



Combo

Usted está jugando un videojuego de acción. El control tiene 4 botones: A, B, X e Y. En este juego, puede obtener monedas realizando combos. Puede hacer un combo presionando una secuencia de botones.

Este juego tiene una secuencia secreta de botones que puede ser representada como una cadena S de esos 4 caracteres. Usted no conoce la cadena S , pero conoce su largo N .

Usted también sabe que el primer caracter de S nunca se repite en la misma cadena. Por ejemplo, S puede ser "ABXYY" o "XYAA", pero no puede ser "AAAAA" ni "BXYBX".

Un combo puede ser formado por una secuencia de hasta $4N$ botones. Sea p la cadena que representa la secuencia de botones que presionó. El número de monedas que obtiene por este combo se calcula como la longitud del prefijo más largo de S que también es una subcadena de p . Una subcadena de una cadena t es una secuencia contigua (posiblemente vacía) de caracteres dentro de t . Un prefijo de t es una subcadena de t que es vacía o que contiene al primer caracter de t .

Por ejemplo, si S es "ABXYY" y p es "XXYYABYABXAY", obtendrá 3 monedas ya que "ABX" es el prefijo más largo de S que también es subcadena de p .

Su tarea es determinar la cadena secreta S utilizando la menor cantidad de combos posible.

Detalles de implementación

Debe implementar la siguiente función:

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

- N : el largo de la cadena S
- Esta función es invocada exactamente una vez para cada caso de prueba.
- Esta función debe retornar la cadena S .

Su programa puede llamar a la siguiente función:

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

- p : una secuencia de botones que usted presiona
- p debe ser una cadena de largo entre 0 y $4N$, inclusive. Cada caracter de p debe ser A, B, X ó Y.
- No puede llamar a esta función más de 8 000 veces para cada caso de prueba.
- Esta función devuelve el número de monedas que obtiene cuando presiona la secuencia de botones representada por p .

Si alguna de las condiciones anteriores no es satisfecha, su programa será juzgado con **Wrong Answer**. De lo contrario, su programa será juzgado como **Aceptado** y su puntuación se calculará en relación al número de llamadas a `press` (ver las subtareas).

Ejemplo

Sea S la secuencia "ABXY". El calificador llama `guess_sequence(5)`. Un ejemplo de comunicación es el siguiente:

Llamada	Valor de 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

Para la primera llamada a `press`, "ABX" aparece en "XXYYABYABXAY" como una subcadena pero "ABXY" no, por lo que se devuelve 3.

Para la tercer llamada a `press`, "ABXY" misma aparece en "ABXYABXY" como una subcadena, así que se devuelve 5.

Para la sexta llamada a `press`, ningún prefijo de "ABXY" más que la cadena vacía aparece en "BXYY" como una subcadena, por lo que se devuelve 0.

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

El archivo `sample-01-in.txt` en el paquete zip adjunto corresponde a este ejemplo.

Restricciones

- $1 \leq N \leq 2000$
- Cada caracter de la cadena S es A, B, X ó Y.
- El primer caracter de S nunca vuelve a aparecer en S .

En este problema, el calificador NO es adaptivo. Esto significa que S es fija al inicio de la ejecución del calificador y no depende de las preguntas realizadas por su solución.

Subtareas

1. (5 points) $N = 3$
2. (95 points) Sin restricciones adicionales. Para esta subtarea, su puntuación para cada caso de prueba es calculado de la siguiente manera. Sea q el número de llamadas a `press`.
 - Si $q \leq N + 2$, su puntuación es 95.
 - Si $N + 2 < q \leq N + 10$, su puntuación es $95 - 3(q - N - 2)$.
 - Si $N + 10 < q \leq 2N + 1$, su puntuación es 25.
 - Si $\max\{N + 10, 2N + 1\} < q \leq 4N$, su puntuación es 5.
 - De lo contrario, su puntuación 0.

Note que su puntuación para cada subtarea es el mínimo de las puntuaciones para los casos de prueba en la subtarea.

Calificador ejemplo

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

- línea 1: S

Si su programa es juzgado como **Accepted**, el calificador de ejemplo imprimirá `Accepted: q`, siendo q el número de llamadas a la función `press`.

Si su programa es juzgado como **Wrong Answer**, imprime `Wrong Answer: MSG`. El significado de `MSG` es el siguiente:

- `invalid press`: Un valor de p dado a la función `press` es inválido. Es decir, el largo de p no está entre 0 y $4N$, inclusive, o algún caracter de p no es A, B, X ó 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 .