

Ukryty skarb



[Franek i Stefan](#) bawią się w poszukiwanie skarbu. Gra toczy się w pewnym obszarze, w obrębie którego Franek ukrywa skarb (w wybranym przez siebie miejscu o współrzędnych całkowitych dodatnich), a Stefan, przy użyciu specjalnego elektronicznego skanera, musi ten skarb znaleźć.

Użycie skanera polega po prostu na jego włączeniu, a wtedy na wyświetlaczu pojawia się jeden z dziewięciu komunikatów: "N", "E", "S", "W", "NE", "NW", "SE", "SW" lub "OK". Jeśli Stefan zobaczy na ekranie komunikat "OK", to oznacza, że znajduje się właśnie w miejscu ukrycia skarbu. Pozostałe komunikaty informują, w którym kierunku (względem miejsca użycia skanera) skarb się znajduje. Niestety, bateria zasilająca skaner jest na wyczerpaniu, więc należy używać go bardzo oszczędnie.

Napisz program, który na podstawie informacji o rozmiarze obszaru i odpowiedzi uzyskiwanych ze skanera, będzie wyznaczał współrzędne kolejnych miejsc, w których należy dokonać skanowania, tak aby w jak najmniejszej liczbie prób znaleźć skarb.

Wejście/Wyjście

UWAGA! W zadaniu użyty jest specjalny sędzia, symulujący działanie skanera, co wymaga odpowiedniej interakcji Twojego programu.

W pierwszej linii wejścia znajduje się liczba całkowita t oznaczająca liczbę testów ($1 \leq t \leq 10000$).

Dla każdego testu należy najpierw odczytać z wejścia dwie liczby całkowite X i Y oznaczające wymiary obszaru, w którym ukryty jest skarb. X to rozpiętość w kierunku wschód-zachód, a Y - północ-południe ($1 \leq X, Y \leq 10^9$). Wysunięty najbardziej na południowy-zachód punkt obszaru ma współrzędne $(1, 1)$.

Następnie Twój program powinien podawać na standardowe wyjście po dwie liczby całkowite x, y ($1 \leq x \leq X, 1 \leq y \leq Y$) oznaczające współrzędne miejsca, w którym Stefan używa skanera, po czym odczytywać z wejścia komunikat wyświetlony na jego ekranie. Odczytanie komunikatu "OK" powinno skutkować przejściem do kolejnego z t testów. Po zakończeniu ostatniego testu, Twój program musi zakończyć działanie.

Liczba dopuszczalnych użycí skanera zależy od wymiarów obszaru. Jej przekroczenie skutkuje zakończeniem działania programu i komunikatem "błąd wykonania (other)".

Przykład

Wejście: 2

Wejście: 10 12

Wyjście (Twój program): 3 6

Wejście (Skaner): NE

Wyjście: 7 8

Wejście: SW

Wyjście: 5 7

Wejście: W

Wyjście: 4 7

Wejście: OK

Wejście: 1 1

Wyjście: 1 1

Wejście: OK

UWAGA: Program po wypisaniu każdej linii powinien opróżniać bufor wyjściowy. Np. w C++ można to zrobić poleceniem `fflush(stdout)`, a przy wykorzystaniu strumienia `cout` wystarczy wypisywaną linię zakończyć używając `endl`.