물통

용량이 다른 두 개의 빈 물통 A, B가 있다. 이 물통들에 물을 채우고 비우는 일을 반복하여 두 물통을 원하는 상태(목표하는 양의 물을 담은 상태)가 되도록 만들고자 한다. 물통 이외에는 물의 양을 정확히 잴 수 있는 방법이 없으며, 가능한 작업은 다음과 같은 세 종류가 전부이다.

[F(x): Fill x]: 물통 x에 물을 가득 채운다. (물을 채우기 전에 물통 x가 비어 있는지 여부는 관계없음. 다른 물통은 그대로 둠)

[E(x): Empty x]: 물통 x의 물을 모두 버린다. (다른 물통은 그대로 둠)

[M(x,y): Move water from x to y]: 물통 x의 물을 물통 y에 붓는다. 이때 만약 물통 x에 남아 있는 물의 양이 물통 y에 남아 있는 빈 공간보다 적거나 같다면 물통 x의 물을 물통 y에 모두 붓는다. 만약 물통 x에 남아 있는 물의 양이 물통 y에 남아 있는 빈 공간보다 많다면 부을 수 있는 만큼 최대로 부어 물통 y를 꽉 채우고 나머지는 물통 x에 남긴다.

예를 들어, 물통 A와 B의 용량이 각각 2리터와 5리터라고 하자. 두 물통 모두 빈 상태에서 시작하여 최종적으로 물통 A에는 2리터, 물통 B에는 4리터 물을 남기길 원할 경우, 다음과 같은 순서로 작업을 수행하면 총 8회의 작업으로 원하는 상태에 도달할 수 있다.

(0,0)→[F(B)]→(0,5)→[M(B,A)]→(2,3)→[E(A)]→(0,3)→[M(B,A)]→(2,1)→[E(A)]→(0,1)→[M(B,A)]→(1,0)→[F(B)]→(1,5)→[M(B,A)]→(2,4)

하지만, 작업 순서를 아래와 같이 한다면 필요한 작업 총 수가 5회가 된다.

(0,0)→[F(A)]→(2,0)→[M(A,B)]→(0,2)→[F(A)]→(2,2)→[M(A,B)]→(0,4)→[F(A)]→(2,4)

두 물통의 용량과 원하는 최종 상태를 입력으로 받은 후, 두 물통이 비어 있는 상태에서 시작하여 최종 상태에 도달하기 위한 최소 작업 수를 구하는 프로그램을 작성하시오.

소스파일의 이름은 bucket.c 또는 bucket.cpp를 권장하지만, 서버에 제출하는 데는 다른 이름도 상관없다.

입력 형식

표준 입력으로 물통 A의 용량을 나타내는 정수 $a(1 \le a < 100{,}000)$, 물통 B의 용량을 나타내는 정수 $b(a < b \le 100{,}000)$, 최종 상태에서 물통 A에 남겨야 하는 물의 용량을 나타내는 정수 $c(0 \le c \le a)$, 최종 상태에서 물통 B에 남겨야 하는 물의 용량을 나타내는 정수 d $(0 \le d \le b)$가 공백으로 분리되어 한 줄에 주어진다.

출력 형식

목표 상태에 도달하는 최소 작업 수를 나타내는 정수를 표준 출력으로 출력한다. 만약 목표 상태에 도달하는 방법이 없다면 -1을 출력한다.

부분문제의 제약 조건

- **부분문제 1:** 전체 점수 100점 중 9점에 해당하며 A 물통의 용량은 1리터이다.
- **부분문제 2:** 전체 점수 100점 중 14점에 해당하며 B 물통의 용량은 A 물통의 용량의 배수이다.
- **부분문제 3:** 전체 점수 100점 중 34점에 해당하며 a,b의 범위를 $1 \leq a,b \leq 1{,}000$으로 제한한다.
- **부분문제 4:** 전체 점수 100점 중 43점에 해당하며 원래의 제약조건 이외에 아무 제약조건이 없다.

