 c4.large 인스턴스에서 진행합니다.
- 채점 서버의 운영체제는 Ubuntu 16.04.7 LTS입니다.
- 아래 언어 중 원하는 언어를 선택해 사용할 수 있습니다.
 - C11: gcc 11.1.0
 - C++17: g++ 11.1.0
 - Java 15: OpenJDK version "16.0.1" 2021-04-20
 - Python 3: Python 3.13.1
 - PyPy3: Python 3.10.14, PyPy 7.3.12 with GCC 10.2.1 20210130
 - 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 scanf와 printf를 사용하는 경우, 입출력 속도는 문제를 해결할 수 있을 정도로 충분히 빠릅니다.
- C++17에서 cin과 cout을 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 BufferedReader와 BufferedWriter를 사용하여야 합니다.
- Python 3 및 PyPy3에서는 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(stdandard input)을 통해 문제의 개수 T를 입력받은 다음 T줄에 걸쳐 공백으로 구분된 두 정수를 입력으로 받아 표준 출력(stdandard output)을 통해 두 정수의 합을 총 T줄에 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)
```

대회 중 유의 사항

- 2025 숭실대학교 프로그래밍 대회(2025 SCON) 유의 사항입니다.
- 대회 정보
 - 이 대회는 숭실대학교 IT대학이 주최, 컴퓨터학부 문제해결 소모임 SCCC가 주관합니다.
 - 이 대회는 스타트링크와 채널코퍼레이션, 한빛미디어의 후원을 받아 진행됩니다.
- 사진 촬영 안내
 - 대회 당일 현장 스태프가 대회 현장을 촬영하여 사진으로 기록 및 온라인 게시할 예정입니다.
 - 사진은 추후 숭실대학교, 숭실대학교 IT대학, IT대학 소속 학부, SCCC의 홍보와 SCCC가 주관하는 대회 홍보에 사용될 수 있습니다.
- 대회 진행 관련
 - 대회는 3시간 동안 10문제로 진행됩니다.
 - 6문제 이상 해결한 팀은 스코어보드 갱신이 중단됩니다.
 - 대회 종료 1시간 전부터 모든 팀의 스코어보드 갱신이 중단됩니다.
 - 대회 도중에는 대회 스태프와 팀원을 제외한 타인과 대화할 수 없습니다.
 - 대회 사이트 및 언어 공식 레퍼런스 사이트를 제외한 모든 인터넷의 사용은 금지됩니다.
 - * BOJ Help : <https://help.acmicpc.net/language/info>
 - * C/C++ : <https://en.cppreference.com/w/>
 - * Java : <https://docs.oracle.com/en/java/javase/15/docs/api/>
 - * Python : <https://docs.python.org/3/>
 - 화장실 이용 시 스태프와 동행해야 하며, 한 번에 한 명씩 이용 가능합니다.
 - 대회 도중 휴대전화 사용이 불가능합니다. 불가피한 경우 스태프의 감독하에 사용 가능합니다.
 - 문제와 관련된 질문은 대회 페이지의 ‘질문’ 탭을 이용해야 합니다. 현장 스태프는 문제에 대한 질문을 받지 않습니다.
 - 대회 공지는 대회 페이지의 ‘공지’ 탭을 이용해 전달합니다. 주기적으로 확인해 주시길 바랍니다.
- 문제 관련
 - 문제는 운영진들이 생각하는 난이도순으로 정렬되어 있지만, 모든 문제를 읽고 고민하는 것을 권장합니다.
 - 모든 문제는 C++, Java, PyPy3으로 해결할 수 있음이 보장됩니다. (Python3는 보장하지 않음)
 - 모든 문제의 메모리 제한은 1024MB로 동일합니다.
 - 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.
 - 언어별 추가 시간과 추가 메모리가 주어지지 않습니다.
 - 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.

- 팀의 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
 - 컴파일 에러는 패널티에 포함되지 않습니다.
- 문제 관련 질문에 대한 답변은 다음 중 하나로 주어집니다.
