



Combo

Estás jugando un video juego. El control del juego tiene 4 botones, A, B, X e Y. En este juego, puedes hacer combos para conseguir monedas. Para hacer un combo debes presionar una secuencia de botones.

Este juego contiene una secuencia secreta de botones, que puede ser representada mediante una cadena S de esos 4 caracteres. No conoces la cadena S , pero sí conoces su longitud N .

También sabes que el primer caracter de S nunca vuelve a aparecer en la cadena. Por ejemplo, S puede ser "ABXXY" o "XYYA", pero no puede ser "AAAAAA" o "BXYBX".

Puedes presionar una secuencia de a lo más $4N$ botones para hacer un combo. Sea p 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 subcadena de p . Una subcadena de una cadena t es una secuencia contigua (posiblemente vacía) de caracteres en t . Un prefijo de una cadena t es una subcadena de t que contiene a su primer caracter, o es vacía.

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

Debes determinar la cadena secreta S realizando pocos combos.

Detalles de implementación

Debes implementar la siguiente función:

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

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

Puedes llamar a la siguiente función:

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

- p : una secuencia de botones a presionar.
- p debe ser una cadena de longitud 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 en cada caso de prueba.
- Esta función retorna la cantidad de monedas que se obtienen al presionar la secuencia de botones representada por p .

Si alguna de las condiciones anteriores no se cumple, el programa será evaluado como **Wrong Answer**. De lo contrario, el programa será evaluado como **Accepted** y obtendrá un puntaje calculado según el número de llamadas a la función `press` (ver Subtareas).

Ejemplo

Sea S la cadena "ABXYY". El evaluador llama a `guess_sequence(5)`. A continuación se muestra un ejemplo de posibles llamadas a `press` y su retorno:

Llamada	Retorno
<code>press("XXYYABYABXAY")</code>	3
<code>press("ABXYY")</code>	5
<code>press("ABXYYABXYY")</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 subcadena pero "ABXY" no, por lo que el retorno es 3.

Para la tercera llamada a `press`, "ABXYY" aparece completa en "ABXYYABXYY" como subcadena, por lo que el retorno es 5.

Para la sexta llamada a `press`, únicamente el prefijo de "ABXYY" correspondiente a la cadena vacía aparece como subcadena en "BXYY", por lo que el retorno es 0.

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

El archivo `sample-01-in.txt` en el archivo `.zip` adjunto corresponde a 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 nunca reaparece en S .

En este problema, el evaluador **NO** es adaptativo. Esto significa que S se fija al principio de la ejecución del evaluador, y no depende de las consultas realizadas por el programa evaluado.

Subtareas

1. (5 puntos) $N = 3$
2. (95 puntos) Ninguna restricción adicional. Para esta subtarea, si q es la cantidad de llamadas a `press`, el puntaje de cada caso de prueba será:
 - 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 es el puntaje mínimo obtenido entre todos los casos de prueba de la subtarea.

Evaluador de ejemplo

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

- línea 1: S

Si el programa es evaluado como **Accepted**, el evaluador de ejemplo imprime `Accepted: q`, siendo q la cantidad de llamadas a la función `press`.

Si el programa es evaluado como **Wrong Answer**, imprime `Wrong Answer: MSG`. Los posibles significados de `MSG` son: :

- `invalid press`: Un valor de p entregado a `press` es inválido. Es decir, el largo de p no esta entre 0 y $4N$, inclusive, o algún caracter de p no es A, B, X, o Y.
- `too many moves`: La función `press` es llamada más de 8 000 veces.
- `wrong guess`: La cadena retornada por `guess_sequence` no es S .