

Zadanie 6: Wyszukiwanie

Opened: Monday, 15 January 2024, 10:00 AM

Due: Wednesday, 24 January 2024, 4:00 PM

Uwaga: Osoby, które chciałyby być zwolnione z egzaminu i chcą w tym celu wykorzystać wynik z tego zadania, zachęcamy, żeby oddały swoje rozwiązanie przed swoim ostatnim laboratorium.

Ala wymyśliła sobie pewną liczbę x z zakresu od 1 do n . Bolek próbuje zgadnąć tę liczbę. Jedyne pytania, jakie może zadawać Bob, są postaci: "Czy $x < y$?" dla dowolnie wybranych przez niego liczb y . Na każde z takich pytań Bolek otrzymuje odpowiedź "Tak" lub "Nie". Zadaniem Boleka jest odgadnąć x , zadając jak najmniej pytań. Niestety Ala może czasem kłamać. Twoim zadaniem jest napisanie programu, który gra w tę grę jako Bolek.

Rolę Ali odgrywa biblioteka, której interfejs otrzymujesz w pliku [wys.h](#). Na początku interakcji z biblioteką powinieneś wywołać funkcję

```
void dajParametry(int &n, int &k, int &g);
```

Funkcja podaje Ci trzy liczby: liczbę n określającą zakres, liczbę k oznaczającą maksymalną liczbę razy, ile Ala może składać w jednej grze, oraz liczbę g oznaczającą liczbę gier do rozegrania przy danych parametrach n oraz k . Od wywołania tej funkcji rozpoczyna się pierwsza gra.

W każdej z gier możesz zadawać Ali pytania jedynie za pomocą funkcji

```
bool mniejszaNiz(int y);
```

Wynikiem funkcji jest odpowiedź Ali na pytanie "Czy $x < y$?", dla nieznanej Ci liczby x . Dokładniej, jeśli wynikiem funkcji jest `true`, oznacza to, że Ala odpowiada "Tak", a jeśli wynikiem jest `false`, to Ala odpowiada "Nie". Odpowiedź Ali nie musi być prawdziwa, jednak w ciągu jednej gry Ala może składać co najwyżej k razy.

Na końcu gry powinieneś podać Ali swoją odpowiedź za pomocą funkcji

```
void odpowiedz(int x);
```

Wywołanie tej funkcji powoduje automatycznie rozpoczęcie kolejnej gry, jeśli jeszcze nie zostało rozegranych g gier. W przeciwnym razie wywołanie funkcji zakończy działanie Twojego programu.

Twój program powinien zadawać minimalną możliwą liczbę zapytań w następującym sensie: Jeśli dla danych n oraz k Bob może zgadnąć odpowiedź za pomocą y pytań niezależnie od tego, jaka jest wartość x oraz kiedy Ala zdecyduje się kłamać (jeśli w ogóle), to Twój program dla tych n oraz k musi zawsze udzielać poprawnej odpowiedzi za pomocą co najwyżej y pytań.

Twój program powinien działać dla parametrów: $1 \leq n \leq 12$, $0 \leq k \leq 3$, $1 \leq g \leq 10.000$. Czas działania programu nie powinien mieć znaczenia, o ile nie będzie przesadnie zły (kilka sekund będzie OK).

Przykładową strategię Ali znajdziesz w pliku [wyslib.cpp](#). Przykładowe, nieoptymalne rozwiązanie implementujące strategię Boba możesz znaleźć w pliku [wys_naive.cpp](#). (**UWAGA:** rozwiązanie to zostało lekko poprawione.) Rozwiązanie to możesz skompilować wraz z podaną biblioteką za pomocą poniższej komendy, przy takich samych opcjach kompilacji jak w poprzednich dwóch zadaniach:

```
g++ @opcje wys_naive.cpp wyslib.cpp -o wys_naive
```



[wys_naive.cpp](#)

17 January 2024, 2:45 PM



[wys.h](#)

14 January 2024, 11:37 PM




[wyslib.cpp](#)


14 January 2024, 11:37 PM

?

Submission status

Attempt number	This is attempt 1.
Submission status	Submitted for grading
Grading status	Graded
Time remaining	Assignment was submitted 2 hours 4 mins early
Last modified	Wednesday, 24 January 2024, 1:55 PM
File submissions	<div><div> wys.cpp</div><div>24 January 2024, 1:55 PM</div></div>
Submission comments	<div><div></div><div>Comments (0)</div></div>

Feedback

Grade	5.00 / 5.00
Graded on	Wednesday, 24 January 2024, 2:20 PM
Graded by	<div> Łukasz Bożyk</div>

Contact us



Follow us

 Contact site support

You are logged in as Witold Formański (Log out)

Data retention summary

Get the mobile app

Get the mobile app

This theme was developed by

conecti.me

Moodle, 4.1.10 (Build: 20240422) | moodle@mimuw.edu.pl