 - 문제를 잘 읽어주시길 바랍니다 / **Read the problem statement**
질문에 대한 답변이 문제 지문에 있다는 의미입니다.
 - 답변할 수 없습니다 / **No comments**
질문이 잘못되었거나 답변으로 인해 대회 공정성이 해쳐지는 등 답변하기 힘들다는 의미입니다.
 - 전체 공지사항을 참고하시길 바랍니다 / **Please refer to the announcement**
답변 대신 전체 공지사항으로 공지한다는 의미입니다.
 - 네 / **Yes**
 - 아니오 / **No**
 - 기타 답변

간식 안내

- 제공되는 간식에 알레르기 유발 물질이 포함되어 있으니 주의해서 섭취하시길 바랍니다.
- 영양성분표 및 원재료명 관련 문의는 대회 페이지의 질문 기능을 통해 질문하면 답변드리겠습니다.
- 오예스
 - 우유, 밀, 계란, 대두 함유
 - 땅콩을 사용한 제품과 같은 시설에서 제조
- 쿠크다스 케이크
 - 밀, 우유, 대두, 계란, 돼지고기 함유
- 트워스 미니
 - 밀, 우유, 대두 함유
 - 땅콩을 사용한 제품과 같은 시설에서 제조
- 비초비
 - 밀, 우유, 대두, 쇠고기 함유
 - 달걀, 땅콩, 호두, 복숭아, 토마토, 돼지고기, 닭고기, 오징어, 새우, 게, 조개류(굴, 홍합) 혼입 가능
- 허쉬 쿠앤크 미니
 - 밀, 우유, 대두 함유
- 오리온 카스타드
 - 밀, 계란, 대두, 우유 함유
 - 땅콩, 호두, 복숭아, 토마토, 돼지고기, 쇠고기, 닭고기, 오징어, 새우, 게, 조개류(굴, 홍합) 혼입 가능

2025 Soongsil Programming Contest

Problem List

#	Problem Name	Time limit	Memory limit	Page
A	알파벳 블록	1 second	1024MiB	8 – 8
B	합의 최소	2 seconds	1024MiB	9 – 9
C	특별상 눈치게임	1 second	1024MiB	10 – 11
D	N 거리 건너기	1 second	1024MiB	12 – 13
E	even하게 익은 scon	1 second	1024MiB	14 – 14
F	$A = B \oplus C$	1 second	1024MiB	15 – 15
G	불꽃놀이의 아름다움 2	1 second	1024MiB	16 – 17
H	서로소 조합	1 second	1024MiB	18 – 18
I	대결	1 second	1024MiB	19 – 20
J	맛있는 스콘 만들기	3 seconds	1024MiB	21 – 22

문제지에 있는 문제가 총 10문제가 맞는지 확인하시길 바랍니다.

문제는 출제진이 생각하는 난이도순으로 정렬되어 있지만, 모든 문제를 읽고 고민하는 것을 권장합니다.

모든 문제는 C++17, Java 15, PyPy3으로 풀 수 있음을 보장합니다. (단, Python 3는 보장하지 않음)

A. 알파벳 블록

S, C, O, N 4종류의 알파벳 블록을 가지고 놀던 도현이는 두 가지 사실을 알아냈다.

- 알파벳 O 블록 1개를 반으로 자르면 알파벳 C 블록 2개로 사용할 수 있다. 알파벳 C 블록 2개를 합쳐서 알파벳 O 블록 1개로 사용하는 것도 가능하다.
- 알파벳 S 블록 1개를 뒤집으면 알파벳 N 블록 1개로 사용할 수 있다. 알파벳 N 블록 1개를 뒤집어서 알파벳 S 블록 1개로 사용하는 것도 가능하다.

도현이는 교내 대회 홍보를 위해 문자열 “SCON”과 “SCCC”가 하나씩 들어있는 상자를 사람들에게 나눠주기로 했다. 도현이는 가지고 있던 블록들로 문자열을 만들어 최대한 많은 사람들에게 상자를 나눠주려 한다. 각 알파벳 블록의 개수가 주어졌을 때, 도현이가 나눠줄 수 있는 상자 개수의 최댓값을 구해보자.

