

**ТРЕТО КОНТРОЛНО СЪСТЕЗАНИЕ НА
РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ЯМБОЛ, 6 май, 2018 г.
Група А, 11-12 клас**

ЗАДАЧА АК1. ЕТАЖЕРКА

В стаята на Пешо има етажерка с N тетрадки, които съдържат описание на всички гениални идеи, които витаят в неговата глава. Тетрадките са номерирани с числата от 1 до N . Пешо си има любимо подреждане на тетрадките върху етажерката, което не е задължително да бъде по нарастване на номерата им, и той никак не обича някой да му ги размества. Пешо е купил специален робот, който може да запомня подреждането на тетрадките и да пресмята броя на инверсиите в дадено подреждане. Роботът смята, че две тетрадки образуват инверсия, ако тетрадката с по-малък номер стои по-надясно от тетрадката с по-голям номер. Например в подреждането (2,1,5,3,4) има три двойки тетрадки, които образуват инверсия (2,1), (5,3) и (5,4), т.е. броят на инверсиите в това подреждане е 3.

За нещастие, след ремонт на стаята, тетрадките върху етажерката били разместени, а Пешо забравил любимото си подреждане. Роботът го помни, но той може да съобщава само броя на инверсиите в запомненото от него подреждане. Пешо може да поиска от робота да размени местата на две тетрадки в подреждането, което роботът помни в момента. След такава размяна, роботът ще запомни новото подреждане (като забрави старото) и ще съобщи броя на инверсиите в него. Пешо може да повтаря заявките си към робота до момента, в който реши, че има достатъчно информация, за да възстанови любимото си подреждане на тетрадките.

Вие ще изпълнявате ролята на Пешо, а програма на журито – ролята на робота.

Задача

Напишете функция *bookshelf()*, която ще се компилира с програма на журито и ще и задава въпроси докато възстанови първоначалната подредба на книгите..

Детайли по реализацията

Функцията *bookshelf()* трябва да има следния формат:
`void bookshelf(int N, long long Inv);`

Тя се вика веднъж от програмата на журито с аргументи N и Inv – броя на тетрадките и броя на инверсиите в първоначалната им подредба.

За комуникация с програмата на журито Ви се предоставят функциите:

`long long bookswap(int i, int j)`

и

`void answer(int p[])`

Чрез извикване на функция *bookswap()* Вие подавате заявка към програмата на журито (робота) да размени местата на тетрадките, които се намират на позиции i и j и да Ви върне броя на инверсиите в новото подреждане на тетрадките.

Когато Вашата програма реши, че е намерила първоначалното подреждане на тетрадките, тя трябва да извика функция *answer()*, като в масив p предаде на програмата на журито намереното подреждане – елементът $p[i]$ трябва да съдържа номера на тетрадката, която се намира на позиция с номер i .

Вие трябва да предадете към системата файл **bookshelf.cpp**, който съдържа функция *bookshelf()*. Той може да съдържа и друг код, необходим за работата на функция *bookshelf*, но не трябва да съдържа *main()*.

В началото си Вашият файл трябва да съдържа `#include "bookshelf.h"`.

**ТРЕТО КОНТРОЛНО СЪСТЕЗАНИЕ НА
РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР
ЯМБОЛ, 6 май, 2018 г.
Група А, 11-12 клас**

Ограничения

$$2 \leq N \leq 100\,000$$

$0 \leq$ брой на заявките към робота за разместване на тетрадки $\leq 200\,000$

В 10% от тестовете $2 \leq N \leq 400$;

В други 20% от тестовете $400 < N \leq 5\,000$;

В други 30% от тестовете $5\,000 < N \leq 50\,000$;

Оценяване

Всеки тест се оценява отделно.

Примерна комуникация с програмата на журито

Нека първоначалното подреждане на тетрадките, което Вашата програма трябва да познае, е 4 3 1 5 2. Броят на инверсиите в това подреждане е 6 – (4,3), (4,1), (4,2), (3,1), (3,2) и (5,2). Програмата на журито ще извика Вашата функция *bookshelf* така:

bookshelf(5,6);

Един от възможните диалози след това е:

Функция на журито, която викате	Резултат, който получавате
<i>bookswap</i> (1,2)	5
<i>bookswap</i> (1,2)	6
<i>bookswap</i> (3,2)	5
<i>bookswap</i> (3,2)	6
<i>bookswap</i> (5,4)	5
<i>bookswap</i> (5,4)	6
<i>bookswap</i> (4,1)	7
<i>bookswap</i> (4,1)	6
<i>bookswap</i> (5,1)	3
<i>bookswap</i> (5,1)	6
<i>bookswap</i> (5,2)	5
<i>bookswap</i> (5,2)	6
<i>bookswap</i> (5,3)	7
<i>bookswap</i> (5,6)	6
<i>answer</i> (4 3 1 5 2) Тук като параметър на <i>answer</i> , разбира се, стои масив.	С това работата на програмата приключва – Вашата програма даде верен резултат.

Локално тестване

За да можете да тествате Вашата функция *bookshelf()* на локалния си компютър, Ви се предоставят файлове *Lgrader.cpp* и *bookshelf.h*. Компилирайте ги заедно с вашия файл **bookshelf.cpp** и ще получите програма, с която да тествате функцията си. Програмата изисква от стандартния вход да се въведе следната последователност от данни:

От първи ред: две цели положителни числа – брой на тетрадките и брой на инверсиите в първоначалното подреждане.

От втори ред: номерата на тетрадките в първоначалното подреждане, разделени с интервали.

На изход ще получите първоначалното подреждане, което Вашата програма е „познала“.