

Sogang Programming Contest

2021

Host

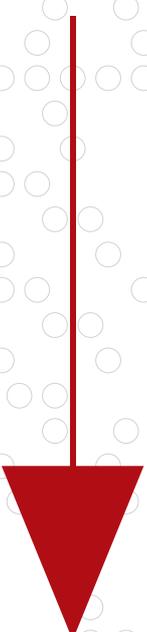


2021 제17회
서강대학교
프로그래밍 대회

Division

Champion

문제지



Champion
Official Problemset

Sponsors



대회 규칙

- 대회 중 제출한 소스코드는 채점 서버에 의해 자동으로 채점되며, 실시간으로 결과를 알 수 있습니다. 문제를 제출하였을 때 ‘맞았습니다!!’ 를 받으면 문제를 푼 것으로, 이외의 결과를 받으면 틀린 것으로 생각합니다.
- 문제를 풀 때마다 패널티 점수가 누적됩니다. 패널티 점수는 모든 맞은 문제에 대해, 대회 시작 시간부터 그 문제를 풀기까지 걸린 시간을 t 분, 처음으로 문제를 맞기 직전까지 제출한 횟수를 w 번이라고 할 때 $(t + 20w)$ 점입니다.
- 순위는 푼 문제가 많은 순서대로, 푼 문제가 같을 경우에는 패널티 점수의 합이 적은 순서대로 결정됩니다.
- 사용 가능 언어는 C, C++, Java, Python 3, Kotlin입니다. 모든 문제는 출제진이 C++로 정답을 작성했음이 보장되며, 문제를 해결하지 못하는 언어가 존재할 수도 있습니다. 각 언어의 컴파일 옵션과 컴파일러 버전은 아래와 같습니다.

C11 gcc (GCC) 11.1.0

```
컴파일 gcc Main.c -o Main -O2 -Wall -lm -static -std=gnu11 -DONLINE_JUDGE -DBOJ
실행 ./Main
```

C++11 g++ (GCC) 11.1.0

```
컴파일 g++ Main.cc -o Main -O2 -Wall -lm -static -std=gnu++11 -DONLINE_JUDGE -DBOJ
실행 ./Main
```

C++14 g++ (GCC) 11.1.0

```
컴파일 g++ Main.cc -o Main -O2 -Wall -lm -static -std=gnu++14 -DONLINE_JUDGE -DBOJ
실행 ./Main
```

C++17 g++ (GCC) 11.1.0

```
컴파일 g++ Main.cc -o Main -O2 -Wall -lm -static -std=gnu++17 -DONLINE_JUDGE -DBOJ
실행 ./Main
```

C++20 g++ (GCC) 11.1.0

```
컴파일 g++ Main.cc -o Main -O2 -Wall -lm -static -std=gnu++20 -DONLINE_JUDGE -DBOJ
실행 ./Main
```

Java 8 openjdk version "16.0.1" 2021-04-20

```
컴파일 javac -release 8 -J-Xms1024m -J-Xmx1920m -J-Xss512m -encoding UTF-8
Main.java
실행 java -Xms1024m -Xmx1920m -Xss512m -Dfile.encoding=UTF-8 -XX:+UseSerialGC
-DONLINE_JUDGE=1 -DBOJ=1 Main
```

Python 3 Python 3.9.5

```
컴파일 python3 -c "import py_compile; py_compile.compile(r'Main.py')"
실행 python3 Main.py
```

PyPy3 PyPy 7.3.4 with GCC 7.3.1 20180303 (Red Hat 7.3.1-5) (Python 3.7.10)

```
컴파일 pypy3 -c "import py_compile; py_compile.compile(r'Main.py')"
실행 pypy3 Main.py
```

Kotlin/JVM kotlinc-jvm 1.5.0 (JRE 1.8.0_201-b09)

컴파일 `kotlinc-jvm -J-Xms1024m -J-Xmx1920m -J-Xss512m -include-runtime -d Main.jar Main.kt`

실행 `java -Xms1024m -Xmx1920m -Xss512m -Dfile.encoding=UTF-8 -XX:+UseSerialGC -DONLINE_JUDGE=1 -DBOJ=1 -jar Main.jar`

- 네트워크 사용은 금지됩니다. 단, 솔루션을 제출하거나 언어 레퍼런스를 확인하는 것은 가능합니다. 접속이 허용된 사이트의 예는 다음과 같습니다.

