

Task 1. Namuhs

Humanoidy, obcy z planety Htrae (którzy są nazywani namuhs) są na szczycie rozwoju i potrafią tworzyć żyjące postacie. Po długiej podróży wokół wszechświata, specjalny zespół stworzył listę planet posortowaną po ich odległości od Htrae. Zespół wysłał stworzoną listę do „Najwyższej rady” namuhs’ów, którzy analizują te informacje szczegółowo i przypisują każdej planecie liczbę całkowitą oznaczającą jej potencjał do wykonania eksperymentu. Prezydentem “Najwyższej rady” jest Deni i to ona musi zdecydować, które planety będą brały udział w eksperymencie. Ponieważ odległości w przestrzeni są bardzo duże, nawet dla zaawansowanej rasy, musi zostać wybrany zbiór planet, które są sąsiednie na liście, tzn. muszą tworzyć spójny fragment. Całkowita suma potencjałów planet jest równa sumie potencjałów poszczególnych planet. „Najwyższa rada” zdecydowała, że wybrany zbiór planet musi mieć najwyższy możliwy potencjał. Deni przypomniała sobie o jednej planecie, nie tak daleko odległej – o Ziemi, gdzie cywilizacja (zwana ludźmi) nie jest bardzo zaawansowana, ale są tam stworzenia, które potrafią jej pomóc przygotować program, który rozwiąże jej problem. Namuhs nie chcą stracić sekretnej informacji, dlatego jedynym sposobem dostępu do danych jest zadawanie pytań dotyczących porównywania potencjałów dwóch przedziałów planet.

Zadanie

Należy zaimplementować funkcję *find_max*, która będzie kompilowana z plikami źródłowymi Jury i musi znaleźć dwie liczby całkowite opisujące przedział z maksymalnym potencjałem. Funkcja otrzyma jeden parametr *N* – liczbę planet. Jury przypisze planetom potencjały w odpowiedni sposób. Twoim celem jest znalezienie segmentu z maksymalnym potencjałem za pomocą pytań porównujących sumy przedziałów. Mamy gwarancję, że istnieje tylko jedna poprawna odpowiedź.

Szczegóły implementacyjne

Funkcja *find_max* ma następujący nagłówek:

```
void find_max (int N, int& left, int& right);
```

Ta funkcja jest wołana tylko raz przez program Jury z argumentem *N* – liczbą planet na liście. Kiedy Twój program znajdzie odpowiedź, powinnaś/powinniście zapisać ją w parametrach *left* i *right* – lewy i prawy koniec znalezionej przedziału (planety numerujemy od 1).

Do komunikacji z programem Jury jest dostarczona funkcja:

```
bool compare_segments (int left1, int right1, int left2, int right2);
```

Ta funkcja pyta, czy suma potencjałów planet na przedziale od *left1* do *right1* (włącznie) jest wyższa lub równa sumie potencjałów planet na przedziale od *left2* do *right2* (włącznie). Dla każdego wywołania musi zachodzić, że $1 \leq left1 \leq right1 \leq N$ i $1 \leq left2 \leq right2 \leq N$. Zauważ, że funkcja *compare_segments* działa w czasie liniowym od rozmiaru wejścia, tzn. potrzebuje ona $(right1 - left1 + 1) + (right2 - left2 + 1)$ iteracji, aby znaleźć odpowiedź.

Należy wysłać na system tylko jeden plik **namuhs.cpp**, który będzie zawierał funkcję *find_max*. Twój kod może zawierać również inne linie kodu i inne funkcje potrzebne do działania funkcji *find_max*, ale nie powinien zawierać głównej funkcji - *main*. Na początku swojego kodu powinnaś/powinieneś napisać **#include "namuhs.h"**.

Ograniczenia

$$2 \leq N \leq 10^5$$

W 10% testów zachodzi: $N \leq 10^2$

W innych 30% testów zachodzi: $N \leq 10^3$

Przykładowa komunikacja z programem Jury

Niech listą planet będzie: 2, -3, 5, -2, 3. Wtedy odpowiedzią jest przedział od 3. do 5. pozycji włącznie. Całkowita liczba planet to 5, zatem program Jury wywoła funkcję *find_max* z parametrami:

find_max(5, left, right);

Przykładowa komunikacja służąca do znalezienia odpowiedzi może być następująca:

Wołanie funkcji programu Jury	Zwracana wartość	Wyjaśnienie
<i>compare_segments(3,5,1,1)</i>	true	Twój program pyta, czy całkowita suma potencjałów planet w przedziale od 3. do 5. jest większa lub równa niż całkowita suma potencjałów planet na przedziale od 1. do 1., tzn. czy $5+(-2)+3 \geq 2$, to jest prawdą, zatem funkcja zwraca true.
<i>compare_segments(1,1,3,5)</i>	false	To znaczy, że całkowita suma potencjałów planet w przedziale od 3. do 5. jest ściśle większa niż suma potencjałów od 1. do 1.
<i>compare_segments(3,5,2,2)</i>	true	Funkcja wykonuje 4 iteracje.
<i>compare_segments(3,5,3,3)</i>	true	Funkcja wykonuje 4 iteracje.
<i>compare_segments(3,5,4,4)</i>	true	Funkcja wykonuje 4 iteracje.
<i>compare_segments(3,5,5,5)</i>	true	Funkcja wykonuje 4 iteracje.
<i>compare_segments(3,5,1,2)</i>	true	Funkcja wykonuje 5 iteracji.
<i>compare_segments(3,5,2,3)</i>	true	Funkcja wykonuje 5 iteracji.
<i>compare_segments(3,5,3,4)</i>	true	Funkcja wykonuje 5 iteracji.
<i>compare_segments(3,5,4,5)</i>	true	Funkcja wykonuje 5 iteracji.
<i>compare_segments(3,5,1,3)</i>	true	Funkcja wykonuje 6 iteracji.
<i>compare_segments(3,5,2,4)</i>	true	Funkcja wykonuje 6 iteracji.
<i>compare_segments(3,5,1,4)</i>	true	Funkcja wykonuje 7 iteracji.
<i>compare_segments(3,5,2,5)</i>	true	Funkcja wykonuje 7 iteracji.
<i>compare_segments(3,5,1,5)</i>	true	Udało nam się sprawdzić, że suma potencjałów planet z przedziału od 3. do 5. jest wyższa lub równa sumie wszystkich pozostałych przedziałów. Zatem ten przedział daje nam optymalny wynik, a więc ustawiamy parametry funkcji <i>find_max</i> na wartości: <i>left</i> =3 i <i>right</i> =5 i funkcja kończy działanie.

Lokalne testowanie

Dostarczone są pliki **Lgrader.cpp** i **namuhs.h** do lokalnego testowania. Powinieneś umieścić je w tym samym folderze co Twój program **namuhs.cpp** i musisz skompilować plik **Lgrader.cpp**. W ten sposób otrzymasz program do sprawdzania poprawności Twojej funkcji. Ten program będzie wymagał następujące dane na standardowym wejściu:

- W pierwszej linii jedną dodatnią liczbę całkowitą – liczbę planet na liście.
- W drugiej linii potencjały kolejnych planet na liście.

Wyjściem tego programu będzie odpowiedź znaleziona przez Twoją funkcję.

Testowanie na systemie



Możesz testować swoje rozwiązania na systemie. Format wejścia musi być taki sam jak dla lokalnego programu oceniającego. Wyjście, które otrzymasz to liczba większa niż 0, jeśli odpowiedź Twojego programu jest poprawna lub 0, jeśli odpowiedź Twojego programu jest nieprawidłowa.