입력 형식

첫째 줄에 S, C, O, N 알파벳 블록의 개수를 의미하는 b_S, b_C, b_O, b_N 이 공백으로 구분되어 주어진다.

출력 형식

도현이가 나눠줄 수 있는 상자 개수의 최댓값을 출력한다.

제한

- $0 \leq b_S, b_C, b_O, b_N \leq 1\,000\,000$
- 입력으로 주어지는 수는 모두 정수이다.

예제

표준 입력(stdin)	표준 출력(stdout)
2 4 1 1	1
1 2 2 2	1
20 24 20 25	10

B. 합의 최소

길이가 N 인 정수로 구성된 수열 A_1, A_2, \dots, A_N 이 주어진다.

당신은 아래 연산을 0번 이상 사용하여 수열의 모든 원소들의 합 $\sum_{i=1}^N A_i$ 를 최소화하려고 한다.

- 1 이상 $N - 1$ 이하인 정수 i 를 선택한 뒤, A_i 의 값을 A_{i+1} 로 변경한다.

만들 수 있는 수열의 합의 최솟값을 구해보자.

입력 형식

첫째 줄에 수열의 길이 N 이 주어진다.

둘째 줄에 수열의 원소 A_1, A_2, \dots, A_N 이 공백으로 구분되어 주어진다.

출력 형식

수열의 합의 최솟값을 출력한다.

제한

- $1 \leq N \leq 500\,000$
- $1 \leq A_i \leq 1\,000\,000$ ($1 \leq i \leq N$)
- 입력으로 주어지는 수는 모두 정수이다.

예제

표준 입력(stdin)	표준 출력(stdout)
5 5 4 3 2 1	5
5 4 2 7 3 6	16

참고

정답이 32비트 정수 범위를 넘을 수 있으므로, C/C++에서는 `long long`, Java에서는 `long`과 같은 자료형을 사용하는 것을 권장한다.

C. 특별상 눈치게임

당신의 팀을 포함해 총 $N + 1$ 개의 팀이 특별상을 받기 위해 눈치게임을 시작했다! 눈치게임의 규칙은 다음과 같다.

1. 각 팀은 1 이상 100 이하의 서로 다른 정수를 3개 선택해서 제출한다.
2. 두 팀 이상이 선택한 수를 모두 제거한다. 이때 제출된 수들이 모두 제거되면 특별상은 사회에 환원된다.
3. 100에 가장 가까운 수를 선택한 팀이 특별상의 주인공이 된다.

예를 들어, A팀이 (1, 50, 100), B팀이 (2, 99, 100), C팀이 (3, 97, 98), D팀이 (2, 4, 50)을 선택했다고 하자. 두 팀 이상이 선택한 2, 50, 100은 추첨에서 제외되므로, 남은 수 중 100과 가장 가까운 수인 99를 선택한 B팀이 특별상의 주인공이 된다.

당신은 은밀한 방법으로 다른 N 개 팀이 선택한 수들을 전부 파악했다! 이제 남은 일은 제출할 3개의 정수를 최선을 다해 고르는 것뿐이다. 특별상을 노리기 위해, 가능한 조합들을 치밀하게 계산해보려 한다. 당신의 팀이 선택할 수 있는 정수 조합 중 특별상을 받을 수 있는 경우의 수를 구해보자. 단, 순서만 다른 조합은 같은 경우로 센다.

입력 형식

첫째 줄에 자신의 팀을 제외한 참가 팀의 수 N 이 주어진다.

둘째 줄부터 N 개의 줄에 걸쳐 다른 팀에서 고른 3개의 수 A_i, B_i, C_i 가 공백으로 구분되어 주어진다.