C/C++ <https://en.cppreference.com/w/>

Java <https://docs.oracle.com/javase/8/docs/api/>

Python <https://docs.python.org/3/>

Kotlin <https://kotlinlang.org/docs/>

- 대회 종료 전에 퇴실할 수 없습니다.
- 책이나 개인이 준비한 인쇄된 참고자료를 열람할 수 있습니다. 참고자료의 양에는 제한이 없습니다.
- 대회 중에 휴대폰 및 전자기기는 사용할 수 없습니다.

문제 목록

문제지에 있는 문제가 총 8문제가 맞는지 확인하시기 바랍니다.

- A 문자열 압축 해제
- B 워드
- C 카드컨트롤
- D 방문 판매
- E Leader-based Team Distribution
- F 던전 릴레이
- G 카드 잘 섞기
- H 연결 요소와 쿼리

모든 문제의 메모리 제한은 1GB로 동일합니다.

문제 A. 문자열 압축 해제

시간 제한 1 초
메모리 제한 1024 MB

특정 소문자 문자열 패턴을 대문자 한 글자로 압축하는 프로그램 SPC(String Pattern Compressor)가 있다. 예를 들어, 다음과 같은 방법으로 압축하는 경우, “aabbaaac”은 “ABAC”로 압축된다.

소문자 문자열 패턴	대문자
aa	A
bba	B
c	C

압축 프로그램과 압축된 문자열이 주어지면, 압축되기 전 문자열의 일부를 출력하는 프로그램을 작성하시오.

입력

첫 번째 줄에 압축 방법의 개수 N 이 주어진다. ($1 \leq N \leq 26$)

두 번째 줄부터 N 개의 줄에 소문자 문자열 패턴과 대응되는 대문자가 공백으로 구분되어 주어진다. 각 소문자 문자열 패턴의 길이는 1000을 넘지 않으며, 같은 대문자는 두 번 이상 주어지지 않는다.

$N + 1$ 번째 줄에 압축된 문자열이 주어진다. 압축된 문자열 길이는 1000을 넘지 않는다.

마지막 줄에 두 정수 S 와 E 가 주어진다. ($1 \leq S \leq E \leq$ (압축되기 전 문자열 길이))

출력

압축되기 전 문자열의 S 번째에서 E 번째 문자까지 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 aa A bba B c C ABAC 4 6	baa

aabbaaac

표준 입력(stdin)	표준 출력(stdout)
5 abcde A abcde B abcde C abcde D abcde E ABCDE 1 25	abcdeabcdeabcdeabcdeabcde

abcdeabcdeabcdeabcdeabcde

표준 입력(stdin)	표준 출력(stdout)
4 e E f F g G h H EEEEEEEE 4 5	fe

eefeee

문제 B. 와드

시간 제한 1 초
메모리 제한 1024 MB

한별이는 출근하던 도중 이세계 대환장 버스에 치였다.



그림 B.1: 이세계 대환장 버스



그림 B.2: 출근하는 한별이

올해 휴가를 전부 써 버려 당장 판교로 돌아가야 하는 한별이는 돌아가기 위한 방법을 어떻게든 찾아보기 위해 이세계를 돌아다녀 보려고 한다.

이세계는 $R \times C$ 의 격자로 되어 있다. 지금은 밤이어서 한별이는 자신이 위치한 칸 및 그 칸에서 위, 아래, 왼쪽 또는 오른쪽으로 인접한 칸만을 볼 수 있지만, 와드를 설치하면 조금 더 넓은 영역의 시야를 확보할 수 있다. 구체적으로는, 격자의 모든 칸은 각각 어떤 영역 하나에 속해 있는데, 와드를 놓으면 와드가 놓인 칸이 속한 영역에 있는 모든 칸을 볼 수 있게 된다.

한별이의 여행 기록이 주어질 때 한별이가 얼마나 넓은 시야를 확보했는지 계산해 보자.

입력

첫 번째 줄에는 격자의 크기를 나타내는 두 정수 R 과 C 가 주어진다. ($1 \leq R, C \leq 1000$)

다음 줄부터 R 개의 줄에 걸쳐 격자의 정보가 주어진다. 각 줄은 C 개의 알파벳 소문자로 이루어져 있으며, 위, 아래, 왼쪽 또는 오른쪽으로 인접해 있는 칸이 같은 문자라는 것은 두 칸이 같은 영역에 속해 있음을 의미한다.

