

# Chandelier

## Условие

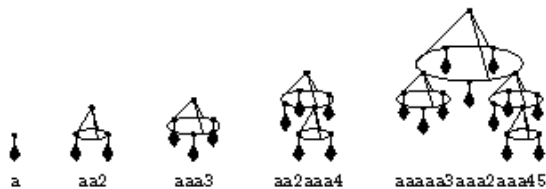
Компания Ламп-о-матик сглобява много големи полилеи. Един полилей се състои от няколко нива. На първото ниво кристални окраснения се закачат за пръстени. Сглобени пръстени и нови окраснения се закачат на пръстени от следващото ниво и т.н. В най-горния край има единствен голям пръстен – завършения полилей с по малки пръстени и окраснения висящи от него.

Специален робот сглобява полилеите. Той има запас от украшения и празни пръстени, както и стек в който да пази елементите на полилея по време на сглобяването. В началото стека е празен. Робота изпълнява списък от команди за да сглоби полилей.

При команда "а" робота взема ново кристално украшение и го слага на върха на стека. При команда от "1" до "9" робота взема съответния брой елементи от върха на стека и ги закрепя последователно към нов пръстен, след което този пръстен бива поставен на върха на стека. В края на програмата има единствен елемент в стека – завършения полилей.

За съжаление, за някои програми се оказва, че по време на изпълнение стека трябва да съхранява твърде много елементи в даден момент. Вашата задача е да оптимизирате дадена програма, така че общата структура на съответния полилей си остава същия, но максималния брой елементи в стека по време на изпълнение да е минимален. Украшение или сложно сглобен многопластов пръстен се броят за един елемент в стека.

Два полилея имат еднаква структура ако съответни пръстени съдържат същия брой елементи в същия ред. Тъй като пръстените са кръгли няма значение какъв елемент стои на върха на стека, когато роботът получи команда за сглобяване на нов пръстен, а относителния ред на елементите от стека които участват в операцията. Например, ако роботът получи команда "4" когато елементите  $\langle i_1, i_2, i_3, i_4 \rangle$  са на върха на стека в този ред ( $i_1$  е най-отгоре), тогава същия пръстен би се получил ако елементите на върха на стека бяха подредени по следните начини:  $\langle i_4, i_1, i_2, i_3 \rangle$ , или  $\langle i_3, i_4, i_1, i_2 \rangle$ , или  $\langle i_2, i_3, i_4, i_1 \rangle$ .



## Вход

Използва се стандартния вход. Входа съдържа единствен ред с валидна програма за работа. Програмата съдържа най-много 10000 символа.

## Изход

Използва се стандартния изход. На първия ред на изхода напишете минималния търсен капацитет на стека (колко елемента може да съдържа) за да се сглоби полилея. На следващия ред напишете програма за работа която ползва стек със този капацитет и генерира същия полилей.

Примерен Вход  
aaaaa3aaa2aaa45

Примерен Изход  
6

Разяснение за примерния вход/изход  
Оптималната програма има вида: aaa3aaa2aaa4aa5

## Ограничения:

Време: 0.1s

Памет: 128MB

## Заглавен коментар (header)

за C: /* TASK:chandelier LANG:C */	за C++: /* TASK:chandelier LANG:C++ */	за Pascal: { TASK:chandelier LANG:Pascal }
--	--	--