

ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА

Велико Търново, 19 - 21 април 2019 г.

Група А, 11 – 12 клас

Задача А1. MONOPOLY

Всички знаят българската фирма, която продава популярната игра „Дени Монопол“. Ръководството решава, че ще търси ново лого, където ще изобрази всички служители. *За целта им трябва такава тяхна наредба, че да няма служител, който да е разположен след някой от неговите непосредствени подчинени.* Поради големината на фирмата се оказва, че съществуват много подредби и първо трябва да се види колко са те. Тази важна задача възлагат на Дени – най-добрата програмистка на фирмата. За съжаление тя е затрупана от много работа напоследък и няма време. Дени се обръща с молба към Вас да напишете програма **monopoly**, която да изпълни възложената ѝ задача.

В резултат на големия брой реорганизации и прилагането на най-съвременни методи на управление, фирмата има, меко казано, странна организационна структура. В тази структура има две нормални неща:

- Всеки служител без обикновените работници има известен брой подчинени служители.

- Няма служител, при който някой от подчинените му служители (не непременно непосредствено) да му се окаже началник (не непременно непосредствен).

Следват странните резултати от реорганизациите:

- Един служител може да има повече от един непосредствен началник.

- Ако за даден служител x_1, x_2, \dots, x_k са непосредствените му началници, то съществува наредба i_1, i_2, \dots, i_k на $1, 2, \dots, k$, така че ако ги подредим в реда $x_{i_1}, x_{i_2}, \dots, x_{i_k}, x_{i_1}$ ще е началник (не задължително непосредствен) на x_{i_2} , от своя страна x_{i_2} ще е началник (не задължително непосредствен) на x_{i_3} и \dots , $x_{i_{k-1}}$ ще е началник на x_{i_k} .

Дени Ви дава структурата на фирмата от N -те служители и M -те връзки между тях и иска търсената бройка подредби на служителите. Служителите са номерирани с числата от 1 до N . Понеже това може да е много голямо число, Дени ще се задоволи само с остатъка му по модул 10^9+7 .

Вход

От първия ред на стандартния вход се въвеждат две цели положителни числа N и M – броят служители и броят връзки между тях. От следващите M реда се въвеждат по две цели числа x и y , които показват, че служителят с номер x е непосредствен началник на служителя с номер y (съответно y е непосредствен подчинен на x).

Изход

Едно единствено число – остатък при деление с 10^9+7 на намерения брой подредби на служителите.

Ограничения

- ♣ $1 \leq N \leq 100000$
- ♣ $1 \leq M \leq 200000$

ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА

Велико Търново, 19 - 21 април 2019 г.

Група А, 11 – 12 клас

Подзадачи

Подзадача	Точки	N	M	Други ограничения
1	15	≤ 10	≤ 45	Няма допълнителни ограничения.
2	35	≤ 19	≤ 171	Няма допълнителни ограничения.
3	20	≤ 100	≤ 200	Реалният отговор е до $2 \cdot 10^5$.
4	30	≤ 100000	≤ 200000	Няма допълнителни ограничения.

Точките за дадена подзадача се получават, когато преминат успешно всички тестове за нея.

Примери

Вход	Изход	Обяснение на примера
6 7 1 2 1 3 2 3 3 4 2 4 3 5 2 6	8	Служителите, които имат повече от 1 непосредствен началник са с номера 3 и 4. Непосредствените началници на 3 са 2 и 1 и ако ги подредим в ред 1, 2 тогава 1 е началник на 2. Непосредствените началници на 4 са 3 и 2 и ако ги подредим в ред 2, 3 тогава 2 е началник на 3. Съответно всички възможни наредби на служителите са: <ul style="list-style-type: none">• 1 2 3 6 4 5• 1 2 3 6 5 4• 1 2 3 4 5 6• 1 2 3 4 6 5• 1 2 3 5 6 4• 1 2 3 5 4 6• 1 2 6 3 4 5• 1 2 6 3 5 4
4 2 1 2 3 4	6	Тук всички възможни наредби на служителите са: <ul style="list-style-type: none">• 1 2 3 4• 1 3 2 4• 1 3 4 2• 3 1 2 4• 3 1 4 2• 3 4 1 2

ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА

Велико Търново, 19 - 21 април 2019 г.

Група А, 11 – 12 клас

Задача А2. БРОЯЧ

Тюринг си има машина. Тя е доста особена – състои се от една безкрайна лента, разбита на клетки, като във всяка клетка има записан символ. Машината има четяща и записваща глава, която във всеки момент е разположена над някоя клетка от лентата. Машината също така има и брояч, чиято стойност е цяло неотрицателно число. Тя винаги се намира в едно от K състояния (номерирани от 0 до $K - 1$). Машината работи на итерации, като на всяка итерация се изпълнява една инструкция, която е еднозначно определена от текущото ѝ състояние и от символа, върху който се намира главата. Всяка инструкция предизвиква изпълнението на следните четири действия, *именно в дадената по-долу последователност*:

1. Главата записва някакъв символ в клетката, над която се намира (символът може да е същият като този, който се намира в клетката).
2. Главата се премества една клетка наляво или надясно, или остава на същата позиция.
3. Машината преминава в друго състояние или остава в същото.
4. Машината увеличава стойността на брояча си с едно или не я променя.