입력과 출력의 예

입력(1)

```
3 7 3 2
```

출력(1)

```
9
```

입력(2)

```
2 5 0 1
```

출력(2)

```
5
```

입력(3)

```
3 5 2 4
```

출력(3)

```
-1
```

문명

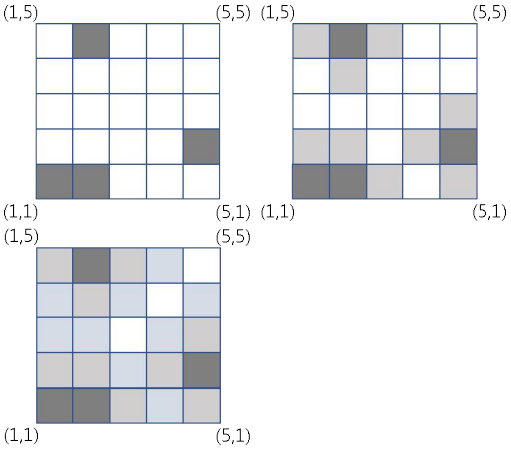
인류의 역사를 돌이켜보면, 문명의 발전은 독자적으로 진행되기도 하지만 서로 다른 문명이 만나 결합되기도 한다. 여러분은 이 가설을 바탕으로, 세계 문명의 발전 과정을 시뮬레이션 해보려고 한다.

세계를 $N \times N$의 2차원 공간으로 생각할 수 있다. 즉, 1×1 크기의 정사각형이 가로, 세로로 각각 N개씩 쌓여있는 형태로 생각할 수 있다. 가장 왼쪽 아래 정사각형은 $(1,1)$, 가장 오른쪽 위 정사각형은 (N,N) 위치에 있다. 두 정사각형 (a,b)와 (a',b')은 다음 두 조건 중 하나만 만족할 때 서로 **인접**해 있다고 하자.

- $|a-a'|=1$ 이고 $b=b'$.
- $|b-b'|=1$ 이고 $a=a'$.

문명의 최초 발상지는 모두 서로 다른 K곳에 있다. 각 정사각형에 해당하는 공간은 문명 지역이거나, 미개 지역이다. 문명의 최초 발상지는 문명 지역이며, 만약 문명 최초 발상지끼리 인접해 있다면, 이들은 처음부터 하나로 결합된다. 한 해가 지날 때마다, 문명 지역은 자신과 인접한 지역에 문명을 전파한다. 정사각형 (a,b)가 문명 지역이면, 다음 해에는 세계의 경계를 넘지 않는 한 이 정사각형과 인접한 네 정사각형 $(a+1,b)$, $(a-1,b)$, $(a,b+1)$, $(a,b-1)$에 문명이 전파된다. 만약 두 인접하는 지역에 다른 문명이 전파되었거나, 한 지역에 둘 이상의 다른 문명이 전파된다면 이 문명들은 결합된다.

예를 들어, 다음과 같이 5×5 크기의 세계에 4곳의 정사각형 $(1,1)$, $(2,1)$, $(2,5)$, $(5,2)$가 문명의 발상지라고 하자. 정사각형 $(1,1)$, $(2,1)$의 문명은 인접해 있으므로 처음부터 결합되어 있다. 1년 후 문명이 전파된 지역은 오른쪽 그림과 같고, 2년 후에 문명이 전파된 지역은 아래 그림과 같다. 이때 모든 문명은 서로 결합되어 하나의 문명이 된다. $(2,5)$에서 발상한 문명과 $(5,2)$에서 발상한 문명은 직접적으로 결합되지는 않았지만, $(1,1)$,$(2,1)$에서 발상한 문명을 통하여 결합됨에 주의하라.



세계의 크기, 문명 발상지의 수 및 위치를 입력으로 받아 모든 문명이 하나로 결합될 때까지 걸리는 최소 햇수를 구하는 프로그램을 작성하시오.

소스파일의 이름은 civilization.c 또는 civilization.cpp를 권장하지만, 서버에 제출하는 데는 다른 이름도 상관없다.

