



Combinaciones

Tú estás jugando un video juego de acción. El control del juego tiene 4 botones, A, B, X, and Y. En este juego, puedes obtener monedas con combinaciones de movidas. Puedes hacer una combinación presionando los botones en determinada secuencia.

Este juego tiene una secuencia secreta de botones, la cual puede ser representada con una cadena S de esos 4 caracteres. Tú no conoces la cadena S , pero sabes que su tamaño es N .

Además, tú sabes que el primer caracter de S nunca reaparece en esta. Por ejemplo, S puede ser "ABXYY" o "XYAA", pero no puede ser "AAAAA" o "BXYBX".

Tú puedes presionar una secuencia de arriba de $4N$ botones para una combinación. Sea p la cadena que representa la secuencia de botones que presionaste. El número de monedas que obtienes por esta movida es calculada como el tamaño del prefijo más largo de S el cual es además una subcadena de p . Una subcadena de una cadena t es una contigua (posiblemente vacía) secuencia de caracteres en t . Un prefijo de t es una subcadena de t que es vacía o contiene el primer caracter 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 es además una subcadena de p .

Tu tarea es determinar la cadena secreta S usando pocas combinaciones de movidas.

Detalles de implementación

Debes implementar la siguiente función:

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

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

Tu programa puede llamar la siguiente función:

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

- p : una secuencia de botones que presionas.

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

Si alguna de las anteriores condiciones no son satisfacidas, tu programa es juzgado como **Wrong Answer**. De otra manera, tu progrma es juzgado como **Accepted** y tu puntaje es calculado por el número de llamadas a `press` (ver Subtareas).

Ejemplo

Si S es "ABXY". El evaluador llama `guess_sequence(5)`. Un ejemplo de comunicación es mostrado a continuacion.

Llamada	Retorna
<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 una subcadena, pero "ABXY" no, entonces 3 es retornado.

Para la tercera llamada a `press`, "ABXY" por si misma aparece en "ABXYABXY" como una subcadena , entonces 5 es retornado.

Para la sexta llamada a `press`, no hay un prefijo de "ABXY" pero la cadena vacía aparece en "BXY" como una subcadena, entonces 0 es retornado.

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

El archivo `sample-01-in.txt` en el paquete comprimido 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 aparece en S .

En este problema, el evaluador es NO adaptativo.

Esto significa que S está puesta al principio de la corrida del evaluador y esta no depende de las solicitudes hechas por tu solución.

Subtareas

1. (5 puntos) $N = 3$
2. (95 puntos) Sin restricciones adicionales. Para esta sub tarea, tu puntaje para cada caso de prueba es calculado como sigue. 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.
 - De otra forma, tu puntaje es 0.

Note que tu puntaje para cada subtarea es el mínimo de los puntajes para los casos de prueba en la sub tarea.

Evaluador de ejemplo

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

- línea 1: S

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

Si tu programa es juzgado como **Wrong Answer**, este imprime `Wrong Answer: MSG`. El significado de `MSG` es como sigue:

- `invalid press`: Un valor de p dado a `press` es inválido. Llamemos, el tamaño de p no está entre 0 y $4N$, inclusive, o algunos caracteres de p no son A, B, X, o Y.
- `too many moves`: La función `press` es llamada más de 8 000 veces.
- `wrong guess`: El valor de retorno de `guess_sequence` no es S .