



Комбо

Вы играете в видеоигру. Игровой контроллер содержит 4 кнопки: A, B, X и Y. В этой игре вы можете получать монеты за комбо-действия. Вы можете совершить комбо-действие, нажимая кнопки в определенной последовательности.

В этой игре есть секретная последовательность нажатий кнопок, которую можно представить строкой S , каждый символ которой — один из 4 символов кнопок. Строка S неизвестна, но известна ее длина N .

Также известно, что первый символ строки S никогда в ней больше не встречается. Например, S может быть равна "ABXYU" или "XYUAA", но не может быть равна "AAAAA" или "VXUVX".

Для совершения комбо-действия вы можете последовательно нажать не более $4N$ кнопок. Пусть строка p представляет последовательность нажатых кнопок. Количество монет, которое вы получите за это действие, вычисляется как длина наибольшего префикса строки S , который также является подстрокой p . Подстрока строки t — это непрерывная (возможно пустая) последовательность символов строки t . Префикс строки t — это пустая или содержащая первый символ подстрока строки t .

Например, если S равна "ABXYU" и p равна "XXUYABYABXAY", то будет получено 3 монеты, потому что "ABX" это наибольший префикс S , который также является подстрокой p .

Требуется определить секретную строку S , используя комбо-действия.

Детали реализации

Вам следует реализовать одну функцию:

```
string guess_sequence(int N)
```

- N : длина строки S .
- Функция вызывается ровно один раз для каждого теста.
- Функция должна вернуть строку S .

Ваша программа может вызывать следующую функцию:

```
int press(string p)
```

- p : последовательность нажатых кнопок.
- p должна быть строкой длины от 0 до $4N$ символов, включительно. Каждый символ строки p должен быть одним из A, B, X или Y.
- Вы можете вызывать эту функцию не более 8 000 раз для каждого теста.
- Функция возвращает количество монет, которое будет получено при нажатии последовательности кнопок, представленной строкой p .

Если какие-либо из вышеописанных условий не выполняются, ваша программа получит вердикт **Wrong Answer**. В противном случае, ваша программа получит вердикт **Accepted**, и ваш балл определяется количеством вызовов функции `press` (смотрите раздел "Подзадачи").

Пример

Пусть S равна "ABXY". Проверяющий модуль (`grader`) вызывает функцию `guess_sequence(5)`. Пример взаимодействия описан ниже.

Вызов	Результат
<code>press("XXYYABYABXAY")</code>	3
<code>press("ABXY")</code>	5
<code>press("ABXYABXY")</code>	5
<code>press("")</code>	0
<code>press("X")</code>	0
<code>press("BXY")</code>	0
<code>press("YYXBA")</code>	1
<code>press("AY")</code>	1

Для первого вызова функции `press`, "ABX" встречается в "XXYYABYABXAY" как подстрока, а "ABXY" не встречается, поэтому возвращаемое значение 3.

Для третьего вызова функции `press`, "ABXY" целиком встречается в "ABXYABXY" как подстрока, поэтому возвращаемое значение 5.

Для шестого вызова функции `press`, никакой префикс "ABXY", кроме пустой строки, в "BXY" не встречается как подстрока, поэтому возвращаемое значение 0.

Наконец, `guess_sequence(5)` должна вернуть строку "ABXY".

Файл `sample-01-in.txt` в прикрепленном архиве соответствует этому примеру.

Ограничения

- $1 \leq N \leq 2000$.
- Каждый символ строки S является одним из A, B, X или Y.
- Первый символ строки S никогда больше не встречается в S .

В этой задаче проверяющий модуль НЕ ЯВЛЯЕТСЯ адаптивным. Это означает, что строка S зафиксирована на момент запуска проверяющего модуля и не зависит от выполненных запросов в вашем решении.

Подзадачи

1. (5 баллов) $N = 3$.
2. (95 баллов) Без дополнительных ограничений. Для этой подзадачи ваш результат за каждый тест вычисляется следующим образом. Пусть количество вызовов функции `press` равно q .
 - Если $q \leq N + 2$, ваш результат 95.
 - Если $N + 2 < q \leq N + 10$, ваш результат $95 - 3(q - N - 2)$.
 - Если $N + 10 < q \leq 2N + 1$, ваш результат 25.
 - Если $\max\{N + 10, 2N + 1\} < q \leq 4N$, ваш результат 5.
 - Иначе, ваш результат 0.

Обратите внимание, что результат за каждую подзадачу равен минимуму среди результатов на тестах этой подзадачи.

Пример проверяющего модуля

Пример проверяющего модуля читает входные данные в следующем формате:

- Строка 1: S

Если ваша программа получает вердикт **Accepted**, проверяющий модуль печатает Accepted: q , где q равно количеству вызовов функции `press`.

Если ваша программа получает вердикт **Wrong Answer**, проверяющий модуль печатает Wrong Answer: MSG. Значение MSG может быть:

- `invalid press`: Значение p , переданное в функцию `press`, ошибочно. А именно, длина строки p не в промежутке от 0 до $4N$, включительно, или какой-то из символов строки p не является ни одним из A, B, X или Y.
- `too many moves`: Функция `press` вызвана более 8000 раз.
- `wrong guess`: Возвращаемое функцией `guess_sequence` значение не совпадает со строкой S .