

# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг  
София, 10 - 12 март 2023 г.  
Група С, 7 - 8 клас

## Задача С2. ОПТИМИЗАЦИЯ

Надяваме се, че помните езика `mono`. 😊

Минаха над две години, в които Дени изучаваше и придобиваше все повече умения за работа с командния език `mono`. Както при всеки млад език за програмиране, постоянно излизаха нови версии с подобрения. Като един от малкото ползватели и навлезли толкова навътре в езика, Дени беше повикана за помощ с реализацията на последната оптимизация на езика. Идеята на новата оптимизация била да се подобри ефективността на изпълнението на програмите, като в някои случаи се преизползва паметта, заета от някои променливи, при инициализацията на нови променливи.

Например, нека имаме програмата на езика `mono`,  
`set proba 2+3*5`  
показана вдясно. Можем да забележим, че променливата `proba`  
`set test (2+proba)/7`  
се използва последно на втори ред. Затова би било най-  
`set t test`  
ефективно да не отделяме нова памет за променлива `test`, а да  
`output test`  
преизползваме паметта, заета от променливата `proba`. Така,  
вместо две клетки памет за тези две променливи, ще имаме само  
една клетка памет.

В някои случаи такива оптимизации са много ефективни и дори могат да смъкнат значително времето и паметта на някои неефективно написани програми. За тази цел, Дени измисля идеята, че може да се пренареди кода на дадена програма, така че ресурсите да се използват по-ефективно. Ако разгледаме отново примерната програма, то, ако разменим трети и четвърти ред, бихме могли да преизползваме паметта на променливата `test` при инициализацията на `t`. Затова вашата задача е да напишете програма `optimize`, която по въведена програма на езика `mono` от  $N$  реда, да помага на Дени, като отговаря на въпроса дали може да се пренареди програмата по такъв начин, че да се преизползва паметта на всяко възможно място.

За улеснение, ще получавате кода на програмата по следния начин. На всеки ред ще бъде зададен един ред на програмата. Той ще е във формат:  $c\ k\ v_1\ v_2\ \dots\ v_k$  и съответно ще описва команда  $c$ , която се изпълнява над  $k$  променливи с имена  $v_1, v_2, \dots, v_k$ . Ако  $c = \text{set}$ , то тази команда ще инициализира променливата  $v_1$  като израз на променливите  $v_2, v_3, \dots, v_k$  (възможно е  $k = 1$ , т.е. този списък да е празен). Освен това, ако  $k \geq 2$ , знаете, че за паметта на  $v_1$ , езикът ще пробва да преизползва паметта на променливата  $v_2$ . Това преизползване е възможно само ако няма следващи команди, които използват променливата  $v_2$ . Вашата цел е така да пренаредите редовете на програмата, че всеки път, когато езикът се пробва да преизползва памет, да успее. Освен това нещо важно, което също трябва да спазвате, е да може програмата да се компилира след вашата промяна. Затова трябва всяка променлива да се използва след като вече е инициализирана. Така например, за примерната програма не може да сложите ред 2 преди ред 1, защото променливата `proba` няма да е инициализирана, когато използвате стойността ѝ. **Гарантирано е, че зададената начална програма може да се компилира и освен това всяка променлива, която се използва, е инициализирана точно веднъж.**

# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг  
София, 10 - 12 март 2023 г.  
Група С, 7 - 8 клас

## Вход

От първия ред на стандартния вход се въвежда едно естествено число  $N$  – броят на редовете в програмата. От всеки от следващите  $N$  реда се въвеждат низове с малки латински букви и едно число, разделени с по един интервал, задаващи съответния ред от кода на програмата във формат  $c_i, k_i, v_{i,1}, v_{i,2}, \dots, v_{i,k_i}$ , както беше описано по-рано.

## Изход

Ако е възможно да пренаредите програмата, така че да се преизползва паметта на всяко възможно място, отпечатайте последователност от номерата на редовете на пренаредената програма, която удовлетворява исканото условие. Ако има няколко правилни наредби на редовете на програмата, то отпечатайте най-малката лексикографски наредба (спрямо началните номера на редовете), която удовлетворява условието.

Ако не е възможно, то трябва да намерите най-дългата начална част на въведената програма, която има правилно пренареждане и да изведете най-малкото лексикографски правилно нейно пренареждане.

## Ограничения

- ♣  $1 \leq N \leq 2 \cdot 10^4$
- ♣  $1 \leq k_1 + k_2 + \dots + k_N \leq 10^5$
- ♣  $1 \leq |c_i|, |v_{i,j}| \leq 10$  ( $|s|$  е дължината на низа  $s$ )
- ♣ Ако  $c_i = \text{set}$ , то е гарантирано, че  $v_{i,j} \neq v_{i,1}$  за  $j > 1$

## Подзадачи и оценяване

Подзадача	Точки	$N$	$k_1 + k_2 + \dots + k_N$	Други ограничения
1	0	–	–	Примерите.
2	5	$\leq 2 \cdot 10^4$	$\leq 10^5$	Има правилно пренареждане на цялата програма и всяка променлива участва най-много на два реда.
3	13	$\leq 10$	$\leq 50$	Има правилно пренареждане на цялата програма.
4	16	$\leq 11$	$\leq 50$	–
5	45	$\leq 2 \cdot 10^4$	$\leq 10^5$	Има правилно пренареждане на цялата програма.
6	21	$\leq 2 \cdot 10^4$	$\leq 10^5$	–

Точките за подзадача се получават само ако се преминат успешно всички тестове, предвидени за нея. Един тест се брои за успешно преминат, ако е изведено правилно пренареждане. Ако не е изведена най-малката лексикографски наредба на някой тест, но всички тестове са преминати успешно в подзадачата, то ще получите само 70% от предвидените точки за подзадачата.

# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг  
София, 10 - 12 март 2023 г.  
Група С, 7 - 8 клас

## Примери

Вход	Изход	Обяснение
9 set 1 proba set 2 test proba set 2 t test output 1 test set 1 a set 3 b t a cmd 1 proba blabla 2 a b komanda 1 b	1 5 7 2 4 3 6 8 9	Пренаредената програма е: set 1 proba set 1 a cmd 1 proba set 2 test proba output 1 test set 2 t test set 3 b t a blabla 2 a b komanda 1 b Чрез това правилно нареждане се осигурява използването на: <ul style="list-style-type: none"> <li>• proba при инициализацията на променливата test (нов ред 4)</li> <li>• test при инициализацията на променливата t (нов ред 6)</li> <li>• t при инициализацията на променливата b (нов ред 7)</li> </ul>
4 set 1 x set 1 y set 3 a x y set 3 b y x	1 2 3	Няма правилна наредба на цялата програма. Променливата a ще иска да използва променливата x, а променливата b – y. За да се използва x за променливата a, то трябва ред 4, в който участва x, да е преди ред 3. Аналогично, за да се използва y за променливата b, трябва ред 3 да е преди ред 4. Тези две условия са невъзможни заедно. От друга страна първите три реда имат правилна наредба. Има две възможни – 1, 2, 3, както и 2, 1, 3. Първата е най-малката лексикографски.