

Задача 2. Sgame

По случай третата година откакто Международния турнир в Шумен стана IATI, организаторите са измислили нова интересна игра. Дени решила да се включи. Тя отишла при таблет, където е заредена играта. Там била написана някаква дума w от малки латински букви с дължина N . Играта се провеждала в Q рунда и всеки протичал по следния начин: играчът си намисля някаква дължина m и се търси най-често срещан подниз s с тази или по-голяма дължина. Точките за рунд се определят от броя срещания на s в w , нека ги означим с cnt . Само че, за да е по-интересно, играчът задава и дължина $k \geq m$ с цел следното ограничение – намереният подниз s не трябва да има дължина по-голяма от k . Също не трябва да може да се продължи с някакви букви отляво или отдясно до дължина по-голяма от k , така че полученият низ да се среща cnt или повече пъти. Казано по друг начин, дължината на s трябва да е от m до k и за всеки възможен низове u и v , така че слепенният низ usv , с дължина поне k , да се среща в w по-малко от cnt на брой пъти. Ако няма възможен подниз s , тогава точките за рунда са 0. Вие сте от организационния комитет на състезанието и трябва да осигурите максимално удоволствие на участниците. Дени е много нетърпелива, затова иска максимално бързо да разбира резултата от зададените от нея рундове.

Задача.

Напишете програма **Sgame**, която да работи на таблетчетата, така че нашата героиня да е удовлетворена. Тя трябва да съдържа функциите `start()` и `round()`, които ще се компилират с програмата на журито и за всеки рунд ще се връща броят точки – максималния брой срещания на търсения подниз или 0 (ако такъв не съществува за текущия рунд). Функцията `start()` ще получи думата, която се зарежда на таблета. След което програмата на журито ще извика Q пъти функцията `round()` с две числа m и k – съответно ограниченията за търсения подниз.

Детайли по реализацията.

Функцията `start()` трябва да има следния формат:

```
void start (char w[]);
```

Тя се вика веднъж от програмата на журито с параметър w – думата, която се зарежда в таблета. Другата функция `round()` за обработване на рунд трябва да има следния формат:

```
int round (int m, int k);
```

Параметрите са числата, които са ограниченията за търсения подниз. Тук са изпълнени следните условия: $1 \leq m \leq k \leq N$, където N е дължината на думата w . Когато приключи работа за текущия рунд, функцията връща едно число – броя точки за рунда.

Вие трябва да предадете към системата файл **Sgame.cpp**, който съдържа функциите `start()` и `round()`. Той може да съдържа и друг код, и функции, необходими за работата Ви, но не трябва да съдържа главната функция `main()`.

В началото си Вашият файл трябва да съдържа: `#include "Sgame.h"`.

Ограничения. $1 \leq N \leq 5 \cdot 10^5$; $1 \leq Q \leq 10^6$

Подзадачи.

Подзадача	Точки	N	Q	Допълнителни ограничения
1	5	$\leq 10^3$	≤ 10	$k = N$
2	7	$\leq 5 \cdot 10^3$	$\leq 10^3$	$k = N$
3	12	$\leq 10^2$	$\leq 10^3$	Няма
4	18	$\leq 10^5$	$\leq 10^5$	$k = N$
5	24	$\leq 5 \cdot 10^5$	$\leq 10^6$	Гледаме за продължение само вляво на подниз.
6	34	$\leq 5 \cdot 10^5$	$\leq 10^6$	Няма

Отговори на тестовете за пета подзадача, които спазват условията в оригиналното условие на задачата също се зачитат за верни!

Примерна комуникация с програмата на журито.

Нека думата w , заредена на таблета е $abcabcabcdede$ и разглеждаме 3 рунда. Програмата на журито ще извика Вашата функция по този начин: $start(w)$. След като тя приключи работа, програмата на журито ще извика три пъти Вашата функция $round$:

Извикване на $round$	Верен отговор	Обяснение
$round(1,2)$	4	Поднизът с най-много срещания е a . Понеже нито отляво, нито отдясно може да се продължи, за да направим низ с 4 или повече срещания, това е и отговорът.
$round(2,3)$	3	Подниз с най-много срещания е ab . Той може да бъде продължен отдясно най-много като abc също с три срещания. Това е допустимо, защото $k = 3$. Друг възможен отговор е bc .
$round(2,2)$	2	Тук понеже k е само 2, то нито ab , нито bc са валидни отговори. Затова поднизът с най-много срещания е de .

Локално тестване.

За локално тестване са предоставени файловете **Lgrader.cpp** и **Sgame.h**. Сложете ги в същата папка, в която е Вашият файл **Sgame.cpp** и компилирайте **Lgrader.cpp**. Така ще получите програма, с която ще проверите верността на функциите Ви. Програмата ще изисква от стандартния вход следната последователност от данни:

- на първия ред: едно положително число – дължината N на думата w .
- на втория ред: думата w
- на третия ред: едно положително число – броя на рундовете Q .
- на последните Q реда: по две положителни числа – съответно дължините m и k за съответния рунд.

На изход ще получите броя точки, които намирате за всеки рунд.

Предаване на тестове към системата.

Можете да предавате тестове към системата. Форматът на входните данни е същият като за локалния грейдър. На изход ще получите броя точки, които намирате за всеки рунд.