



마리타의 남동생은 모든 장난감을 거실 바닥에 어질러놓았다! 다행히도 마리타는 장난감을 정리하는 특별한 로봇들을 개발하였다. 마리타는 어떤 로봇이 어떤 장난감을 집어야 하는지 결정하도록 당신에게 도움을 청했다.

장난감은 총 T 개가 있으며, 각각은 정수 무게 $w[i]$ 와 정수 크기 $s[i]$ 를 가진다. 로봇들은 연약한 로봇과 작은 로봇 두 가지 종류가 있다.

- 연약한 로봇은 총 A 개가 있다. 각 연약한 로봇에는 무게 제한 $x[i]$ 가 있어서, $x[i]$ 미만의 무게를 가진 장난감만을 운반할 수 있다. 장난감의 크기는 상관없다.
- 작은 로봇은 총 B 개가 있다. 각 작은 로봇에는 크기 제한 $y[i]$ 가 있어서, $y[i]$ 미만의 크기를 가진 장난감만을 운반할 수 있다. 장난감의 무게는 상관없다.

마리타의 로봇들 각각은 하나의 장난감을 정리하는 데 1분이 걸린다. 여러 로봇들이 서로 다른 장난감들을 동시에 정리할 수 있다.

당신의 임무는 마리타의 로봇들이 모든 장난감들을 정리할 수 있는지 결정하고, 만약 가능하다면, 모든 장난감을 정리하는데 걸리는 가장 짧은 시간을 찾는 것이다.

예시

첫 번째 예시로, 무게 제한 $X=[6, 2, 9]$ 를 가진 $A=3$ 개의 연약한 로봇과 크기 제한 $Y=[4, 7]$ 을 가진 $B=2$ 개의 작은 로봇이 있고, $T=10$ 개의 장난감이 아래와 같이 있다고 하자:

장난감 번호	0	1	2	3	4	5	6	7	8	9
무게	4	8	2	7	1	5	3	8	7	10
크기	6	5	3	9	8	1	3	7	6	5

모든 장난감들을 정리하는 데 걸리는 가장 짧은 시간은 3분이다:

	연약한 로봇 0	연약한 로봇 1	연약한 로봇 2	작은 로봇 0	작은 로봇 1
1분 째	장난감 0	장난감 4	장난감 1	장난감 6	장난감 2
2분 째	장난감 5		장난감 3		장난감 8
3분 째			장난감 7		장난감 9

두 번째 예시로, 무게 제한 $X=[2, 5]$ 를 가진 $A=2$ 개의 연약한 로봇과 크기 제한 $Y=[2]$ 를 가진 $B=1$ 개의 작은 로봇이 있고, $T=3$ 개의 장난감이 아래와 같이 있다고 하자:

장난감 번호	0	1	2
무게	3	5	2
크기	1	3	2

어떤 로봇도 무게 5, 크기 3짜리의 장난감을 정리할 수 없기 때문에, 모든 장난감들을 치우는 것은 불가능하다.

구현

다음 조건을 만족하는 함수 `putaway()` 를 구현한 파일을 제출하십시오.

구현해야 하는 함수: **putaway()**

```
C/C++      int putaway(int A, int B, int T,
             int X[], int Y[], int W[], int S[]);

Pascal     function putaway(A, B, T : LongInt;
                          var X, Y, W, S : array of LongInt) : LongInt;
```

설명

이 함수는 로봇들이 모든 장난감들을 정리하는데 걸리는 가장 짧은 시간(분)을 계산하여야 한다. 만약 정리하는 것이 불가능하다면, `-1` 을 리턴해야 한다.

파라미터

- `A`: 연약한 로봇의 개수.
- `B`: 작은 로봇의 개수.
- `T`: 장난감의 개수.
- `X`: 각 연약한 로봇의 무게 제한을 나타내는 길이가 `A` 인 배열.
- `Y`: 각 작은 로봇의 크기 제한을 나타내는 길이가 `B` 인 배열.
- `W`: 각 장난감의 무게를 나타내는 길이가 `T` 인 배열.
- `S`: 각 장난감의 크기를 나타내는 길이가 `T` 인 배열.
- 리턴값: 모든 장난감을 정리하는데 걸리는 가장 짧은 시간(분). 만약, 정리하는 것이 불가능하다면, `-1`.

예제 세션

다음 예제 세션은 위의 첫 번째 예시를 나타낸 것이다:

파라미터	값
A	3
B	2
T	10
X	[6, 2, 9]
Y	[4, 7]
W	[4, 8, 2, 7, 1, 5, 3, 8, 7, 10]
S	[6, 5, 3, 9, 8, 1, 3, 7, 6, 5]
리턴값	3

다음 예제 세션은 위의 두 번째 예시를 나타낸 것이다:

파라미터	값
A	2
B	1
T	3
X	[2, 5]
Y	[2]
W	[3, 5, 2]
S	[1, 3, 2]
리턴값	-1

제약 조건

- 시간 제한: 3 초
- 메모리 제한: 64 MB
- $1 \leq T \leq 1,000,000$
- $0 \leq A, B \leq 50,000$ 및 $1 \leq A + B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2,000,000,000$

서브태스크

서브태스크	점수	추가적인 입력 제한 사항
1	14	$T=2$ 및 $A+B=2$ (정확히 두 개의 장난감 및 두 개의 로봇)
2	14	$B=0$ (모든 로봇이 연약함)
3	25	$T \leq 50$ 및 $A+B \leq 50$
4	37	$T \leq 10,000$ 및 $A+B \leq 1,000$
5	10	(없음)

테스트용 입력 형식

당신의 컴퓨터에 있는 샘플 그레이더는 입력을 파일 `robots.in` 에서 읽어들이는데, 포맷은 다음과 같아야 한다.

- line 1: `A B T`
- line 2: `X[0] ... X[A-1]`
- line 3: `Y[0] ... Y[B-1]`
- the next `T` lines: `W[i] S[i]`

예를 들어, 위의 첫 번째 예시는 다음과 같은 형식을 따라야 한다:

```
3 2 10
6 2 9
4 7
4 6
8 5
2 3
7 9
1 8
5 1
3 3
8 7
7 6
10 5
```

만약 $A=0$ 또는 $B=0$ 이라면, 해당하는 (두 번째 또는 세 번째) 줄은 비어 있어야 한다.

언어 유의사항

C/C++ 당신의 프로그램은 `#include "robots.h"` 명령어를 통해 헤더 파일을 추가시켜야 한다.

Pascal 당신의 프로그램은 `unit Robots` 를 정의해야 한다. 모든 배열의 인덱스는 `1` 이 아닌 `0` 부터 시작한다.

예시를 위하여 컴퓨터에 있는 솔루션 템플릿을 참조하십시오.