

### Задатак 1. Namuhs

Ванземаљци са планете Htrae се зову namuhs се налазе на цивилизацијској прекретници. Током еволуције специјални тим је направио списак планета сортиран према растојању од планете Htrae. Специјалци су сали тај списак Влади namuhs која детаљно анализира добијене информације и придружује свакој планети цео број да опише њен потенцијал у овом интерактивном експерименту. Шеф Владе је Deni и она мора да одлучи које ће планете ући у план. Биће изабран само један скуп планета, које ће чинити суседи у списку тј. планете формирају сегмент. Укупан потенцијал једног скупа планета је једнак суми потенцијала сваке планете која је чланица скупа. “Влада” је одлучила да одабере скуп са највећим укупним потенцијалом. Deni моли Земљане из Србије да реше интерактивни задатак. Namuhs је неповерљив, те морати написати програм који ће постављати упите у вези поређења сума два сегмента планета.

### Task

Морате имплементирати функцију *find\_max* која ће бити компајлирана уз изворни код Комисије (тј. Deni) који мора да врати два броја за сегмент суседних планета које одликује највећи могући укупни потенцијал. Ова функција ће као параметре имати: број  $N$  – број планета. Комисија поседује вредности потенцијала планета у одговарајућем редоследу. Ваш задатак је да путем постављања питања у вези поређења сегмената суседних планета да пронађете сегмент са највишим могућим потенцијалом. Гарантује се да постоји само један одговор!

### Детаљи имплементације

Функција *find\_max* има следећи прототип:

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

Комисија једном позове ову функцију у свом програму са параметром  $N$  – број планета у списку. Када Ваш програм нађе одговор, морате резултат рада сачувати у два параметра *left* и *right* – леви и десни крај сегмента редом (нумерација почиње од 1).

За комуникацију са програмом Комисије морате имати следеће функције:

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

Ова функција поставља питање да ли је укупни потенцијал планета на позицијама између *left1* и *right1* (укључено) већи или једнак укупном потенцијалу планета на позицијама између *left2* и *right2* (укључено). Следећи услови морају бити испуњени:  $1 \leq left1 \leq right1 \leq N$  и  $1 \leq left2 \leq right2 \leq N$ . Уочите, функција *compare\_segments* има линеарну сложеност у односу на улазне податке тј. Потребно је укупно  $(right1 - left1 + 1) + (right2 - left2 + 1)$  итерација да произведе резултат!

Морате предати интерактивном систему само датотеку **namuhs.cpp**, у којој се налази функција *find\_max*. Ова датотека може да садржи и друге функције или линије кода које су потребне за рад функције *find\_max*, али датотека НЕ СМЕ да има функцију - *main*. На почетку Ваше датотеке морати додати линију: **#include "namuhs.h"**.

### Ограничења

$2 \leq N \leq 10^5$

In 10% of tests:  $N \leq 10^2$

In other 30% of tests:  $N \leq 10^3$

### Примерак рада интерактивног задатка

Let the list with the potential of the planets is: 2, -3, 5, -2, 3. Here the answer is the segment with planets from position 3 to position 5 inclusive. The total number of planets is 5 so the program of the jury will call your function *find\_max* as follows:

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

Sample communication for finding the answer can be the following:

Calling function from the jury program	Return result	Explanation
<i>compare_segments(3,5,1,1)</i>	True	Your program asks whether the total potential of the planets in the segment from 3 to 5 is higher than or equal to the total potential of the planets from position 1 to position 1, i.e. if $(5+(-2)+3) \geq 2$ , which is true and the function returns true.
<i>compare_segments(1,1,3,5)</i>	False	This means that the total potential of the planets in the segment from 3 to 5 is strictly higher than the potential from 1 to 1.
<i>compare_segments(3,5,2,2)</i>	True	Here are needed 4 iterations for the function.
<i>compare_segments(3,5,3,3)</i>	True	Here are needed 4 iterations for the function.
<i>compare_segments(3,5,4,4)</i>	True	Here are needed 4 iterations for the function.
<i>compare_segments(3,5,5,5)</i>	True	Here are needed 4 iterations for the function.
<i>compare_segments(3,5,1,2)</i>	True	Here are needed 5 iterations for the function.
<i>compare_segments(3,5,2,3)</i>	True	Here are needed 5 iterations for the function.
<i>compare_segments(3,5,3,4)</i>	True	Here are needed 5 iterations for the function.
<i>compare_segments(3,5,4,5)</i>	True	Here are needed 5 iterations for the function.
<i>compare_segments(3,5,1,3)</i>	True	Here are needed 6 iterations for the function.
<i>compare_segments(3,5,2,4)</i>	True	Here are needed 6 iterations for the function.
<i>compare_segments(3,5,1,4)</i>	True	Here are needed 7 iterations for the function.
<i>compare_segments(3,5,2,5)</i>	True	Here are needed 7 iterations for the function.
<i>compare_segments(3,5,1,5)</i>	True	We have already checked that the total potential of the planets in the segment from 3 to 5 is higher than or equal to the potential of each of the other segments. So this segment gives us the optimal answer. Now the program set the parameters of the function <i>find_max</i> the values: <i>left</i> =3 and <i>right</i> =5 and the function has to stop its work.

### Local testing

You are given the files **Lgrader.cpp** and **namuhs.h** for local testing. You have to place them in the same folder as your program **namuhs.cpp** and you have to compile the file **Lgrader.cpp**. In such a way, you will make a program to check the correctness of your function. This program will require the following data from the standard input:

- on the first line: one positive number – the number of the planets in the list.
- on the second line: the potentials of the corresponding planets.

The output of the program will be the answer that your function found.

### Submitting user tests to the system

You can give tests to the system. The format of the input data has to be the same as in the local grader. The output you will receive is a number greater than 0, if the answer of your program is correct, and 0, if the output of your program is not correct.