다음 줄에는 한별이가 이세계에 떨어진 위치를 나타내는 두 정수 H_R, H_C 가 주어진다. 이는 한별이가 위에서 H_R 번째 줄, 왼쪽에서 H_C 번째 칸에 떨어졌음을 의미한다. ($1 \leq H_R \leq R, 1 \leq H_C \leq C$)

마지막 줄에는 한별이의 여행 기록을 나타내는 200 000글자 이하의 문자열이 주어진다. 여행 기록의 각 문자는 **U, D, L, R, W** 중 하나로 이루어져 있으며, **U, D, L, R**은 각각 위, 아래, 왼쪽, 오른쪽으로 한 칸 이동했다는 뜻이고, **W**는 지금 있는 칸에 와드를 설치했다는 뜻이다. 한별이가 격자를 벗어나는 경우는 주어지지 않는다.

출력

R 개의 줄에 걸쳐 한별이의 시야를 출력한다. 각 줄은 C 개의 문자로 되어 있어야 하며, R 번째 줄 C 번째 문자가 **.**이라면 한별이의 시야에 해당 칸이 들어와 있다는 뜻이고 **#**이라면 그렇지 않다는 뜻이다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 5 aaabc dcbbc dcaa dada 3 4 WLLLWUURD	##.### .#...
3 3 abc def ghi 2 2 LU	..# .## ###

두 번째 입출력 예시에서는 와드를 설치하지는 않았지만, 한별이의 최종 위치의 위, 아래, 왼쪽, 오른쪽 칸은 시야로 확보하고 있다.

지나온 경로를 모두 시야로 확보하지는 않는다.

문제 C. 카드컨트롤

시간 제한 1 초
메모리 제한 1024 MB

준석이와 수현이는 카드게임을 하고 있다. O가 적힌 카드 N 장, X가 적힌 카드 N 장이 섞여 있다. 게임은 다음 순서로 진행된다.

1. 준석이가 제일 위에 있는 카드를 가져간다.
2. 수현이가 다음 카드를 가져간다.
3. 준석이의 카드와 수현이의 카드가 같으면 무승부, 다른 카드면 O를 가진 사람의 점수가 1점 추가된다.

마지막에 점수가 높은 사람이 이긴다.

준석이는 게임 시작 전에 카드를 조작할 수 있다. 정확히는 한 번의 조작으로 원하는 카드 하나를 빼내서 카드의 제일 위에 올리는 동작을 할 수 있다.

준석이가 초기 배치를 안다고 할 때, 최소한의 조작으로 준석이가 이기려면 몇 번 조작해야 하는가?

입력

첫 번째 줄에 N 이 주어진다. ($1 \leq N \leq 5$)

다음 줄에 위에 있는 카드부터 순서대로 카드에 적혀있는 문양이 주어진다.

출력

카드의 최소 조작 횟수를 출력하라.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2 X O O X	1

이 페이지는 공백입니다

문제 D. 방문 판매

시간 제한 1 초
메모리 제한 1024 MB

SG그룹은 이번에 획기적인 제품 X , Y 를 출시했다. SG그룹의 영업 부서에서 외판원으로 일하는 판매왕 레오는 이 두 제품을 주어진 각 할당량 X , Y 만큼 N 명의 고객의 집을 모두 방문하여 팔아야 한다. 고객마다 1번부터 N 번까지 번호가 주어지고, i 번 고객의 집에 방문하여 판매에 성공했을 때 팔 수 있는 제품 X , Y 의 양이 각각 x_i , y_i 로 주어진다. 그러나 어떤 고객은 방문하더라도 제품 구매를 거절하여 판매에 실패할 수 있다.

방문 판매를 할 때는 영업 부서에서 정한 매뉴얼에 따라 1번 고객부터 N 번 고객까지 오름차순으로 한 번씩 방문해야 한다. 또한, 레오가 판매하면서 가지고 다니는 차에는 제품 X , Y 가 부족할 일이 없도록 충분히 많은 양을 채운다.

레오는 이번 방문 판매로 최소 몇 명의 고객에게 두 제품 X , Y 를 팔아야 주어진 할당량을 채울 수 있는지 궁금해졌다. 방문 순서를 만족하면서 할당량을 채우기 위해 제품을 구매하도록 만들어야 하는 최소 고객의 수와 이때 가장 마지막으로 판매에 성공하는 고객 번호를 구해보자.

