Zadanie: GRA
Gra Ulama

Olimpiada Informatyczna

V OI, etap trzeci, dzień drugi. Plik źródłowy `gra.*` Dostępna pamięć: 256 MB.

Wybitny polski matematyk Stanisław Ulam zaproponował następującą grę dwuosobową. Gracz A wybiera ze zbioru liczb całkowitych $X = \{0, 1, \ldots, 2047\}$ liczbę x, nie zdradzając jej graczowi B.

Gracz B musi dowiedzieć się, jaką liczbę wybrał gracz A. W tym celu może zadawać pytania postaci: „Czy x jest elementem zbioru Y?", gdzie Y jest dowolnym podzbiorem zbioru X. Gracz A udziela odpowiedzi „TAK" lub „NIE", przy czym wolno mu co najwyżej raz skłamać.

Twoim zadaniem jest znalezienie i zaprogramowanie strategii zadawania pytań dla gracza B, która pozwoli mu wskazać poprawnie liczbę wybraną przez gracza A, przy możliwie najmniejszej liczbie zadanych pytań.

Przykład

Rozważmy grę Ulama dla zbioru $X = \{0, 1, 2, 3\}$. Przyjrzyjmy się następującemu ciągowi pytań i odpowiedzi:

B: Czy x należy do $\{0, 1\}$?
A: TAK
B: Czy x należy do $\{0\}$?
A: TAK

W tym miejscu B nie może jeszcze być pewnym, że liczbą wybraną przez A jest 0, ponieważ A mógł skłamać odpowiadając na drugie pytanie.

B: Czy x należy do $\{0\}$?
A: NIE

B nie wie, która z dwóch ostatnich odpowiedzi jest prawdziwa.

B: Czy x należy do $\{0\}$?
A: NIE

Szukaną liczbą na pewno nie jest 0. Odpowiedź na pierwsze pytanie musiała być prawdziwa, zatem szukaną liczbą jest 1.

B: x=1.

Zadanie

Twoim zadaniem jest zaprogramowanie strategii zadawania pytań umożliwiającej graczowi B znalezienie liczby x przy pomocy jak najmniejszej liczby pytań. Rolę gracza A będzie pełnił program dostarczony przez organizatorów.

Musisz zaprogramować moduł zawierający następujące trzy procedury (funkcje w przypadku języka C):

- `nowa_gra` – ta procedura będzie wywoływana na początku każdej gry i tylko wtedy, możesz za jej pomocą dokonywać inicjalizacji swoich struktur danych;
- `daj_pytanie` – ta procedura służy do zadawania pytania i w tym celu powinna odpowiednio wypełniać tablicę pytanie opisaną w następnym paragrafie;
- `analizuj_odpowiedz` – ta procedura powinna odczytać z opisanej dalej zmiennej globalnej odpowiedź na ostatnio zadane pytanie i dokonać analizy tej odpowiedzi, jeśli w wyniku analizy ostatniej odpowiedzi liczba x zostanie zidentyfikowana, procedura powinna zasygnalizować ten fakt nadając właściwe wartości odpowiednim zmiennym globalnym.

Program nadzorujący grę na początku wywoła napisaną przez Ciebie procedurę `nowa_gra` po czym cyklicznie będzie wykonywał następujęce czynności:

- wywołanie procedury `daj_pytanie`
- odpowiadanie na pytanie
- wywołanie procedury `analizuj_odpowiedz`

do momentu, gdy Twój program stwierdzi, że odgadł liczbę x (procedura `analizuj_odpowiedz` umieści właściwą wartość w odpowiedniej zmiennej).

UWAGA: Nie zakładaj, że program symulujący gracza A faktycznie ustala liczbę do odgadnięcia przed rozpoczęciem gry. Może dobierać ją w trakcie gry w taki sposób, aby zmusić Twój program do zadania największej liczby pytań lecz by po zakończeniu gry, kiedy znana jest „pomyślana" liczba oraz lista zadanych pytań i odpowiedzi, nie można było udowodnić, że gracz A skłamał więcej niż raz. Tak więc, powinieneś dążyć do tego, aby liczba pytań, jakie Twój program będzie musiał zadać w najgorszym przypadku, była jak najmniejsza.

Komunikacja

Komunikacja pomiędzy napisanym przez Ciebie modułem, a programem nadzorującym grę odbywa się przez zmienne globalne.

Zadanie pytania „Czy x należy do Y ?" w procedurze `daj_pytanie` polega na wypełnieniu tablicy pytanie typu `array[0..2047] of boolean` (w języku C tablicy `char pytanie [2048]`) w taki sposób, że liczba i jest elementem Y wtedy i tylko wtedy, gdy `pytanie[` i`] = TAK`, gdzie `TAK` to predefiniowana stała o wartości `true` w Pascalu oraz 1 w C (podobnie predefiniowano stałą `NIE` o wartości $false$ w Pascalu oraz 0 w C). Odpowiedź na zadane pytanie jest umieszczana w zmiennej odpowiedz typu `boolean` (w języku C typu `char`): `odpowiedz = TAK` wtedy i tylko wtedy, gdy odpowiedź jest twierdząca. O odnalezieniu szukanej liczby x Twój program powinien poinformować w procedurze analizuj odpowiedz, zapisując w zmiennej `wiem` typu `boolean` (w języku C typu `char`) `TAK`, a następnie w zmiennej `x` znalezioną liczbę. Podanie odpowiedzi kończy grę, jeśli jest to odpowiedź błędna, gracz B przegrywa.

Pliki

W sekcji Pliki znajdziesz następujące pliki:

- `gramod.pas` (odpowiednio `gramod.h` i `gramod.c`) – moduł zawierający podstawowe definicje oraz deklaracje zmiennych komunikacyjnych;
- `gra.pas` – szkielet modułu grającego; powinieneś uzupełnić ten plik definicjami funkcji `nowa_gra`, `daj_pytanie` i `analizuj_odpowiedz` (dla piszących w języku C nagłówki powyższych 3 funkcji znajdują się w pliku `gra.h`, musisz napisać plik `gra.c` z definicjami tych funkcji);
- `rywal.pas` (odpowiednio `rywal.c`) – przykładowy program nadzorujący grę i korzystający z Twojego modułu, możesz z jego pomocą sprawdzić, czy Twój moduł spełnia specyfkację zadania