

Robocik i paczka

Limit pamięci: 64 MB

Firma BajteX planuje w tym roku wysłać na księżyc robocika. Jest to ogromne przedsięwzięcie poprzedzone niezliczonymi badaniami, testami, ulepszeniami. Aby być pewnym, że robocik będzie działał na księżycu, firma chce przeprowadzić testy na Ziemi.

Zadanie jest bardzo proste. Robocik znajduje się aktualnie w punkcie $(0, 0)$. Tam znajduje się też paczka. Na początku robocik musi podnieść tę paczkę przy okazji sprawdzając, dokąd ma ją zanieść. Następnie robocik może poruszać się do przodu, do tyłu, w prawo oraz w lewo za każdym razem pokonując odległość 1. Kiedy robocik dotrze do celu, powinien odstawić paczkę.

Tak się złożyło, że to Ty zaprogramujesz robocika.

Komunikacja

Aby użyć biblioteki, należy wpisać w swoim programie:

- C/C++: `#include "r0clib.h"`
- Python: `from r0clib import podnies, odstaw, do_przodu, do_tylu, w_prawo, w_lewo`

Biblioteka udostępnia następujące funkcje:

- `podnies()` – Ta funkcja zwraca parę liczb całkowitych (x, y) oznaczające położenie celu. Tę funkcję należy wywołać raz, na początku, przed jakimkolwiek ruchem.
- `odstaw()` – Ta funkcja odstawia paczkę w miejsce, gdzie aktualnie znajduje się robocik. Po wywołaniu tej funkcji program zostanie automatycznie zakończony.
- `do_przodu()`, `do_tylu()`, `w_prawo()`, `w_lewo()` – Te funkcje odpowiednio zwiększają y , zmniejszają y , zwiększają x , zmniejszają x robocika o 1.

Twój program nie może otwierać żadnych plików ani używać standardowego wejścia i wyjścia. Może on korzystać ze standardowego wyjścia diagnostycznego (`stderr`), jednak pamiętaj, że zużywa to cenny czas.

Rozwiązanie będzie kompilowane wraz z biblioteką następującymi poleceniami:

- C++: `g++ -O3 -static -std=c++20 r0clib.cpp r0c.cpp`
- Python: `python3 r0c.py`

Ocenianie

Dla każdego testu zachodzi $|x| \leq 10$, $|y| \leq 10$ oraz $(x, y) \neq (0, 0)$, a ponadto:

Podzadanie	Ograniczenia	Limit czasu	Punkty
1	$y = 0$ oraz $x > 0$	1 s	20
2	$y = 0$	1 s	20
3	brak dodatkowych ograniczeń	1 s	60

Przykładowy przebieg programu

Poniżej przedstawiono przebieg programu dla testu przykładowego, gdzie robocik musi dostarczyć paczkę w miejsce $(3, 0)$.

Wywołanie	Wynik	Komentarz
<code>podnies()</code>	<code>[3, 0]</code>	Robocik jest w punkcie $(0, 0)$, podniósł paczkę i ma ją dostarczyć do punktu $(3, 0)$.
<code>w_prawo()</code>		Robocik jest w punkcie $(1, 0)$.
<code>do_przodu()</code>		Robocik jest w punkcie $(1, 1)$.
<code>w_prawo()</code>		Robocik jest w punkcie $(2, 1)$.
<code>do_tylu()</code>		Robocik jest w punkcie $(2, 0)$.
<code>w_prawo()</code>		Robocik jest w punkcie $(3, 0)$.
<code>odstaw()</code>		Robocik odstawia paczkę w punkcie docelowym.



Robocik i paczka

Limit pamięci: 64 MB

Eksperymenty

Przykładowe błędne rozwiązania wraz z przykładowymi bibliotekami znajdują się w folderze d1azaw. Biblioteki mogą różnić się zachowaniem od tych na sprawdzaczkach i nie spełniać założeń zadania. Mają one jedynie pokazać sposób interakcji z programem.

Twoje rozwiązanie skompilowane z przykładową biblioteką wczytuje ze standardowego wejścia dwie liczby całkowite x oraz y oznaczające położenie celu. Przykładowa biblioteka wypisuje wykonane ruchy przez robocika.