입력 형식

표준 입력으로 다음 정보가 주어진다. 첫 번째 줄에는 세계의 크기를 나타내는 정수 N ($2 \le N \le 2{,}000$)과 문명 발상지의 수 K ($1 \le K \le 100{,}000$)가 주어진다. 다음 K줄에는 한 줄에 하나씩 문명 발상지에 해당하는 정사각형의 위치 (x,y)를 나타내는 두 자연수 x와 y가 주어진다. ($1 \le x,y \le N$)

출력 형식

표준 출력으로 모든 문명이 하나로 결합되는데 걸리는 최소 햇수를 정수로 출력한다.

부분문제의 제약 조건

- **부분문제 1:** 전체 점수 100점 중 19점에 해당하며 $2 \le N \le 100$, $2 \le K \le 100$이다.
- **부분문제 2:** 전체 점수 100점 중 35점에 해당하며 $2 \le N \le 1{,}000$, $2 \le K \le 100{,}000$이다.
- **부분문제 3:** 전체 점수 100점 중 46점에 해당하며 원래의 제약조건 이외에 아무 제약조건이 없다.

입력과 출력의 예

입력(1)

```
5 4
1 1
2 1
2 5
5 2
```

출력(1)

```
2
```

입력(2)

```
10 3
1 1
1 3
1 8
```

출력(2)

```
2
```

요리 강좌

여러분은 요리에 관심이 많아 요리 자격증을 취득하려고 한다. 요리 자격증을 취득하기 위해서는 총 M개의 강좌를 순서대로 한 번씩만 수강해야 한다. 이 요리 강좌는 N개의 학원에서 수강할 수 있는데, 학원마다 강좌별 수강비용은 다를 수 있다.

아래 표는 $M=5$, $N=4$인 경우, 학원별, 강좌별 수강비용의 예를 보여준다.

	강좌1	강좌2	강좌3	강좌4	강좌5
학원1	1	2	1	3	8
학원2	1	2	3	7	2
학원3	1	8	8	1	2
학원4	10	1	1	8	8

여러분은 비용을 줄이기 위해 중간에 학원을 변경할 수 있는데, 학원을 변경할 때마다 T의 추가 비용이 든다. 단, 학원 변경은 다음 규칙을 지켜야 한다.

(a) 하나의 학원에서 연속으로 수강할 수 있는 강좌의 수는 최소 S개, 최대 E개다.

$S=2$이고 $E=3$이라고 가정하자. 강좌1을 학원1에서 수강했다면, 강좌2는 무조건 학원1에서 수강해야 하고, 강좌 3은 학원1에서 수강할 수도 있고 다른 학원에서 수강할 수도 있다. 만약 강좌1부터 강좌3을 학원1에서 수강했다면, 강좌4는 무조건 다른 학원에서 수강해야 한다.

또한, $S=1$이고 $E=2$인 경우, 강좌1과 강좌2를 학원3에서 수강하고, 강좌3과 강좌4를 학원1에서 수강하고, 강좌5를 다시 학원3에서 수강하는 것도 가능하다.

(b) 학원마다 직전 강좌를 특정한 학원에서 수강한 학생은 받지 않는데, 이를 '불허용 학원'이라고 하자. 각 학원마다 불허용 학원은 하나씩 존재한다.

예를 들어, 아래 표의 학원1에 대한 정보는 학원2→학원1로 변경하는 것은 불가능하다는 것을 의미한다. 단, 학원2→학원4→학원1처럼 학원4 수강을 거쳐 변경하는 것은 가능하다.

	불허용 학원
학원1	학원2
학원2	학원3
학원3	학원4
학원4	학원3

예를 들어, $S=2$, $E=3$, $T=2$이고, 위 수강비용 표와 불허용 학원 표가 주어졌을 때, 강좌 순서대로 수강하는 학원의 번호가

예1) 1 → 1 → 1 → 1 → 3 인 경우, 규칙 (a)에 위배되어 수강이 가능하지 않다.

예2) 2 → 2 → 1 → 1 → 1 인 경우, 규칙 (b)에 위배되어 수강이 가능하지 않다.

예3) 3 → 3 → 1 → 1 → 3 인 경우는 가능한 수강 방법이고, 전체 비용은 $1+8+T+1+3+T+2 = 19$이다.

예4) 1 → 1 → 1 → 3 → 3 인 경우는 가능한 수강 방법이고, 전체 비용은 1+2+1+T+1+2 = 9이다.

