



## International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

# robots

Polish — 1.0

Braciszek Bajtyny porozrzucił w pokoju mnóstwo zabawek. Bajtyna chciałaby zaprowadzić porządek w pokoju i odstawić wszystkie zabawki z powrotem na półkę. Aby się zbytnio nie namęczyć, zaprzęgnie do tej czynności skrecone naprędce robociki. Bajtyna poprosiła Cię o pomoc w określeniu, które robociki powinny zgarnąć poszczególne zabawki.

Na podłodze leży łącznie  $T$  zabawek. Każda z nich charakteryzuje się dwoma parametrami wyrażającymi się liczbami całkowitymi: masą  $W[i]$  i rozmiarem  $S[i]$ . Bajtyna ma dwa typy robocików: *słabe* oraz *małe*.

- Jest  $A$  słabych robocików. Każdy z nich ma pewną wytrzymałość  $X[i]$ , co oznacza, że może podnieść dowolny przedmiot o masie ściśle mniejszej niż  $X[i]$ . Rozmiar podnoszonego przedmiotu jest nieistotny.
- Jest  $B$  małych robocików. Każdy z nich ma pewną pojemność  $Y[i]$ , co oznacza, że może podnieść dowolny przedmiot rozmiaru ściśle mniejszego niż  $Y[i]$ . Masa podnoszonego przedmiotu jest nieistotna.

Odstawienie zabawki na półkę zajmuje każdemu robocikowi jedną minutę. Różne robociki mogą odstawiać różne zabawki jednocześnie.

Zadaniem Twojego programu jest stwierdzić, czy robociki Bajtyny mogą odstawić wszystkie zabawki na półkę oraz, jeśli odpowiedź jest twierdząca, wyznaczyć najkrótszy czas, w którym mogą to zrobić.

## Przykłady

Przykład 1. Załóżmy, że Bajtyna ma  $A=3$  słabe robociki o wytrzymałościach  $X=[6, 2, 9]$  i  $B=2$  małe robociki o pojemnościach  $Y=[4, 7]$ , a na podłodze leży  $T=10$  następujących zabawek:

Numer zabawki	0	1	2	3	4	5	6	7	8	9
Masa	4	8	2	7	1	5	3	8	7	10
Rozmiar	6	5	3	9	8	1	3	7	6	5

W tej sytuacji uporządkowanie wszystkich zabawek zajmie robocikom Bajtyny dokładnie trzy minuty:

	Słaby robocik 0	Słaby robocik 1	Słaby robocik 2	Mały robocik 0	Mały robocik 1
Pierwsza minuta	Zabawka 0	Zabawka 4	Zabawka 1	Zabawka 6	Zabawka 2
Druga minuta	Zabawka 5		Zabawka 3		Zabawka 8
Trzecia minuta			Zabawka 7		Zabawka 9

Przykład 2. Załóżmy, że Bajtyna ma  $A=2$  słabe robociki o wytrzymałościach  $X=[2,5]$  i  $B=1$  mały robocik o pojemności  $Y=[2]$ , a na podłodze leżą  $T=3$  następujące zabawki:

Numer zabawki	0	1	2
Masa	3	5	2
Rozmiar	1	3	2

Żaden z robocików nie jest w stanie podnieść zabawki o masie 5 i rozmiarze 3, więc w tej sytuacji nie da się uporządkować wszystkich zabawek.

## Implementacja

Powinieneś zgłosić plik z implementacją funkcji `putaway()`:

Twoja funkcja: `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;
```

## Opis

Ta funkcja powinna obliczyć, jaki jest najkrótszy czas (w minutach) potrzebny na uporządkowanie wszystkich zabawek. Jeśli uporządkowanie zabawek nie jest możliwe, funkcja powinna zwrócić `-1`.

### Argumenty

- `A` : liczba słabych robocików
- `B` : liczba małych robocików
- `T` : liczba zabawek
- `X` : tablica liczb całkowitych rozmiaru `A` określająca wytrzymałości poszczególnych słabych robocików
- `Y` : tablica liczb całkowitych rozmiaru `B` określająca pojemności poszczególnych małych robocików
- `W` : tablica liczb całkowitych rozmiaru `T` określająca masy poszczególnych zabawek
- `S` : tablica liczb całkowitych rozmiaru `T` określająca rozmiary poszczególnych zabawek
- *Wynik funkcji*: najmniejsza liczba minut potrzebnych do uporządkowania wszystkich zabawek lub `-1`, jeśli uporządkowanie nie jest możliwe.

## Przykład

Oto przykładowe argumenty funkcji oraz prawidłowy wynik dla pierwszego przykładu.

Argument	Wartość
<code>A</code>	<code>3</code>
<code>B</code>	<code>2</code>
<code>T</code>	<code>10</code>
<code>X</code>	<code>[6, 2, 9]</code>
<code>Y</code>	<code>[4, 7]</code>
<code>W</code>	<code>[4, 8, 2, 7, 1, 5, 3, 8, 7, 10]</code>
<code>S</code>	<code>[6, 5, 3, 9, 8, 1, 3, 7, 6, 5]</code>
Wynik	<code>3</code>

Oto przykładowe argumenty funkcji oraz prawidłowy wynik dla drugiego przykładu.

Argument	Wartość
<b>A</b>	2
<b>B</b>	1
<b>T</b>	3
<b>X</b>	[2, 5]
<b>Y</b>	[2]
<b>W</b>	[3, 5, 2]
<b>S</b>	[1, 3, 2]
<b>Wynik</b>	-1

## Ograniczenia

- **Maksymalny czas działania: 3 sekundy**
- **Limit pamięci: 64 MiB**
- $1 \leq T \leq 1\,000\,000$
- $0 \leq A, B \leq 50\,000$  i  $1 \leq A+B$
- $1 \leq X[i], Y[i], W[i], S[i] \leq 2\,000\,000\,000$

## Podzadania

Podzadanie	Punkty	Dodatkowe ograniczenia
1	14	$T=2$ i $A+B=2$ (dokładnie dwie zabawki i dwa robociki)
2	14	$B=0$ (wszystkie robociki są słabe)
3	25	$T \leq 50$ i $A+B \leq 50$
4	37	$T \leq 10,000$ i $A+B \leq 1000$
5	10	(brak)

---

## Uruchamianie lokalne

Przykładowy moduł oceniający na Twoim komputerze wczytuje dane z pliku `robots.in` w następującym formacie:

- wiersz 1: `A B T`
- wiersz 2: `X[0] ... X[A-1]`
- wiersz 3: `Y[0] ... Y[B-1]`
- kolejne `T` wierszy: `W[i] S[i]`

Dane z pierwszego przykładu powinny być więc podane w następującym formacie:

```
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
```

Jeśli `A=0` lub `B=0`, odpowiedni wiersz (wiersz 2 lub wiersz 3) powinien być pusty.

---

## Uwagi natury językowej

C/C++ Użyj dyrektywy `#include "robots.h"`.

Pascal Musisz zdefiniować `unit Robots`. Wszystkie tablice są indeksowane od `0` (a nie od `1`).

Przykłady znajdziesz w przykładowych rozwiązaniach na Twoim komputerze.