



Combo

Estás jugando un videojuego. El control tiene 4 botones: A, B, X, y Y. En este juego obtienes monedas haciendo *combos*. Puedes hacer un *combo* presionando los botones en una secuencia específica.

El juego tiene una secuencia secreta de botones, que se representa como una cadena S de esos 4 caracteres. Desconoces la cadena S , pero sabes que su longitud es N .

También sabes que el primer carácter de S nunca volverá a aparecer en S . Por ejemplo, S puede ser "ABXYY" o "XYyAA", pero no puede ser "AAAAA" o "BXYBX".

Puedes presionar una secuencia de hasta $4N$ botones para hacer un *combo*. Definimos p como la cadena que representa la secuencia de botones que presionaste. La cantidad de monedas que obtienes por este *combo* se calcula como la longitud del prefijo más largo de S que también es una subcadena de p . La subcadena de una cadena t se define como una secuencia contigua (puede ser vacía) de caracteres dentro de t . El prefijo de t es una subcadena de t que es vacía o inicia en el primer carácter de t .

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

Tu tarea es encontrar la cadena secreta S usando la menor cantidad de *combos*.

Detalles de implementación

Debes implementar la función:

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

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

Tu programa puede llamar la función:

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

- p : la secuencia de botones que presionaste.
- p debe ser una cadena de longitud entre 0 y $4N$, inclusive. Cada carácter de p

debe ser A, B, X, o Y.

- No puedes llamar esta función más de 8 000 veces para cada caso de prueba.
- Esta función devuelve el número de monedas que obtienes cuando presionas la secuencia de botones representada por p .

Si alguna de estas consideraciones no se cumplen tu programa será evaluado como **Wrong Answer**. De lo contrario tu programa será evaluado como **Accepted** y tu calificación se calculará de acuerdo al número de llamadas a la función `press` (ver subtareas).

Ejemplo

Sea S "ABXY". El evaluador llama a la función `guess_sequence(5)`. Un ejemplo de la comunicación se muestra a continuación.

Llamada	Valor devuelto
<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

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

En la tercera llamada a `press`, "ABXY" aparece en "ABXYABXY" como una subcadena y se regresa 5.

En la sexta llamada a `press`, no hay prefijo de "ABXY" pero la cadena vacía es subcadena de "BXY", por lo que se devuelve 0.

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

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

Consideraciones

- $1 \leq N \leq 2000$
- Cada carácter de la cadena S es A, B, X, o Y.

- El primer carácter de S nunca vuelve a aparecer en S .

En este problema el evaluador NO es adaptativo. Esto significa que S es fijo desde el principio de la ejecución y no depende de las consultas realizadas por tu solución.

Subtareas

1. (5 puntos) $N = 3$
2. (95 puntos) Sin restricciones adicionales. Para esta subtarea tu puntaje se calcula de la siguiente manera. Sea q el número de llamadas a `press`.
 - Si $q \leq N + 2$, tu puntaje es 95.
 - Si $N + 2 < q \leq N + 10$, tu puntaje es $95 - 3(q - N - 2)$.
 - Si $N + 10 < q \leq 2N + 1$, tu puntaje es 25.
 - Si $\max\{N + 10, 2N + 1\} < q \leq 4N$, tu puntaje es 5.
 - En cualquier otro caso tu puntaje es 0.

El puntaje de cada subtarea será el puntaje mínimo de todos los casos de prueba de esa subtarea.

Evaluador de prueba

El evaluador de prueba lee la entrada en el siguiente formato:

- línea 1: S

Si tu programa es evaluado como **Accepted**, el evaluador de prueba imprime `Accepted: q`. Siendo q el número de llamadas a la función `press`.

Si tu programa es evaluado como **Wrong Answer**, imprime `Wrong Answer: mensaje`. Estos son los posibles mensajes:

- `invalid press`: el valor p en la función `press` es invalido. Ya sea porque la longitud de p no se encuentra entre 0 y $4N$, inclusive, o porque p contiene caracteres distintos a A, B, X, y Y.
- `too many moves`: La función `press` fue llamada mas de 8 000 veces.
- `wrong guess`: `guess_sequence` no devolvió S .