Има и още една специална инструкция, която спира работата на машината. При нея отново може да се увеличи с едно стойността на брояча (но може и да не се промени).

Символите, които могат да се срещат в клетките на лентата са **0** и **1**. Точно в една клетка се намира символът **S**.

При стартиране, състоянието и броячът на машината са равни на 0, а лентата ѝ е пълна само с нули, с изключение на точно една клетка, в която е записан символ **S** и *върху която се намира главата*.

От там нататък главата може да пише само нули и единици по лентата, с изключение на случая, когато се намира върху клетката със символ **S** – тогава тя може да остави символа непроменен, като отново запише **S** в тази клетка (но може да запише и **0** или **1**, като в този случай **S** ще изчезне от лентата).

Тюринг иска да използва тази машина, за да си направи брояч. Той иска да състави набор от инструкции за нея по такъв начин, че машината да приключи изпълнението си (*а не да върви до безкрай*) и, когато това се случи, стойността на брояча ѝ да е точно N . Обаче той иска да може да прави това за много различни стойности на N и е непрактично за всяка да композира нови инструкции на ръка. Затова Тюринг би написал програма, която да върви на някой по-традиционен компютър и да генерира инструкциите за машината му по зададено N . Планът му, обаче, се е оказал неосъществим, защото по негово време такива компютри е нямало.

Той Ви моли за помощ. Напишете програма **counter**, която, по зададено N , генерира набор от инструкции за машината му. Може да използвате каквато стойност искате за K , но поради хардуерни съображения, *тя не трябва да надхвърля 20*.

Наборът от инструкции трябва да включва точно по една инструкция за всяка двойка (символ, състояние), подредени по следния начин:

$(S, 0), (S, 1), \dots, (S, K - 1), (0, 0), (0, 1), \dots, (0, K - 1), (1, 0), (1, 1), \dots, (1, K - 1)$

ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА

Велико Търново, 19 - 21 април 2019 г.

Група А, 11 – 12 клас

Форматът на основния тип инструкции е: **M(S) Символ Състояние**

Тук **M** е желаното движение (**L** – ляво, **S** – оставане на място, **R** – дясно). **S** означава увеличаване на брояча с едно. **Символ** е едно от **0, 1** или **S**. Това е символът, който ще се запише в клетката под главата *като първа стъпка от изпълнението на инструкцията*. **Състояние** е число от **0** до **K – 1**. Това е състоянието, в което ще премине машината след изпълнение на инструкцията.

Форматът на инструкцията за спиране на работата на машината е **H(S)**. Значението на **S** е същото както и при предния вид инструкции.

Скобите не участват в синтаксиса на инструкциите. Те показват, че наличието на S не е задължително. Броячат ще се увеличи, само ако го има. Вижте примера.

Вход

От първия ред на стандартния вход се въвежда едно цяло неотрицателно число **N**.

Изход

На първия ред на стандартния изход програмата трябва да извежда **K** – броя състояния на машината, които тя ще използва. На следващите $3 \times K$ реда трябва да се изведат инструкциите за всяка двойка (символ, състояние) в описания по-горе ред.

Ограничения

$$1 \leq K \leq 20$$

$$1 \leq N \leq 22\,000\,000$$

Машината трябва да приключи работа в рамките на 30 000 000 итерации.

Подзадачи и оценяване

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

Подзадача	Точки	Общо	Ограничение
1	10	10	20
2	5	15	60
3	10	25	120
4	5	30	399
5	10	40	500
6	5	45	1 000
7	5	50	5 000
8	10	60	130 000
9	5	65	260 000
10	5	70	600 000
11	5	75	1 000 000
12	5	80	3 000 000
13	5	85	5 000 000
14	5	90	10 000 000
15	5	95	20 000 000
16	5	100	22 000 000

ПРОЛЕТНИ СЪСТЕЗАНИЯ ПО ИНФОРМАТИКА

Велико Търново, 19 - 21 април 2019 г.

Група А, 11 – 12 клас

Локално тестване

Предоставен Ви е интерпретатор (файл **interpreter.cpp**), еквивалентен на този, който се ползва в системата (с разлики в това какво извежда), за да тествате програмата си локално. Можете да правите каквито си искате модификации по него.

Примерен тест

Вход	Изход
3	2 RC S 0 HC S 1 1 L 1 0 R 1 0 LC 0 1

Обяснение на примерния тест

Лента и глава	... 0 0 S 0 0 0 0 S 0 0 0 0 S 1 0 0 0 S 0 0 ...	–
Състояние	0	0	1	1	–
Брояч	0	1	1	2	3
Инструкция	RC S 0	S 1 1	LC 0 1	HC	–

Инструкциите **L 1 0** и **R 1 0** не се изпълняват нито веднъж.

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 $0 + 0 = 0$