강좌 비용, 강좌 선택에 필요한 규칙 정보가 주어졌을 때, 모든 강좌를 순서대로 수강하기 위해 필요한 최소 비용을 구하시오.

소스파일의 이름은 cook.c 또는 cook.cpp를 권장하지만, 서버에 제출하는 데는 다른 이름도 상관없다.

입력 형식

표준 입력으로 다음 정보가 주어진다. 첫 번째 줄에는 학원의 개수 N과 강좌 개수 M ($3 \le N \le 3{,}000$, $1 \le M \le 3{,}000$, $N \times M \le 3{,}000{,}000$), 학원 한 곳에서 연속 수강 가능한 최소 강좌 개수 S와 최대 강좌 개수 E($1 \le S \le E \le M$), 그리고 학원 변경 비용 T($0 \le T \le 35{,}000$)가 주어진다. 다음 N개의 줄에는 각 학원의 강좌별 수강비용이 주어진다. (수강비용은 1 이상 35,000 이하이다.) 처음 줄에는 학원1의 M개의 수강비용이 강좌 순서대로 공백을 사이에 두고 주어지고, 그 다음 줄부터 학원2, 학원3, ... , 학원N의 정보가 한 줄에 하나씩 순서대로 주어진다. 그 다음 N개의 줄에는 학원마다 불허용 학원 번호가 순서대로 주어진다.

출력 형식

표준 출력으로 최소 비용을 출력한다.

부분문제의 제약 조건

- **부분문제 1:** 전체 점수 100점 중 13점에 해당하며 $N, M \le 50$이다.
- **부분문제 2:** 전체 점수 100점 중 9점에 해당하며 $N, M \le 300$이다.
- **부분문제 3:** 전체 점수 100점 중 11점에 해당하며 $N \le 3{,}000$, $M \le 300$이다.
- **부분문제 4:** 전체 점수 100점 중 23점에 해당하며 $N \le 300$, $M \le 3{,}000$이다.
- **부분문제 5:** 전체 점수 100점 중 44점에 해당하며 원래의 제약조건 이외에 아무 제약조건이 없다.

입력과 출력의 예

입력(1)

```
4 5 2 3 2
1 2 1 3 8
1 2 3 7 2
1 8 8 1 2
10 1 1 8 8
2
3
4
3
```

출력(1)

```
9
```

입력(2)

```
4 5 1 1 0
1 2 1 3 8
1 2 3 7 2
1 8 8 1 2
10 1 1 8 8
2
3
4
3
```

출력(2)

```
9
```

조개 줍기

바닷가에 있는 정올시에는 여러 지역들이 정방형 격자 형태로 나뉘어져 있다. 각 지역에는 한 가구만 살고 있으며 가장 왼쪽 위에 있는 지역에는 수산시장이 있다. (수산시장이 있는 지역에도 한 가구가 산다.)

각 지역에서 수산시장으로 이동하려면 다음의 두 가지 방법만을 이용하여 이동한다.

1. 바로 위에 붙어있는 지역으로 이동
2. 바로 왼쪽에 붙어있는 지역으로 이동

각 지역에 사는 사람들은 매일 수산시장으로 출근하면서 지나가는 지역에서 조개를 주워 수산시장에서 판다. (출발하는 지역과 수산물 시장이 있는 지역에서도 조개를 주울 수 있다.)

각 지역에는 자연보호를 위해 그 지역을 지나가는 한 가구가 주울 수 있는 조개 개수의 최댓값이 있다. (여러 가구가 지나가면 가구마다 최댓값만큼 조개를 주울 수 있을 정도로 조개는 충분하다.)

예를 들어, 격자의 크기가 3 (행) × 3 (열) 이고 지역마다 한 가구가 주울 수 있는 조개 개수의 최댓값이 다음의 왼쪽 표와 같다고 하자.