출력 형식

당신의 팀이 선택할 수 있는 정수 조합 중 특별상을 받을 수 있는 경우의 수를 구해보자.

제한

- $1 \leq N \leq 100$
- $1 \leq A_i, B_i, C_i \leq 100$ ($1 \leq i \leq N$)
- $A_i \neq B_i; B_i \neq C_i; C_i \neq A_i$ ($1 \leq i \leq N$)
- 입력으로 주어지는 수는 모두 정수이다.

예제

표준 입력(stdin)	표준 출력(stdout)
1 1 99 100	97
1 100 99 98	0
3 37 99 73 17 76 100 87 96 51	2

참고

첫 번째 예시에서 당신의 팀이 특별상을 받기 위해서는 99와 100을 선택해 제거한 뒤, 2 이상 98 이하의 정수를 추가로 선택하면 된다.

두 번째 예시에서 당신의 팀이 특별상을 받을 수 있는 경우의 수는 없지만, 특별상의 행방은 당신에게 달려있다.

D. N거리 건너기

한결이는 오늘도 학교에 간다. 왜냐하면 한결이는 졸업과 동시에 다시 입학하기 때문이다.

한결이의 친구 창민이는 횡단보도가 있을 때 하나의 횡단보도를 삼거리, 사거리, 더 나아가 “N거리”로 만들 수 있는 마법을 할 수 있다. 한결이를 놀리고 싶은 창민이는 한결이가 등교할 때 항상 지나가는 횡단보도를 골라 “N거리”로 바꿔버렸다.

“N거리”는 정 N 각형 모양의 교차로로, 시계 방향으로 각각 $1, 2, \dots, N$ 번까지의 번호가 붙은 N 개의 인도와, 아래와 같이 두 인도를 연결하는 N 개의 횡단보도로 구성되어 있다.

- $i(1 \leq i \leq N - 1)$ 번 횡단보도는 i 번 인도와 $i + 1$ 번 인도를 연결한다.
- N 번 횡단보도는 N 번 인도와 1번 인도를 연결한다.

횡단보도에는 신호등이 하나씩 있다. 신호등은 아래 규칙에 따라 돌아가면서 1초씩 초록불이 켜진다.

- 처음에는 A_1 번 횡단보도의 신호등의 초록불이 켜진다.
- A_i 번 횡단보도의 신호등의 초록불이 꺼짐과 동시에 A_{i+1} 번 횡단보도의 신호등의 초록불이 켜진다.
- 단, $i = N$ 일 때는 A_N 번 횡단보도의 신호등의 초록불이 꺼짐과 동시에 A_1 번 횡단보도의 신호등의 초록불이 켜진다.

한결이는 등교할 때마다 1번 인도에서 출발하여 횡단보도를 건너 M 번 인도로 가야 한다. 한결이는 걸음이 매우 빨라서 인도와 횡단보도를 걷는 시간을 무시할 수 있다고 할 때, 시계 방향과 반시계 방향 중 어느 방향으로 이동해야 M 번 인도에 더 빨리 도착할 수 있는지 구해보자.

입력 형식

첫째 줄에 횡단보도의 개수를 의미하는 N , 한결이가 가야 하는 인도의 번호 M 이 공백으로 구분되어 주어진다.

둘째 줄에 N 거리 신호등의 초록불이 켜지는 순서를 의미하는 N 개의 정수 A_1, A_2, \dots, A_N 이 공백으로 구분되어 주어진다.

출력 형식

한결이가 1번 인도에서 M 번 인도로 가기 위해 반시계 방향을 선택하는 것이 더 빠르다면 ‘CCW’, 시계 방향을 선택하는 것이 더 빠르다면 ‘CW’, 두 방향의 소요 시간이 같다면 ‘EQ’를 따옴표를 제외하고 출력한다.

