

Escape cube



Bierzesz udział w zabawie znanej jako "Escape room". Tylko że tym razem musisz wydostać się z labiryntu pomieszczeń, tworzących sześcian.

Grę rozpoczynasz w pomieszczeniu, którego położenia w sześcianie nie znasz. Nie wiesz też jaki jest rozmiar sześcianu, który może wynosić od 2 (8 pokoi) do 27 (19683 pokoje). Z każdego pomieszczenia można przejść do jednego z co najwyżej sześciu sąsiednich (pokoje na skraju sześcianu mają mniej niż sześciu sąsiadów). Sześć kierunków, w których można się poruszać oznaczmy literami: E (East), W (West), N (North), S (South), U (Up), D (Down). Oczywiście pary kierunków przeciwnych to E-W, N-S i U-D.

W niektórych pokojach umieszczone są pułapki i wejście do takiego pomieszczenia kończy się przegraną. Aby wygrać, należy przedostać się do pokoju o współrzędnych (0,0,0) (czyli takiego, że ruch w kierunku W lub S lub D oznaczałby wyjście poza sześcian) i wykonać ruch w kierunku 'W' lub 'S'. Wyjście poza sześcian w innym miejscu oznacza przegraną.

Żeby bezpieczne wydostanie się z sześcianu było możliwe, przy każdym z sześciu wyjść z danego pokoju zapisana jest liczba całkowita (oznaczymy ją X), która daje informacje na temat tego, co znajduje się po drugiej stronie przejścia. Wartość -1 oznacza, że ruch w tym kierunku spowodowałby wyjście z sześcianu. Nieujemna wartość X informuje, że po drugiej stronie przejścia jest pokój. Wiadomo też jaką cechę mają liczby oznaczające przejście do pokoju z pułapką (co jest chyba najcenniejszą dla Ciebie informacją).

Jeśli reszta z dzielenia X przez pewną małą liczbę pierwszą P wynosi R , to w sąsiednim pokoju jest pułapka.

Jeśli reszta z dzielenia X przez P jest różna od R , to sąsiedni pokój nie zawiera pułapki.

Niestety nie znasz ani liczby P ani R . Wiesz za to jakie są dopuszczalne wartości liczby P , oraz że pułapki nie zawiera pokój startowy, żaden z pokoi sąsiadujących z pokojem startowym oraz pokój, do którego należy się przedostać, czyli o współrzędnych (0,0,0).

Napisz program, który na podstawie wartości liczb X , wygeneruje sekwencję ruchów, prowadzących do wydostania się z sześcianu. Zagwarantowane jest, że da się to zrobić dokonując logicznego wnioskowania.

Wejście/Wyjście

UWAGA! W zadaniu użyty jest specjalny sędzia, który wymaga odpowiedniej interakcji Twojego programu.

Jedyna wartość, jaką na początku należy odczytać z wejścia, to liczba k ($1 \leq k \leq 7$) oznaczająca ile najmniejszych liczb pierwszych należy brać pod uwagę jako możliwą wartość liczby P . Np. dla $k=7$, P może mieć jedną z wartości: 2, 3, 5, 7, 11, 13, 17.

Następnie Twój program powinien podawać na wyjście jedno z 13 dopuszczalnych poleceń: **?A, ?E, ?W, ?N, ?S, ?U, ?D, E, W, N, S, U, D.**

Po wydaniu polecenia **?A**, sędzia poda na wejście, sześć liczb całkowitych oznaczających wartości **X** ($0 \leq X \leq 10^9$) znajdujące się przy przejściach w kolejnych kierunkach (obowiązuje kolejność: E W N S U D).

Po wydaniu jednego z sześciu pozostałych poleceń ze znakiem "?", sędzia poda na wejście jedną liczbę całkowitą, oznaczającą wartość **X** przy przejściu w wybranym kierunku (oznaczonym literą po ?).

Wydanie polecenia złożonego z pojedynczej litery, oznacza przejście w wybranym kierunku. W tym wypadku sędzia sprawdza, czy wykonując taki ruch wyszedłeś właśnie z pokoju (0,0,0) w kierunku 'S' lub 'W', co daje wynik "accepted" (pod warunkiem, że Twój program zakończy w tej sytuacji działanie) czy może znalazłeś się w pokoju z pułapką lub poza sześcianem (co skutkuje zakończeniem programu z wynikiem "wrong answer"). Jeśli nie zaszła żadna z wymienionych sytuacji, sędzia czeka na Twoje kolejne polecenie.

Przykład

```
Wejście: 2
Wyjście: ?A
Wejście: -1 12 -1 36 -1 14
Wyjście: D
Wyjście: ?A
Wejście: -1 7 -1 11 18 -1
Wyjście: U
Wyjście: S
Wyjście: ?A
Wejście: -1 10 20 -1 -1 21
Wyjście: N
Wyjście: W
Wyjście: ?A
Wejście: 35 -1 -1 25 -1 4
Wyjście: E
Wyjście: D
Wyjście: S
Wyjście: ?W
Wejście: 2
Wyjście: W
Wyjście: S
```

Pomoc do przykładu:

Z pierwszej linii wejścia wynika, że pod uwagę należy brać jedynie reszty z dzielenia przez 2 i przez 3.

Z przebiegu gry można wywnioskować, że odbywała się ona w sześcianie o wymiarze 2, a pokoje z pułapkami oznaczone były liczbami, które z dzielenia przez 3 dają resztę 1.

Polecenie ?W wydane pod koniec gry jest właściwie zbędne, ponieważ wtedy w kierunku zachodnim znajduje się pokój końcowy, a ten z pewnością nie zawiera pułapki.

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`.