입력

첫 번째 줄에 레오가 방문해야 하는 고객 수인 N 과 판매해야 하는 두 제품 X , Y 의 할당량인 X , Y 가 정수로 주어진다. ($1 \leq N \leq 50$, $1 \leq X, Y \leq 100$)

두 번째 줄부터 N 개의 줄에 걸쳐 1번부터 N 번 고객까지 각 고객을 방문하여 판매에 성공했을 때 해당 고객이 구입하는 두 제품 X , Y 의 양인 x_i 와 y_i 가 정수로 주어진다. ($1 \leq x_i, y_i \leq 100$, $1 \leq i \leq N$)

출력

방문 순서를 만족하면서 할당량을 채우기 위해 제품 판매에 성공해야 하는 최소 고객의 수를 출력한다. 이때 마지막으로 판매에 성공하는 고객의 번호를 그 다음 줄에 출력하는데, 가능한 번호가 여러 개이면 그중 가장 작은 값을 출력한다.

어떠한 고객이 제품을 구입한다고 하더라도 할당량을 채울 수 없는 경우에는 -1 을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 8 10	2
5 2	4
3 7	
2 1	
4 8	

두 제품 X , Y 의 할당량이 각각 8, 10이면, 최소 1번 고객과 4번 고객이 제품을 구입해야 할당량을 채울 수 있다.

이때 4번 고객보다 방문 순서가 빠른 고객에서 가장 마지막으로 판매에 성공할 수는 없다.

표준 입력(stdin)	표준 출력(stdout)
5 8 3	2
3 1	2
5 2	
3 1	
1 2	
2 2	

1번 고객과 2번 고객이 제품을 구매하는 경우와 2번 고객과 3번 고객이 제품을 구매하는 경우 모두 할당량을 채울 수 있지만, 3번보다 2번 고객의 번호가 더 작다.

따라서 가장 마지막으로 판매에 성공한 고객으로 2를 출력한다.

표준 입력(stdin)	표준 출력(stdout)
3 11 12	-1
4 9	
2 3	
2 1	

어떠한 고객이 제품을 구입한다고 하더라도 제품 **X**의 할당량인 11을 채울 수 있는 경우는 없다.

문제 E. Leader-based Team Distribution

시간 제한 2 초
메모리 제한 1024 MB

플레이어 N 명이 M 개의 팀으로 나누어 게임을 진행하려 한다. 각 팀의 인원수는 t_1, t_2, \dots, t_M 이며, i 번째 플레이어는 리더 점수 L_i 와 플레이어 점수 P_i 를 가진다. 각 플레이어는 정확히 한 팀에 속해야 한다.

각 팀의 리더는 팀에 속한 플레이어 중 리더 점수가 제일 큰 사람이다. 만약 한 팀에 리더 점수가 가장 큰 플레이어가 여러 명이려면, 한 플레이어만 리더가 된다. 이때 팀의 능력은 리더의 플레이어 점수로 정의된다.

플레이어를 각 팀에 적절히 분배하여 모든 팀의 능력의 합을 최대한 크게 해 보자!

입력

첫째 줄에 N, M 이 주어진다. ($1 \leq M \leq N \leq 3 \cdot 10^5$)

이어지는 줄부터 N 개의 줄에 걸쳐 두 정수 L_i 와 P_i 가 주어진다. ($1 \leq L_i, P_i \leq 10^5$)

마지막 줄에는 M 개의 정수가 주어진다. i 번째 수는 t_i 이다. ($1 \leq t_i \leq N, \sum_{i=1}^M t_i = N$)

출력

팀을 적절히 분배하였을 때 각 팀의 능력의 합의 최댓값을 출력하라.

입출력 예시

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

이 페이지는 공백입니다

문제 F. 던전 릴레이

시간 제한 2 초
메모리 제한 1024 MB

선호는 RPG 게임을 즐긴다. 이 RPG 게임에서는 지불 또는 보상 수단으로 캐시라는 재화를 사용한다.

이 게임에는 1번부터 N 번까지의 던전이 있으며, 각 던전마다 입장하려면 지불해야 할 캐시 금액인 입장료가 필요하고 던전을 깨면 보상으로 받는 캐시 금액이 있다. 또한 각 던전마다 플레이 난이도가 있지만, 서버 랭킹 1위인 선호의 캐릭터는 워낙 강해서 던전에 입장만 가능하면 난이도와 상관없이 입장한 던전을 깰 수 있다.

