



Combo

Estás jugando un videojuego de acción. El control del juego tiene 4 botones, A, B, X, y Y. En este juego, puedes obtener monedas con combos. Un combo se logra presionando botones en secuencia.

Este juego tiene una secuencia secreta de botones que puede ser representada como una cadena S formada de los 4 caracteres de arriba. No sabes cuál es la cadena S , pero conoces su longitud N .

También sabes que el primer caracter de S nunca aparece más de una vez. Por ejemplo, S puede ser "ABXYY" o "XYAA", pero no "AAAAA" o "BXYBX".

Puedes presionar una secuencia de hasta $4N$ botones para un combo. Sea p la cadena que representa la secuencia de los botones que presionaste. El número de monedas que obtienes para esta movida se calcula como la longitud del prefijo más largo de S que también es una sub-cadena de p . Una sub-cadena de una cadena t es una secuencia contigua (posiblemente vacía) de caracteres dentro de t . Un prefijo de t es una sub-cadena de t que está vacía o contiene al primer caracter de t .

Por ejemplo, si S es "ABXYY" y p es "XXYYABYABXAY", obtienes 3 monedas porque "ABX" es el prefijo más largo de S que también es una sub-cadena de p .

Tu tarea es determinar la cadena secreta S utilizando la menor cantidad de movidas de combo.

Detalles de implementación

Debes implementar la siguiente función:

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

- N : la longitud de la cadena S .
- Esta función se llama exactamente una vez por cada caso de prueba.
- Esta función debe devolver la cadena secreta S .

Tu programa puede llamar a la siguiente función:

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

- p : una secuencia de botones que presionas.
- p debe ser una cadena cuya longitud oscile entre 0 y $4N$, inclusive. Cada caracter de p debe ser A, B, X o Y.
- No puedes llamar a esta función más de 8 000 veces por cada caso de prueba.
- Esta función retorna el número de monedas que obtienes cuando presionas la secuencia de botones representada por p .

Si al menos una de las condiciones de arriba no es satisfecha, tu programa obtendrá un veredicto de **Wrong Answer**. De lo contrario, tu programa obtendrá el veredicto de **Accepted** y tu puntuación será calculada por el número de llamadas a `press` (lee la sección de Sub-tareas).

Ejemplo

Sea S "ABXY". El grader llama a `guess_sequence(5)`. Un ejemplo de comunicación se muestra debajo.

Llamada	Retorno
<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("BXYY")</code>	0
<code>press("YYXBA")</code>	1
<code>press("AY")</code>	1

En la primera llamada a `press`, "ABX" aparece en "XXYYABYABXAY" como una sub-cadena, pero "ABXY" no, por lo que el resultado es 3.

En la tercera llamada a `press`, "ABXY" aparece en "ABXYABXY" como una sub-cadena, por lo que se retorna 5.

En la sexta llamada a `press`, no hay ningún prefijo de "ABXY", pero la cadena vacía aparece en "BXYY" como una sub-cadena, por lo que se retorna 0.

Finalmente, `guess_sequence(5)` debe retornar "ABXY".

El archivo `sample-01-in.txt`, ubicado en el paquete comprimido adjunto, corresponde con este ejemplo.

Restricciones

- $1 \leq N \leq 2000$
- Cada caracter de la cadena S es A, B, X, o Y.
- El primer caracter de S no aparece nuevamente en S .

El grader para este problema NO es adaptivo. Esto quiere decir que S es fija al inicio de la ejecución del grader y ella no depende de las consultas efectuadas por tu solución.

Sub-tareas

1. (5 puntos) $N = 3$
2. (95 puntos) No hay restricciones adicionales. Para esta sub-tarea, tu puntuación para cada caso de prueba se calcula de la siguiente manera. Sea q el número de llamadas a la función `press`.
 - Si $q \leq N + 2$, tu puntuación es 95.
 - Si $N + 2 < q \leq N + 10$, tu puntuación es $95 - 3(q - N - 2)$.
 - Si $N + 10 < q \leq 2N + 1$, tu puntuación es 25.
 - Si $\max\{N + 10, 2N + 1\} < q \leq 4N$, tu puntuación es 5.
 - De lo contrario, tu puntuación es 0.

Nota que tu puntuación para cada sub-tarea es el mínimo de las puntuaciones para los casos de prueba dentro de la sub-tarea.

Grader de ejemplo

El grader de ejemplo lee la entrada en el siguiente formato:

- línea 1: S

Si tu programa se juzga como **Accepted**, el grader de ejemplo imprime `Accepted: q` siendo q el número de llamadas a la función `press`.

Si tu programa se juzga como **Wrong Answer**, este imprime `Wrong Answer: MSG`. El significado de `MSG` es lo siguiente:

- `invalid press`: Un valor de p dado a `press` es inválido. Específicamente, la longitud de p no está entre 0 y $4N$, inclusive, o algún caracter de p no es A, B, X, o Y.
- `too many moves`: La función `press` fue llamada más de 8000 veces.
- `wrong guess`: El valor de retorno de `guess_sequence` no es S .