

ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА

Велико Търново, 19 - 21 април 2019 г.

Група А, 11 – 12 клас

Задача А3. АВТОМАТИЗАЦИЯ

В работата си в Кулбанк на Лора често ѝ се налага да обработва голямо количество данни, изпълнявайки прости повтарящи се действия. Очаквано, тя смята, че това не са задачи, които трябва да се вършат от човек. За жалост обаче, шефовете на Лора не искат да отпуснат пари за наемането на програмист, който да автоматизира процеса. Като неин добър приятел, помогнете безплатно на Лора като напишете програма **automation**, която автоматизира обработката на данни.

Данните са представени като списък от N цели числа $A_0, A_1 \dots A_{N-1}$. На всяка стъпка програмата Ви ще получава заявка от един от следните 3 типа:

- 1) Да се пресметне сумата на елементите с индекси от L до R включително.
- 2) Всеки елемент с индекс от L до R включително става равен на дадена стойност K
- 3) Стойността на всеки елемент с индекс от L до R включително бива разделена целочислено на дадена стойност K

За всяка заявка от тип 1 програмата Ви трябва да пресметне вярната сума.

Детайли по имплементацията

Заявките трябва да бъдат обработвани в реално време. Програмата ви трябва да съдържа функции със следните прототипи:

```
void init(int subtask, int N, const int A[]);  
long long getSum(int L, int R);  
void setValues(int L, int R, int K);  
void divideValues(int L, int R, int K);
```

Функцията *init* ще бъде извикана точно веднъж в началото на програмата преди каквито и да е заявки. Като аргументи на функцията се задават подзадачата, от която е текущият тест (*subtask*), както и големината на списъка от числа (N) и началните им стойности (A).

Останалите три функции и параметрите им отговарят на гореописаните заявки.

Към системата трябва да изпратите файл **automation.cpp**, в който са имплементирани четирите функции. В него може да имате каквито искате помощни функции, структури, променливи и т.н. Той единствено не трябва да съдържа функция *main* и трябва да включва хедър файла **automation.h** чрез указание към препроцесора `#include "automation.h"` в началото.

Ограничения

$$0 \leq L \leq R < N, Q \leq 300\,000$$

$$0 \leq A_i, K \leq 10^9$$

Q бележи общия брой заявки в рамките на един тест.

Във всяка заявка от тип 3 е изпълнено $K \geq 2$

ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА

Велико Търново, 19 - 21 април 2019 г.

Група А, 11 – 12 клас

Подзадачи и оценяване

За да получите точките за дадена подзадача, решението Ви трябва успешно да премине всички тестове в нея.

Подзадача	Точки	N, Q	Допълнителни ограничения
1	9	$N, Q \leq 2\,000$	
2	8	$N, Q \leq 75\,000$	Няма заявки от тип 2, За всяка заявка от тип 3 е изпълнено $K=2$
3	16	$N, Q \leq 75\,000$	За всяка заявка от тип 3 е изпълнено $K=2$
4	16	$N, Q \leq 75\,000$	Няма заявки от тип 2
5	25	$N, Q \leq 75\,000$	
6	26	$N, Q \leq 300\,000$	

Локално тестване

Предоставени са Ви файловете **automation.h** и **Lgrader.cpp**, които можете да компилирате заедно с Вашата програма, за да я тествате.

Форматът на входа за **Lgrader** е:

От първия ред се въвежда единствено цяло число от 1 до 6 – номерът на подзадачата, от която е тестът.

От втория ред се въвеждат N и Q – дължината на списъка от числа и общия брой заявки.

От третия ред се въвеждат N числа – началните стойности.

Всеки от следващите Q реда съдържа по една заявка в един от следните формати:

- "1 L R"
- "2 L R K"
- "3 L R K"

Можете да правите каквито си промени искате по предоставените Ви файлове.

Примерен тест

Вход	Изход
1	9
5 6	13
7 0 5 4 2	0
1 1 3	
2 2 4 8	
3 1 4 3	
1 0 4	
3 0 1 100	
1 0 1	

ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА

Велико Търново, 19 - 21 април 2019 г.

Група А, 11 – 12 клас

Показаният примерен вход е във формата описан за **Lgrader.cpp**, а изходът съответства на стойностите върнати от Вашата имплементация на функцията **getSum**. Програмата Ви **не** трябва да чете на стандартния вход или да пише на стандартния изход.

Обяснение на примерния тест

Тестът е част от подзадача 1 и съдържа $N = 5$ числа и $Q = 6$ заявки. Началните стойности са:

7 0 5 4 2

Първата заявка е сума на елементи с индекси от 1 до 3 и отговорът е $0 + 5 + 4 = 9$

След втората заявка елементите с индекси от 2 до 4 стават равни на 8. Числата са:

7 0 8 8 8

След третата заявка елементите с индекси от 1 до 4 се разделят целочислено на 3. Числата са:

7 0 2 2 2

Четвъртата заявка е за сума и отговорът ѝ е $7 + 0 + 2 + 2 + 2 = 13$

След петата заявка елементите с индекси от 0 до 1 се разделят целочислено на 100. Числата са:

0 0 2 2 2

Последната заявка е за сума и отговорът ѝ е $0 + 0 = 0$