

Задача 1. Чешма 0.1 с.  256 MB

Вие се казвате Боби и забравихте да си купите вода. На разположение са Ви чисто нова смарт чешма и чаша с размер 10^5 литра. Трябва да намерите най-бързия начин да се napiете с N литра вода.

Задача

Напишете програма **cheshma**, съдържаща функция `solve`, която ще се компилира с програмата на журито и ще комуникира с нея, извършвайки действия от долупосочените видове. В края на изпълнението си трябва Боби да е изпил N литра чиста вода.

Детайли по имплементацията

Тази задача е псевдоинтерактивна. Функцията `void solve(int n)`, която трябва да напишете, ще бъде извикана само веднъж от програмата на журито и като аргумент ще получи цялото число N . За комуникация с чешмата и чашата на Боби Ви се предоставят следните функции:

```
void open();  
void close();  
void setCountry(std::string s);  
void drinkCup(int m);  
void store(int m);  
void drink(int m);  
void pay();
```

При всяко извикване на функцията `open` чешмата ще върви и Вие ще можете да събирате вода от нея. Подобно, функцията `close` затваря чешмата. Функцията `setCountry` изтегля VPN (Virtual Private Network) на смарт чешмата и избира дадената държава s , която е написана с малки латински букви и името на държавата е на английски. Функцията `drinkCup` Ви кара да изпиете m литра от чашата, а ако няма толкова в нея се взима всичко в чашата. Функцията `store` излива m литра от чешмата в чашата. Функцията `drink` Ви кара да изпиете m литра директно от чешмата. Функцията `pay` Ви плаща сметките, но не прави нищо, поради Вашата липса на парични средства. **Имайте предвид, че чешмата е по-бърза дори от линейна на някои места!**

Вашата програма `cheshma.cpp` трябва да имплементира функцията `solve`. Тя може да съдържа и друг код, и функции, необходими за работата Ви, но не трябва да съдържа главната функция `main`. Също така, не трябва да четете от стандартния вход или да отпечатвате на стандартния изход. Програмата Ви трябва да включва хедър файла `cheshma.h` чрез указание към предпроцесора:

```
#include "cheshma.h"
```

Ограничения

- $0 \leq N \leq 10^5$

Подзадачи

Нека сумата на m върху заявките е S .

Подзадача	Точки	Ограничения
1	5	$0 \leq N \leq 0$
2	10	$0 \leq N \leq 20$
3	20	$0 \leq S \leq 2N$
4	25	$0 \leq S \leq N$
5	25	$0 \leq S \leq \frac{2}{3}N$
6	15	$0 \leq S \leq \frac{1}{3}N$

Точките за дадена подзадача се получават само ако се преминат успешно всички тестове, предвидени за нея.

Примерна комуникация

Функция на участника	Програма на журито
	<code>solve(6)</code>
<code>open()</code>	Нищо.
<code>store(4)</code>	Нищо.
<code>pay()</code>	Нищо.
<code>setCountry("germany")</code>	Нищо.
<code>drink(2)</code>	Нищо.
<code>drinkCup(4)</code>	Нищо.
Функцията приключва.	

Пояснение: В примера изсипваме 4л в чашата, прием 2л от чешмата, прием 4л от чашата и сме напiti. Тук $S = 10$.

Оценяване

Ако получавате резултат WA, водата е мръсна или не сте изпълнили задачата. Ако получавате RE, ВиК са спрели водата, имате неплатени сметки, опитали сте си да платите сметките или имате грешка по време на изпълнение на вашия алгоритъм. При резултат ML сте наводнили стаята (напълнили сте повече от 10^5 литра вода в чашата). Ако получавате TL, използвали сте прекалено много заявки или пиенето Ви е отнело толкова време, че сте умрели от жажда.

Локално тестване

Предоставен Ви е файлът `Lgrader.cpp`, който може да компилирате заедно с вашата програма, за да я тествате. При стартиране програмата ще чете от стандартния вход стойността на N . След това ще се отпечата комуникацията, която се извършва. Може да модифицирате предоставения файл, както искате.