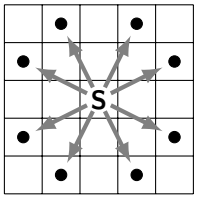
Zadanie: SKO
Skoczek

XXX OI, etap III, dzień pierwszy. Plik źródłowy `sko.*` **Dostępna pamięć: 256 MB.** *21.03.2023*

Dana jest szachownica rozmiaru $n \times n$ składająca się z n^2 pól, które opisujemy współrzędnymi (x, y), przy czym x to numer kolumny, a y to numer wiersza ($1 \leq x, y \leq n$).

Na jednym z pól (x_S, y_S) tej szachownicy (nie wiemy którym) stoi skoczek. Możemy zadawać pytania postaci „W ilu minimalnie ruchach skoczek może się dostać z pola, na którym stoi, na pole o współrzędnych (x, y)?"

Skoczek (zwany potocznie konikiem) jest figurą szachową, która porusza się zawsze o dwa pola w jednym kierunku i jedno pole w kierunku prostopadłym. Zatem skoczek w jednym ruchu może przemieścić się na jedno z ośmiu pól (o ile oczywiście takie pole znajduje się w granicach szachownicy). Na poniższym rysunku przedstawiono możliwe ruchy skoczka z pola oznaczonego literą **S** na pola oznaczone kropkami:



Napisz program, który odgadnie pozycję skoczka za pomocą małej liczby pytań.

Komunikacja

Twój program będzie korzystał z dostarczonej biblioteki odpowiadającej na podane pytania. Aby użyć biblioteki, należy wpisać w swoim programie:

- C/C++: `#include "skolib.h"`
- Python: `from skolib import daj_n, pytanie, odpowiedz`

Biblioteka udostępnia następujące funkcje:

- `daj_n`() – Funkcja daje w wyniku liczbę całkowitą n, oznaczającą rozmiar szachownicy.
- `pytanie`(x, y) – Wynikiem funkcji jest minimalna liczba ruchów, które musi wykonać skoczek, aby przejść z pola (x_S, y_S) na pole (x, y). Musi być spełnione $1 \leq x, y \leq n$. W żadnym ruchu skoczek nie może opuścić szachownicy.
- `odpowiedz`(x_S, y_S) – Ta funkcja pozwala zgłosić odgadniętą przez program pozycję skoczka. Musi być spełnione $1 \leq x_S, y_S \leq n$. Należy ją wykonać dokładnie raz. Po wykonaniu tej funkcji program zostanie automatycznie zakończony. Jeśli podana w funkcji pozycja skoczka będzie błędna, program zakończy się z werdyktem *Błędna odpowiedź*.

Biblioteka **nie musi** ustalać pozycji skoczka na początku interakcji z Twoim programem. Może ona w trakcie interakcji zmieniać pozycję, o ile nowa pozycja jest nadal zgodna z wynikami zwróconymi przez dotychczasowe wywołania funkcji `pytanie`.

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++17 skolib.cpp sko.cpp`
- Python: `python3 sko.py`

Uwaga: *Podane na górze ograniczenie pamięci dotyczy tylko Twojego rozwiązania, a zatem nie wlicza pamięci wykorzystywanej przez bibliotekę.*

Olimpiada Informatyczna finansowana jest ze środków Ministerstwa Edukacji i Nauki w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Warunki	Liczba punktów
1	$4 \leq n \leq 10$	40
2	$4 \leq n \leq 100$	20
3	$4 \leq n \leq 1000$	40

Jeśli m to liczba wywołań funkcji `pytanie`, które Twój program wykonał w danym przypadku testowym, to Twoje rozwiązanie otrzyma następujący procent punktów za ten test (odpowiednio przeskalowany w przypadku przekroczenia połowy limitu czasowego):

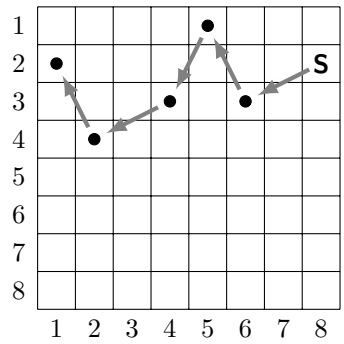
Liczba wywołań	Procent punktów
$m \leq 6$	100% punktów za test
$7 \leq m \leq 12$	$(85 - 5m)$% punktów za test
$13 \leq m \leq 22$	$(46 - 2m)$% punktów za test
$23 \leq m$	0% punktów za test (werdykt *Błędna odpowiedź*)

Przykładowy przebieg programu

Poniżej przedstawiono przykładowy przebieg programu dla testu przykładowego.

Wywołanie	Wynik	Opis
`daj_n()`	`8`	$n = 8$
`pytanie(1, 2)`	`5`	
`pytanie(1, 7)`	`4`	
`pytanie(8, 2)`	`0`	
`pytanie(8, 7)`	`3`	
`odpowiedz(8, 2)`	–	Prawidłowa odpowiedź z użyciem 4 pytań, 100% punktów za test

Na poniższym rysunku przedstawiono jedną z najkrótszych dróg skoczka z pola $(8, 2)$ na pole $(1, 2)$:



Testy „ocen”:

`1ocen:` $n = 10$, $x_S = 3$, $y_S = 7$;

`2ocen:` $n = 100$, $x_S = 3$, $y_S = 7$.

Eksperymenty

Przykładowe **błędne** rozwiązania wraz z przykładowymi bibliotekami znajdują się w folderze `dlazaw`. Biblioteki mogą różnić się zachowaniem od tych używanych do ostatecznej oceny rozwiązań i nie spełniać założeń zadania. Mają one jedynie pokazać sposób interakcji z programem.

Twoje rozwiązanie skompilowane z przykładową biblioteką wczytuje ze standardowego wejścia opis planszy – liczbę n, a następnie n wierszy zawierających po n liczb, będących odległościami skoczka do kolejnych pól. Przykładowa biblioteka używa tych odległości do odpowiadania na pytania Twojego programu.

Olimpiada Informatyczna finansowana jest ze środków Ministerstwa Edukacji i Nauki w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025”.