이 게임에는 던전 릴레이라는 시스템이 있는데, 사냥할 던전의 난이도 범위를 설정하면 이 범위 안에 있는 난이도의 모든 던전을 사냥하는 것이다. 이때, 캐릭터는 난이도 범위에 속하면서 난이도가 가장 낮은 던전부터 가장 높은 던전까지 오름차순으로 던전을 깨야 하고, 난이도가 같은 던전이 있으면 던전 번호의 오름차순으로 던전을 깨야 한다. 즉, 설정한 난이도 범위에 있는 모든 던전을 규칙에 따라 차례대로 깨야 릴레이를 성공적으로 마칠 수 있다. 그런데 릴레이 도중 어떤 한 던전에 입장할 때 가지고 있는 캐시 금액이 입장료보다 적으면 그 던전에 입장이 불가능해서 플레이 중인 릴레이가 중단된다.

던전 릴레이 도중에 깬 던전은 해당 릴레이를 진행하는 동안에는 재입장이 불가능하지만, 이후의 던전 릴레이에서는 입장료만 낼 수 있으면 다시 입장이 가능하여 깰 수 있다. 또한 보상으로 받는 캐시 금액은 각 던전을 깨는 즉시 지급된다.

참가할 던전 릴레이의 난이도 범위가 주어졌을 때 선호가 해당 릴레이를 성공적으로 마칠 수 있을지를 알려주자.

입력

첫 번째 줄에는 던전의 수 N 이 주어진다. ($1 \leq N \leq 10^5$)

두 번째 줄부터 N 개의 줄에 걸쳐 1번부터 N 번까지 차례대로 각 던전의 난이도 d_i , 입장료 캐시 금액 e_i , 던전을 깬 후 받을 수 있는 보상 캐시 금액 f_i 가 정수로 주어진다. ($1 \leq d_i \leq 10^9, 0 \leq e_i, f_i \leq 10^9, 1 \leq i \leq N$)

$N + 2$ 번째 줄에는 쿼리의 수 Q 가 주어진다. ($1 \leq Q \leq 10^5$)

$N + 3$ 번째 줄부터 Q 개의 줄에 걸쳐 쿼리가 한 줄에 하나씩 주어진다. 이때, 쿼리는 아래 형식 중 하나를 따르며, 모든 변수는 정수임이 보장된다.

- **1 c l r**: 선호의 캐릭터가 릴레이 시작 전 c 만큼의 캐시 금액을 가진 상태에서 난이도 범위가 l 이상 r 이하인 던전 릴레이를 성공적으로 마칠 수 있으면 1을, 플레이 도중 중단되거나 주어진 난이도 범위를 만족하는 던전이 존재하지 않으면 -1을 출력한다. 성공적으로 마칠 수 있으면 해당 던전 릴레이를 끝낸 후 선호가 가지고 있을 캐시 금액을 같은 줄에 공백을 두고 출력한다. ($0 \leq c \leq 10^9, 1 \leq l \leq r \leq 10^9$)
- **2 x v**: x 번 던전의 입장료를 v 로 바꾼다. ($1 \leq x \leq N, 0 \leq v \leq 10^9$)
- **3 x v**: x 번 던전의 보상 캐시 금액을 v 로 바꾼다. ($1 \leq x \leq N, 0 \leq v \leq 10^9$)

출력

1로 시작하는 쿼리들에 대해, 정답을 한 줄에 하나씩 순서대로 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5	-1
23 65 84	-1
9 30 42	1 83
11 9 2	
24 10 8	
2 51 14	
5	
1 60 2 15	
2 1 37	
3 3 20	
1 30 13 21	
1 27 10 30	

첫 번째 쿼리에서는 5번, 2번, 3번 던전을 차례로 사냥해야 하는데, 2번 던전을 입장할 때 캐릭터가 가지고 있는 캐시가 부족해서 입장에 불가하여 릴레이가 중단된다.

네 번째 쿼리에서는 설정한 난이도 범위에 속하는 던전이 존재하지 않는다.

다섯 번째 쿼리에서는 3번, 1번, 4번 던전을 차례로 사냥해야 하는데, 릴레이 도중 중단되지 않고 끝까지 마칠 수 있으며 이를 마쳤을 때 캐릭터가 가지고 있는 캐시는 83이다.

문제 G. 카드 잘 섞기

시간 제한 3 초
메모리 제한 1024 MB

준석이는 오늘도 열심히 카드를 섞고 있다.

준석이는 카드 N 장을 가지고 있다. 카드에는 전부 X 가 적혀 있고 위에서 P 번째에 있는 카드 한 장에만 O 가 적혀 있다. 준석이는 다음 2가지 쿼리를 열심히 수행하고 있다.

