Zadanie: POS
Poszukiwania

OLIMPIADA INFORMATYCZNA

XVI OI, etap III, dzień drugi. Plik źródłowy `pos.*` **Dostępna pamięć: 32 MB.** *02.04.2009*

Młodzi i szaleńczo w sobie zakochani Bajtek i Bajtyna wpadli w niezłe tarapaty. Zły czarownik Bitocy porwał Bajtynę, mając nadzieję na sowity okup. Bajtek nie poddaje się jednak — nie jest zbyt majętny, więc postanowił odbić swoją wybrankę. Bitocy nie lubi walki wręcz, zaproponował więc inne rozwiązanie całej sytuacji. Jeśli młodzieniec zgadnie, na którym piętrze wieży znajduje się uwięziona, czarodziej puści ją wolno.

Pięter jest bardzo wiele — są ponumerowane od 1 do n. Jedyną pomocą dla Bajtka mogą być pytania zadawane czarodziejowi. Pytania muszą być postaci „Czy Bajtyna jest wyżej/niżej niż piętro x?". Oczywiście, Bajtek może wybrać dokładnie jeden z wyrazów „wyżej", „niżej", a także dowolnie ustalić liczbę x. Bitocy zawsze odpowiada na tak postawione pytanie zgodnie z prawdą, ale każe sobie zapłacić a bajtalarów, jeśli odpowiedź brzmi „tak", lub b bajtalarów, gdy odpowiedź brzmi „nie". Cóż, jeśli Bajtek zbytnio zubożeje, a Bitocy nadmiernie się wzbogaci, to Bajtyna może zechcieć zostać u czarodzieja...

Bajtek zastanawia się, jakie pytania zadawać. Niestety Bajtyna słyszy całą rozmowę, tzn. kolejne pytania Bajtka i odpowiedzi Bitocego, a jest osobą bardzo oszczędną. Jeśli tylko Bajtek wyda choćby o jednego bajtalara więcej, niż jest to (w najgorszym przypadku) niezbędne do ustalenia jej położenia, to obrazi się na niego śmiertelnie i odejdzie z Bitocym. Dokładniej, jeżeli w pewnym momencie rozmowy da się wywnioskować, że od tego momentu, niezależnie od dalszych odpowiedzi Bitocego, Bajtek może odgadnąć położenie Bajtyny, wydając przy tym nie więcej niż K bajtalarów, a od tego momentu Bajtek wyda kwotę większą niż K, to jego szanse u Bajtyny spadną do zera (punktów za dany test). Pomóż Bajtkowi!

Komunikacja

Powinieneś zaimplementować program, który rozwiąże problem Bajtka, korzystając z dostarczonej biblioteki (symulującej czarodzieja Bitocego). Aby użyć biblioteki, należy wpisać w swoim programie:

- C/C++: `#include "poslib.h"`
- Pascal: `uses poslib;`

Biblioteka udostępnia trzy procedury i funkcje:

- `inicjuj` — podaje liczbę pięter n oraz koszty a i b. Powinna zostać użyta dokładnie raz, na samym początku działania programu.
  - C/C++: `void inicjuj(int *n, int *a, int *b);`
  - Pascal: `procedure inicjuj(var n, a, b: longint);`
- `pytaj` — znak c oznacza rodzaj pytania ('W' dla wyżej lub 'N' dla niżej), a x to numer piętra. Wynikiem funkcji jest wartość logiczna odpowiedzi czarodzieja. Twój program może użyć tej funkcji dowolną liczbę razy.
  - C/C++: `int pytaj(char c, int x);` (0 oznacza fałsz, a 1 prawdę),
  - Pascal: `function pytaj(c: char; x: longint): boolean;`
- `odpowiedz` — za pomocą tej procedury/funkcji podajesz numer piętra, na którym jest Bajtyna. Powinna zostać użyta dokładnie raz. Jej wykonanie zakończy działanie Twojego programu.
  - C/C++: `void odpowiedz(int wynik);`
  - Pascal: `procedure odpowiedz(wynik: longint);`

Twój program nie może otwierać żadnych plików ani używać standardowego wejścia i wyjścia. Rozwiązanie będzie kompilowane wraz z biblioteką następującymi poleceniami:

- C: `gcc -O2 -static poslib.c pos.c -lm`
- C++: `g++ -O2 -static poslib.c pos.cpp -lm`
- Pascal: `ppc386 -O2 -XS -Xt pos.pas`

W katalogu **/home/zawodnik/rozw/lib** możesz znaleźć przykładowe pliki bibliotek i nieoptymalne rozwiązania ilustrujące sposób ich użycia. Aby podane powyżej polecenia kompilacji działały, pliki bibliotek powinny znajdować się w bieżącym katalogu.

Limity

Możesz założyć, że $1 \leqslant n \leqslant 10^9$ oraz $1 \leqslant a, b \leqslant 10\,000$.

Przykładowy przebieg programu

Wywołanie funkcji	**Zwracane wartości i wyjaśnienia**
Pascal: `inicjuj(n,a,b);` C lub C++: `inicjuj(&n,&a,&b);`	Od tego momentu $n = 5$, $a = 1$, $b = 2$.
`pytaj('W',3);`	Wynik równa się 0/`false`. Pytasz, czy Bajtyna jest powyżej trzeciego piętra. Uzyskujesz odpowiedź, że nie. Płacisz 2 bajtalary.
`pytaj('N',2);`	Wynik równa się 0/`false`. Pytasz, czy jest poniżej drugiego piętra. Uzyskujesz odpowiedź „nie”, za którą płacisz kolejne 2 bajtalary.
`pytaj('W',2);`	Wynik równa się 1/`true`. Pytasz, czy jest powyżej drugiego piętra. Uzyskujesz odpowiedź „tak”, za którą płacisz bajtalara.
`odpowiedz(3);`	Odpowiadasz, że Bajtyna znajduje się na trzecim piętrze. Jest to poprawna odpowiedź. Wydałeś łącznie 5 bajtalarów.

Powyższy przebieg interakcji jest poprawny, ale nieoptymalny, a więc program nie uzyskałby punktów za taki test. W szczególności, dobrze napisany program potrafi dla danych $n = 5$, $a = 1$, $b = 2$ tak zadawać pytania, żeby w każdym przypadku wydać co najwyżej 4 bajtalary.