제한

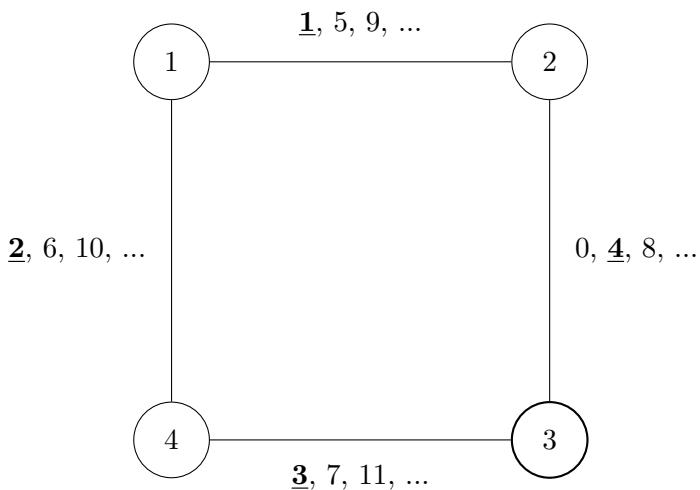
- $3 \leq N \leq 100\,000$
- $2 \leq M \leq N$
- $1 \leq A_i \leq N(1 \leq i \leq N)$
- 수열 A 의 원소는 서로 다르다. 즉, $i \neq j$ 이면 $A_i \neq A_j$ 이다.
- 입력으로 주어지는 수는 모두 정수이다.

예제

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

참고

총 소요 시간이 32비트 정수 범위를 넘을 수 있으므로, C/C++에서는 `long long`, Java에서는 `long`과 같은 자료형을 사용하는 것을 권장한다.



두 번째 예시에서 시계 방향으로 이동하면 1번 횡단보도와 2번 횡단보도를 각각 1초, 4초 뒤에 건널 수 있고, 반시계 방향으로 이동하면 4번 횡단보도와 3번 횡단보도를 각각 2초, 3초 뒤에 건널 수 있다. 따라서 반시계 방향으로 이동하는 것이 3번 인도에 더 빠르게 도착할 수 있다.

E. even하게 익은 SCON

SCON과 SCCC를 바라보던 문성이는 흥미로운 규칙을 발견했다! 바로 두 문자열에서 문자 ‘S’와 ‘C’의 개수를 합하면 항상 짝수가 된다는 것이다. 이를 본 문성이는 문득 알파벳 대문자로 구성된 길이가 N 인 문자열 중에서 ‘S’와 ‘C’의 개수의 합이 짝수인 문자열이 몇 개나 될지 궁금해졌다.

문성이의 궁금증을 해결해 주자!

입력 형식

첫째 줄에 문자열의 길이를 의미하는 정수 N 이 주어진다.

출력 형식

알파벳 대문자로 구성된 길이가 N 인 문자열 중 ‘S’, ‘C’의 개수의 합이 짝수인 문자열의 개수를 $1\ 000\ 000\ 007 (= 10^9 + 7)$ 로 나눈 나머지를 출력하라.

제한

- $1 \leq N \leq 1\ 000\ 000$
- 입력으로 주어지는 수는 모두 정수이다.

예제

표준 입력(stdin)	표준 출력(stdout)
1	24
2	580
1000000	591820787

F. $\mathbf{A} = \mathbf{B} \oplus \mathbf{C}$

X 개의 1과 Y 개의 0을 사용해 길이가 $X + Y$ (단, $X + Y$ 는 3의 배수)인 수열을 만들려고 한다. 아래 조건을 만족하도록 길이가 $X + Y$ 인 수열 $A = \{A_1, A_2, \dots, A_{X+Y}\}$ 를 구성하는 경우의 수를 구해보자.

- $1 \leq k \leq (X + Y)/3$ 인 모든 정수 k 에 대해 $A_{3k-2} = A_{3k-1} \oplus A_{3k}$
- 즉, $A_1 = A_2 \oplus A_3, A_4 = A_5 \oplus A_6, \dots, A_{X+Y-2} = A_{X+Y-1} \oplus A_{X+Y}$

\oplus 는 배타적 논리합(XOR) 연산자이다. 즉, 두 피연산자의 값이 다르면 연산의 결과는 1, 같으면 0이다.

