

Задача 1. Кевочани

Хуманоидните извънземни от планетата Ямез, наричани кевочани, са достигнали връхна точка в развитието си и вече могат да сътворяват живи същества. Затова те решават да се „правят“ на богове и да създават биологични системи на все още необитаеми планети. След голямо пътешествие из познатата ни Вселена специален отряд кевочани изготвят списък от известен брой планети, подредени по разстоянието им от Ямез. Отрядът връчва изготвения списък на „Висшия съвет“ на кевочаните, който разглежда подробно информацията и взема решение да зададе на всяка планета едно цяло число, което да характеризира потенциала ѝ за експеримента. Главното лице на съвета е Дени и тя трябва да реши точно кои планети ще участват в експеримента. Понеже космическите разстояния са много големи дори и за тази напреднала раса, трябва да се избере едно единствено множество от планети, които да са разположени една до друга в списъка – т.е. да образуват някакъв интервал. Общият потенциал на едно множество от планети е равен на сумата от потенциалите на всяка планета в него. „Висшият съвет“ решава, че трябва да се вземе множеството планети с най-голям общ потенциал. Дени си спомнила, че знае една планета немного далеч – планетата Земя, където цивилизацията (на така наречените хора) не е достатъчно напреднала, но там има същества, които могат да ѝ помогнат да напишат програма за този проблем. Кевочаните не искат да дадат директен достъп до събраната секретна информация, а единственият достъп, който хората да имат до информацията, е да питат и да сравняват сборовете на два интервала от планети в списъка.

Задача. Напишете функция *find_max*, която ще се компилира с програмата на журито (на Дени), и ще върне двойка числа за интервал от съседни планети, чийто общ потенциал е възможно най-голям. Тази функция ще получи едно число *N* – броя на планетите. Програмата на журито ще разполага и с потенциалите на планетите, подредени в указания ред. Вашата цел е да задавате въпроси за сравнение на интервали от съседни планети, така че да откриете интервала с най-голям общ потенциал. Гарантирано е, че е точно един!

Детайли по реализацията

Функцията *find_max* трябва да има следния формат:

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

Тя се вика веднъж от програмата на журито с аргумент *N* – броят на планетите в списъка. Когато Вашата програма открие интервала планети с най-голям общ потенциал, трябва да запише отговора в параметрите *left* и *right* – съответно левият и десният край на интервала (номерацията започва от 1).

За комуникация с програмата на журито Ви е предоставена функцията:

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

Тя задава въпрос дали общият потенциал на планетите с позиции от *left1* до *right1* (включително) в списъка е по-голям или равен на общият потенциал на планетите с позиции от *left2* до *right2* (включително). Тук трябва да са изпълнени следните условия: $1 \leq left1 \leq right1 \leq N$ и $1 \leq left2 \leq right2 \leq N$. Имайте предвид, че функцията *compare_segments* работи с линейна сложност по входните данни, т.е. ѝ трябвават $(right1 - left1 + 1) + (right2 - left2 + 1)$ на брой итерации за работа!

Вие трябва да предадете към системата файл **namuhs.cpp**, който съдържа функция *find_max*. Той може да съдържа също и друг код и функции, необходими за работата на *find_max*, но не трябва да съдържа главната функция *main*. В началото си Вашият файл трябва да съдържа: **#include "namuhs.h"**.

Ограничения

$2 \leq N \leq 10^5$

В 10% от тестовете: $N \leq 10^2$

В други 30% от тестовете: $N \leq 10^3$

Примерна комуникация с програмата на журито

Нека списъкът с потенциалите на планетите е следният: 2, -3, 5, -2, 3. Тук отговорът е интервалът от планети от позиция 3 до позиция 5 включително. Общият брой планети е 5, т.е. програмата на журито ще извика Вашата функция *find_max* така:

```
find_max(5, left, right);
```

Примерна комуникация за откриване на отговора може да е следната:

Извикване на функцията на журито	Върнат резултат	Обяснение
<i>compare_segments(3,5,1,1)</i>	true	Програмата Ви пита дали общият потенциал на планетите в интервала от 3 до 5 е по-голям или равен на тези от 1 до 1, т.е. дали $(5+(-2)+3) \geq 2$, което е вярно и функцията връща true.
<i>compare_segments(1,1,3,5)</i>	false	Това означава, че общият потенциал на планетите в интервала от 3 до 5 е строго по-голям на тези от 1 до 1.
<i>compare_segments(3,5,2,2)</i>	true	Тук трябва 4 итерации за функцията.
<i>compare_segments(3,5,3,3)</i>	true	Тук трябва 4 итерации за функцията.
<i>compare_segments(3,5,4,4)</i>	true	Тук трябва 4 итерации за функцията.
<i>compare_segments(3,5,5,5)</i>	true	Тук трябва 4 итерации за функцията.
<i>compare_segments(3,5,1,2)</i>	true	Тук трябва 5 итерации за функцията.
<i>compare_segments(3,5,2,3)</i>	true	Тук трябва 5 итерации за функцията.
<i>compare_segments(3,5,3,4)</i>	true	Тук трябва 5 итерации за функцията.
<i>compare_segments(3,5,4,5)</i>	true	Тук трябва 5 итерации за функцията.
<i>compare_segments(3,5,1,3)</i>	true	Тук трябва 6 итерации за функцията.
<i>compare_segments(3,5,2,4)</i>	true	Тук трябва 6 итерации за функцията.
<i>compare_segments(3,5,1,4)</i>	true	Тук трябва 7 итерации за функцията.
<i>compare_segments(3,5,2,5)</i>	true	Тук трябва 7 итерации за функцията.
<i>compare_segments(3,5,1,5)</i>	true	Вече проверихме, че общият потенциал на планетите в интервала от 3 до 5 е по-голям или равен от общият потенциал на всички възможни интервали, т.е. това е отговорът. Сега програмата присвоява на параметрите на функцията <i>find_max</i> съответно <i>left=3</i> и <i>right=5</i> и трябва да се прекрати работата на тази функция.

Локално тестване

За локално тестване са предоставени файловете **Lgrader.cpp** и **namuhs.h**. Сложете ги в същата папка, в която е Вашият файл **namuhs.cpp** и компилирайте **Lgrader.cpp**. Така ще получите програма, с която ще проверите верността на функцията Ви. Програмата ще изисква от стандартния вход следната последователност от данни:

- на първия ред: едно положително число – броят на планетите в списъка.
- на втория ред: потенциалите на съответните планети.

На изход ще получите отговора, който е намерила Вашата функция.

Предаване на тестове към системата

Можете да предавате тестове към системата. Форматът на входните данни е същият като за локалния грейдър. На изход получавате число по-голямо от 0, ако вашата програма е дала правилен отговор и 0, ако е дала грешен отговор.