

Задача 5 от конкурса на PC Magazine & Musala Soft

СПИСЪЦИ

Зайо много обича реда. Обича всичко да е организирано и подредено. На това се дължи и навикът му да прави списъци. Списък за рождените дни на своите приятели, списък с нещата, които трябва да свърши утре, другата седмица или другия месец, кога трябва да яде мед с Пух или да обсъди важни теми с Бухал като "доматите плод ли са или зеленчук".

На Зайо, често се налага да преработва част от списъците, да обединява или разделя други. Освен това възниква и въпросът, кое е най-важното или маловажно действие в някой списък.

Кристофър Робин се интересува не само от компютри, но и от проблемите на приятелите си. Той реши да направи програма, която да помогне на Зайо за организацията на неговите списъци, както и да изпробва своите знания в областта на програмирането. Ние решихме да изпробваме вашите.

Напишете програма **LISTS**, която поддържа работа със списъци както е описано по-долу. Списъците съдържат числа наредени последователно. Всяко число е в интервала $[1, 10^9]$. Позициите на числата се номерират от 1 до K , където K е големината на съответния списък. Всеки списък се идентифицира с уникално число в интервала $[1, 10^6]$. Програмата трябва да изпълнява следните операции, записани по една на ред във входния файл:

- `create <list_id> <value>` - създава нов списък, идентифициран с номер `<list_id>`, съдържащ единствен елемент `<value>`. В момента на изпълнението на командата, на стойността `<list_id>` няма да отговаря съществуващ списък.
- `merge <list_id1> <list_id2>` - обединява списъците отговарящи на `<list_id1>` и `<list_id2>` като след последния елемент на `<list_id1>` се поставя първия елемент на `<list_id2>`. Новият списък е идентифициран с `<list_id1>`, а `<list_id2>` става невалидна стойност до евентуално следващо дефиниране.
- `split <list_id> <position> <new_list_id>` - разделя списъка `<list_id>` на два списъка, съдържащи съответно числата на позиции от 1 до $(\text{<position>} - 1)$ и от `<position>` до K , където K е големината на списъка `<list_id>`. Списъкът с числата от 1 до $(\text{<position>} - 1)$ запазва номера на първоначалния списък - `<list_id>`. Списъкът с числата от `<position>` до K получава идентификационно число `<new_list_id>`, което няма да отговаря на никой списък към момент на изпълнение на операцията. `<position>` ще бъде число по-голямо от 1 и по-малко или равно от K .
- `reverse <list_id>` - обръща подредбата на елементите в списъка `<list_id>` (първият става последен, вторият става предпоследен и т.н.)
- `min <list_id>` - изписва в изходния файл стойността на най-малкия елемент във списъка с номер `<list_id>`.
- `max <list_id>` - изписва в изходния файл стойността на най-големия елемент във списъка с номер `<list_id>`.
- `stop` - прекратява изпълнението на програмата.

Входните данни са записани в **LISTS.INP**. Всеки ред съдържа по една команда както е описано по-горе. Последният ред на файла съдържа командата **stop**.

Изходният файл **LISTS.OUT** трябва да съдържа по един на ред отговорите, изисквани от командите **min** и **max**, в реда на тяхното появяване във входния файл.

ПРИМЕР

LISTS.INP

```
create 1 4
create 2 3
merge 1 2
min 1
create 3 10
create 4 7
merge 3 4
```

LISTS.OUT

```
3
10
4
3
```

max 3
reverse 1
merge 3 1
split 3 4 2
max 2
min 3
stop