- $1\ x\ y\ z$: 위에서부터 x 번째 카드부터 y 번째 카드까지 리플셔플한다. 이 동작을 z 번 반복한다.
- $2\ x$: 위에서 x 장을 떼서 나머지 카드의 아래쪽에 둔다. 떼어낸 카드의 순서는 그대로 둔다.

리플셔플은 N 장의 카드가 있을 때 카드를 위에서부터 $\lfloor (N+1)/2 \rfloor$ 장의 카드 덩어리와 나머지 덩어리로 나누고 교대로 카드를 두는 방식으로 카드를 섞는 것이다. 섞기 전에 제일 위에 있었던 카드가 섞은 후에도 제일 위로 오도록 한다.

준석이는 카드를 열심히 섞다가 실수로 카드를 떨어트렸다! 준석이는 카드를 떨어트리기 직전에 O 가 적힌 카드가 어디에 있었는지 알고 싶지만 엄청 게으르기 때문에 당신에게 부탁하기로 했다. 준석이를 위해서 카드가 떨어지기 전 O 가 적힌 카드의 위치를 구해보자.

입력

첫 번째 줄에 카드 수 N , 쿼리 수 Q , 카드를 섞기 전 O 가 적힌 카드의 위치 P 가 주어진다. ($1 \leq N \leq 10^9$, $1 \leq Q \leq 10^6$, $1 \leq P \leq N$)

두 번째 줄부터 Q 개의 쿼리가 문제에서 언급한 형식으로 한 줄에 하나씩 주어진다.

- 1번 쿼리에서 z 는 1 이상 10^9 이하이며, $x \leq y$ 를 만족한다.
- 2번 쿼리에서 x 는 1 이상 N 미만이다.

출력

카드가 떨어지기 전 O 가 적힌 카드가 위에서 몇 번째에 있는지 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
10 2 5 1 3 7 1 2 5	2

노트

카드 6장이 있을 때 쿼리 $1\ 1\ 6\ 1$ 을 처리하면

1 2 3 4 5 6 \Rightarrow 1 4 2 5 3 6

이 되고, 카드 7장이 있을 때 1 1 7 1을 처리하면

1 2 3 4 5 6 7 \Rightarrow 1 5 2 6 3 7 4

가 된다.

문제 H. 연결 요소와 쿼리

시간 제한 5 초
메모리 제한 1024 MB

$K \times N$ 격자 그래프 A 가 주어진다. 격자 그래프는 노드가 직사각형 모양으로 배치되어 있는 그래프이며, 각 노드는 위, 아래, 왼쪽, 그리고 오른쪽 노드와 연결되어 있다.

위에서 x 번째 줄 왼쪽에서 y 번째 칸의 노드를 A_{xy} 로 나타내자. 이 때, 다음 쿼리를 수행하는 프로그램을 작성하시오.

- 1 $x_1 y_1 x_2 y_2$: $x_1 \leq x \leq x_2, y_1 \leq y \leq y_2$ 인 A_{xy} 만으로 이루어진 크기 1 이상의 연결 요소 중, 구성 노드의 값의 합이 가장 큰 연결 요소를 찾아 그 합을 출력한다.
- 2 $x y v$: A_{xy} 의 값을 v 로 설정한다.

입력

첫 번째 줄에 격자 그래프의 크기를 나타내는 정수 K, N 이 주어진다. ($K \in \{1, 2, 3\}, 1 \leq N \leq 100\,000$)

두 번째 줄부터 K 개의 줄에 걸쳐 격자 그래프의 노드의 값들이 주어진다. ($-10^9 \leq A_{ij} \leq 10^9$)

다음 줄에는 쿼리의 개수 Q 가 주어진다. ($1 \leq Q \leq 100\,000$)

다음 줄부터 Q 개의 쿼리가 문제에서 언급한 형식으로 한 줄에 하나씩 주어진다.

- 1번 쿼리에서 $1 \leq x_1 \leq x_2 \leq K, 1 \leq y_1 \leq y_2 \leq N$ 이다.
- 2번 쿼리에서 $1 \leq x \leq K, 1 \leq y \leq N, -10^9 \leq v \leq 10^9$ 이다.

입력으로 들어오는 모든 수는 정수다.

출력

각각의 쿼리마다 정답을 한 줄에 하나씩 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 8	48
-1 5 6 3 -1 7 -6 9	45
6 8 -9 2 -4 5 1 8	8
3 4 1 5 -7 -3 2 7	
4	
1 1 2 3 7	
2 1 5 -100	
1 1 2 3 7	
1 2 2 2 4	

이 페이지는 공백입니다