



## 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 apretar una secuencia de botones.

Este juego tiene una secuencia secreta de botones, que puede ser representada como un string  $S$  de esos 4 caracteres. Tú no conoces el string  $S$ , pero sí sabes su longitud  $N$ .

**También sabes que el primer caracter de  $S$  nunca reaparece en él.** Por ejemplo,  $S$  puede ser "ABXXY" o "XYYA", pero no puede ser "AAAAA" o "BXYBX".

Puedes presionar una secuencia de a lo sumo  $4N$  botones para hacer un combo. Sea  $p$  el string que representa la secuencia de botones que presionaste. La cantidad de monedas que obtienes por este combo es calculado como el tamaño del prefijo más largo de  $S$  que también es un substring de  $p$ . Un substring de un string  $t$  es una secuencia contigua (posiblemente vacía) de caracteres en  $t$ . Un prefijo de un string  $t$  es un substring de  $t$  que contiene a su primer caracter, o es vacío.

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 un substring de  $p$ .

Tu tarea es determinar el string secreto  $S$  usando pocos combos.

## Detalles de implementación

Debes implementar la siguiente función:

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

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

Tu programa puede llamar a la siguiente función:

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

- $p$ : una secuencia de botones que apretas.

- $p$  debe ser un string con una 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 para cada caso de prueba.
- Esta función retorna la cantidad de monedas que obtienes cuando presionas la secuencia de botones representada por  $p$ .

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

## Ejemplo

Sea  $S$  el string "ABXY". 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("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

Para la primera llamada a `press`, "ABX" aparece en "XXYYABYABXAY" como substring pero "ABXY" no, por lo que retorna 3.

Para la tercera llamada a `press`, "ABXY" aparece completo en "ABXYABXY" como substring, por lo que retorna 5.

Para la sexta llamada a `press`, solo el prefijo de "ABXY" correspondiente al string vacío aparece como substring en "BXY", por lo que retorna 0.

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

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

## Restricciones

- $1 \leq N \leq 2000$
- Cada caracter del string  $S$  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 tu solución.

## Subtareas

1. (5 puntos)  $N = 3$
2. (95 puntos) Ninguna restricción adicional. Para esta subtarea, tu puntaje para cada caso de prueba se calcula de la siguiente forma: Sea  $q$  la cantidad 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.

Nota que tu puntaje para cada subtarea es el puntaje mínimo obtenido para los casos de prueba en la subtarea.

## Evaluador de ejemplo

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

- línea 1:  $S$

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

Si tu 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 está 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`: El string retornado por `guess_sequence` no es  $S$ .