입력 형식

첫째 줄에 정수 X, Y 가 공백으로 구분되어 주어진다.

출력 형식

수열을 구성하는 경우의 수를 출력한다. 단, 답이 매우 커질 수 있으므로 $1\ 000\ 000\ 007 (= 10^9 + 7)$ 로 나눈 나머지를 출력한다.

제한

- $0 \leq X, Y \leq 3\,000$
- $X + Y$ 는 3의 배수이다.
- $X + Y \geq 3$
- 입력으로 주어지는 수는 모두 정수이다.

예제

표준 입력(stdin)	표준 출력(stdout)
4 5	27
1 2	0
3000 3000	292387267

G. 불꽃놀이의 아름다움 2

봄 축제 때 정보과학관에서는 아무 행사도 진행되지 않는다는 것에 화가 난 민성이는 정보과학관 입구 앞에서 직접 불꽃놀이 행사를 진행하려고 한다.

민성이는 정보과학관 입구 앞에 N 개의 폭죽과 서로 다른 두 폭죽을 직접 연결하는 도화선 N 개를 설치했다. N 개의 폭죽은 도화선을 통해 모두 직간접적으로 연결되어 있다. 즉, 어떤 한 폭죽에 불을 붙이면 도화선을 따라 모든 N 개의 폭죽에 불이 붙는다.

민성이는 도화선으로 직접 연결된 두 폭죽의 색이 다를 때 불꽃놀이가 아름답다고 생각한다. 하지만 불꽃놀이의 색의 종류를 늘리는 것은 비용이 많이 들기 때문에 최소 개수의 색을 사용하려고 한다. 아름다운 불꽃놀이를 만들기 위해 필요한 색의 개수의 최솟값을 구해보자.

입력 형식

첫째 줄에 폭죽의 개수 N 이 주어진다.

둘째 줄부터 N 개의 줄에 걸쳐, i 번째 도화선이 연결하는 두 폭죽의 번호 a_i, b_i 가 공백으로 구분되어 주어진다.

출력 형식

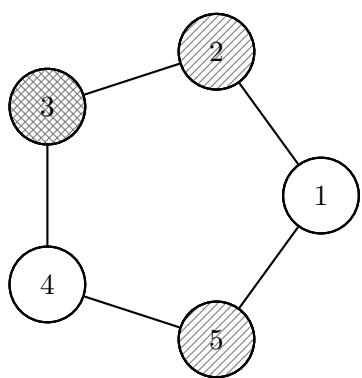
아름다운 불꽃놀이를 만들기 위해 필요한 폭죽 색 종류의 최솟값을 출력한다.

제한

- $3 \leq N \leq 200\,000$
- $1 \leq a_i, b_i \leq N$ ($1 \leq i \leq N$)
- N 개의 폭죽은 도화선을 통해 모두 서로 연결되어 있다.
- 도화선은 서로 다른 두 폭죽을 연결하며, 다른 도화선이 같은 쌍을 연결하는 경우는 없다.
- 입력으로 주어지는 수는 모두 정수이다.

예제

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

참고

첫 번째 예시는 위 그림과 같이 3개의 색을 이용하면 아름다운 불꽃놀이를 만들 수 있고, 이보다 더 적은 색으로는 만들 수 없다.

H. 서로소 조합

서로 다른 n 개의 원소 중 순서를 구분하지 않고 r 개를 선택하는 경우의 수 $C(n, r)$ 은 $C(n, r) = \frac{n!}{r!(n - r)!}$ 으로 구할 수 있다.

$C(n_1, r_1)$ 과 $C(n_2, r_2)$ 이 주어졌을 때, 두 값이 서로소인지 구하는 프로그램을 작성해 보자.

입력 형식

첫째 줄에 테스트 케이스의 수 T 가 주어진다.