3	2	7
4	2	6
5	3	8

3	5	12
7	9	18
12	15	26

각 지역에서 하루에 수산시장에 팔 수 있는 조개 개수의 최댓값은 오른쪽 표와 같다. 예를 들면, 맨 오른쪽 아래 격자에서 출발해서 최대로 조개를 줍는 방법은 위쪽으로 2번 이동하고 왼쪽으로 두 번 이동하는 방법이며 총 8+6+7+2+3=26개의 조개를 주울 수 있다. 그리고 이 도시의 아홉 지역에서 수산시장에 팔 수 있는 조개 개수의 최댓값을 모두 합하면 3+5+12+7+9+18+12+15+26=107 개이다.

정올시의 성실한 공무원들은 주기적으로 각 지역의 조개 숫자를 조사하여 한 가구가 주울 수 있는 조개 개수의 최댓값을 수정한다. 하지만 급격한 변화는 위험하므로 최댓값을 +1이나 -1만큼만 조정하는 것이 가능하다. 조정하지 않은 지역의 조개 개수의 최댓값은 그대로 유지된다. 예를 들면, 위의 왼쪽 표에서 격자의 1행, 2열의 2가 3으로 바뀐다면 각 지역에서 주울 수 있는 조개 개수의 최댓값과 각 지역에서 하루에 수산시장에 팔 수 있는 조개 개수의 최댓값이 다음과 같이 바뀐다.

3	**3**	7
4	2	6
5	3	8

3	6	13
7	9	19
12	15	27

격자 칸마다 주울 수 있는 조개 개수의

최댓값의 초기 값이 주어지고, 격자 칸에서 주울 수 있는 조개 개수의 최댓값의 변화를 입력으로 받아, 각 지역에서 하루에 수산시장에 팔 수 있는 조개 개수의 최댓값을 계산해서 그 합을 출력하는 프로그램을 작성하라.

소스파일의 이름은 shell.c 또는 shell.cpp를 권장하지만, 서버에 제출하는 데는 다른 이름도 상관없다.

입력 형식

표준 입력으로 다음 정보가 주어진다. 첫 번째 줄에는 격자의 행(열)의 개수를 나타내는 정수 $N(2 \le N \le 1{,}500)$이 주어진다. 다음 N줄에는 각 격자 칸에서 주울 수 있는 조개의 개수가 제일 윗 행부터 순서대로 한 줄에 한 행씩 주어진다. 한 행의 값은 가장 왼쪽 열의 값부터 하나씩 나열된다. 주어지는 값들은 0이상 1,000이하이다. 다음 N개의 줄에는 각 줄에 변화 명령이 하나씩 주어진다. 변화 명령의 첫 글자는 U 혹은 D이다. 이어서 빈칸을 하나 두고 두 자연수가 주어지는데, 첫 번째는 행 번호, 두 번째는 열 번호이다. 첫 글자가 U인 경우 행 번호, 열 번호에 해당하는 격자 칸에서 주울 수 있는 조개의 개수가 1 증가한다. D인 경우 해당 격자 칸에서 주울 수 있는 조개의 개수가 1 감소한다. 감소한 결과가 음수가 되는 경우는 없다. 각 변화에 대해서 아래에 지정한 값을 출력해야 한다. 주어진 각 변화 명령은 **이전 변화들이 모두 적용된 결과에** 적용된다.

출력 형식

표준 출력으로, 초기에 각 격자 칸의 입력을 기준으로 모든 지역에서 팔 수 있는 조개 개수의 최댓값의 합을 출력한다. 그 다음, 각 변화 명령에 대해 그 변화 명령을 적용한 후, 모든 지역에서 팔 수 있는 조개 개수의 최댓값의 합을 출력한다. 전체 출력은 $N+1$줄임에 주의하라.

부분문제의 제약 조건

- **부분문제 1:** 전체 점수 100점 중 12점에 해당하며 $N \le 100$이다.
- **부분문제 2:** 전체 점수 100점 중 34점에 해당하며, 변화는 모두 D이고 출력의 첫 줄이 20,000,000이하인 입력만 주어진다.
- **부분문제 3:** 전체 점수 100점 중 54점에 해당하며 원래의 제약조건 이외에 아무 제약조건이 없다.

입력과 출력의 예

입력

```
3
3 2 7
4 2 6
5 3 8
U 1 2
D 3 2
U 1 2
```

출력

```
107
111
110
114
```