이후 T 줄에 걸쳐 네 개의 정수 n_1, r_1, n_2, r_2 가 공백으로 구분되어 주어진다.

출력 형식

각 테스트 케이스마다 한 줄에 하나씩 $C(n_1, r_1)$ 과 $C(n_2, r_2)$ 가 서로소라면 1, 아니면 0을 출력한다.

제한

- $1 \leq T \leq 5\,000$
- $0 < n_1, n_2 \leq 5\,000$
- $0 \leq r_1 \leq n_1$
- $0 \leq r_2 \leq n_2$
- 입력으로 주어지는 수는 모두 정수이다.

예제

표준 입력(stdin)	표준 출력(stdout)
5	0
2 1 2 1	1
3 1 5 1	0
5 2 6 3	1
7 3 4 2	0
2024 511 2025 517	

I. 대결

N 명의 선수가 0번 팀과 1번 팀으로 나누어 대결을 한다. $i(1 \leq i \leq N)$ 번 선수는 A_i 번 팀에 속해있으며, 능력치는 B_i 이다. (단, A_i 는 0 또는 1)

대결의 진행자인 당신은 대결을 여러 라운드에 걸쳐 진행하려고 한다. 당신은 1개 이상 N 개 이하의 라운드를 자유롭게 만든 뒤, 아래 규칙에 따라 라운드에 참가할 선수를 선택해야 한다.

- 각 라운드에는 연속된 번호의 선수들만 참가할 수 있다.
- 모든 선수는 정확히 하나의 라운드에 참가해야 한다.
- 모든 라운드는 한 명 이상의 선수가 참가해야 한다.

예를 들어, $(1, 2), (3), (4, 5)$ 와 같이 라운드를 구성할 수 있지만, $(1, 3), (2, 4, 5)$ 와 같이 불연속적인 번호의 선수들로 라운드를 구성하거나, $(1, 2, 3), (3, 4)$ 와 같이 한 선수가 0개 또는 2개 이상의 라운드에 참가하도록 라운드를 구성할 수 없다.

대결에서 각 팀의 점수는 다음과 같이 계산한다.

- 참가하는 선수의 번호가 작은 라운드부터 차례대로 $1, 2, \dots$ 번째 라운드라고 하자.
- j 번째 라운드의 점수는 해당 라운드에 참가한 팀 소속 선수들의 능력치의 합에 j 를 곱한 값이다.
- 팀의 최종 점수는 해당 팀이 모든 라운드에서 얻은 점수의 총합이다.

예를 들어, $A = [1, 1, 0, 0, 1]$, $B = [1, 2, 3, 4, 5]$ 이고, 라운드를 $(1, 2), (3), (4, 5)$ 로 구성했다고 하자. 0번 팀은 2 라운드에서 $3 \times 2 = 6$ 점, 3 라운드에서 $4 \times 3 = 12$ 점을 획득해서 최종 점수는 18점이고, 1번 팀은 1 라운드에서 $(1+2) \times 1 = 3$, 3 라운드에서 $5 \times 3 = 15$ 점을 획득해 최종 점수는 18점이다.

1번 팀의 열렬한 팬인 당신은 1번 팀의 최종 점수에서 0번 팀의 최종 점수를 뺀 값 S 를 최대화하려고 한다. 만약 S 를 최대화하는 라운드 구성 방법이 여러가지라면, 라운드 개수 M 이 최소가 되도록 라운드를 구성해야 한다.

만들 수 있는 최대 점수 차이 S 와, 이를 달성하기 위한 가장 적은 라운드의 개수 M 을 구하는 프로그램을 작성해 보자.

입력 형식

첫째 줄에 선수의 수를 의미하는 정수 N 이 주어진다.

둘째 줄에 각 선수의 소속 팀을 나타내는 N 개의 정수 A_1, A_2, \dots, A_N 이 공백으로 구분되어 주어진다.

셋째 줄에 각 선수의 능력치를 나타내는 N 개의 정수 B_1, B_2, \dots, B_N 이 공백으로 구분되어 주어진다.

출력 형식

1번 팀의 최종 점수에서 0번 팀의 최종 점수를 뺀 값이 최대가 되도록 하는 최소 라운드 개수 M 과 그때의 최대 점수 차 S 를 공백으로 구분해서 출력한다.

제한

- $1 \leq N \leq 1\,000\,000$
- $0 \leq A_i \leq 1$ ($1 \leq i \leq N$)
- $0 \leq B_i \leq 1\,000\,000$ ($1 \leq i \leq N$)
- 입력으로 주어지는 수는 모두 정수이다.

예제

표준 입력(stdin)	표준 출력(stdout)
5 0 0 0 0 0 1 2 3 4 5	1 -15
5 1 1 0 0 1 1 2 3 4 5	3 7

참고

정답이 32비트 정수 범위를 넘을 수 있으므로, C/C++에서는 long long, Java에서는 long과 같은 자료형을 사용하는 것을 권장한다.

J. 맛있는 스콘 만들기

도현이는 SCON 참가자들에게 나눠줄 간식으로 스콘을 만들기로 했다.

매 시각마다 스콘의 맛이 가장 맛있게 구워지는 최적의 온도가 존재한다는 사실을 알고 있는 도현이는 시각 0부터 시작하여 시각 $N - 1$ 까지 오븐의 온도를 총 N 번 조절하여 스콘을 굽기로 했다.

스콘을 만드는 데 사용할 오븐은 1부터 M 까지의 정수 온도로 조절이 가능하다. 각 시각마다 한 번만 오븐의 온도를 조절할 수 있는데, 시각 0에서는 자유롭게 오븐의 온도를 조절할 수 있지만, 다른 시각에서는 기존 온도에서 C 의 정수 배만큼만 온도를 높이거나 낮출 수 있다. 또, 한 번에 D 를 초과해서 온도를 높이거나 낮출 수 없다. 예를 들어, 시각 t 에서 오븐의 온도를 182로 조절했고, $C = 5, D = 10$ 일 경우, 시각 $t + 1$ 에서 조절 가능한 온도는 172, 177, 182, 187, 192뿐이다.

시각 t 에서의 최적의 온도를 b_t , 시각 t 에서 조절한 오븐의 온도를 k_t 라고 하면 시각 $t + 1$ 에서 스콘의 맛은 $M - |b_t - k_t|$ 만큼 증가한다.

시각 0부터 시각 $N - 1$ 까지의 최적의 온도가 주어졌을 때, 시각 N 에 완성되는 스콘의 맛의 최댓값을 구해보자. 처음 스콘을 오븐에 넣었을 때의 시각은 0이며, 스콘의 맛은 0이다. 모든 시각은 정수 시각만 고려한다.

입력 형식

첫째 줄에 네 정수 N, M, C, D 가 공백으로 구분되어 주어진다.

둘째 줄에 각 시각의 최적의 온도를 의미하는 N 개의 정수 b_0, b_1, \dots, b_{N-1} 이 주어진다.

출력 형식

시각 N 에 완성되는 스콘의 맛의 최댓값을 출력한다.

제한

- $1 \leq N \leq 200$
- $1 \leq M \leq 25\,000$
- $1 \leq C \leq D \leq M$
- D 는 C 의 배수이다.
- $1 \leq b_i \leq M$ ($0 \leq i \leq N - 1$)
- 입력으로 주어지는 수는 모두 정수이다.

예제

표준 입력(stdin)	표준 출력(stdout)
3 8 2 4 3 7 1	22
3 8 2 4 8 3 2	23

참고

첫 번째 예시에서 오븐의 온도를 차례대로 3, 7, 3으로 조절했을 경우, 스콘의 맛은 $8 + 8 + 6 = 22$ 가 되며, 이보다 더 맛있게 스콘을 만드는 방법은 존재하지 않음을 증명할 수 있다.