



# Combo

Você está jogando um video-game de ação. O controle tem 4 botões, A, B, X e Y. Neste jogo, você pode receber moedas como recompensa por combinações de movimentos. Você pode fazer uma combinação de movimentos pressionando uma sequência de botões.

Este jogo tem uma sequência secreta de botões, que pode ser representada por uma string  $S$  constituída por esses 4 caracteres. Você não conhece a string  $S$ , mas conhece o seu comprimento  $N$ .

**Você também sabe que o primeiro caracter de  $S$  nunca volta a aparecer nessa mesma string.** Por exemplo,  $S$  pode ser "ABXYY" ou "XYYAA", mas não pode ser "AAAAA" ou "BXYBX".

Você pode pressionar uma sequência de no máximo  $4N$  botões para formar uma combinação de movimentos. Seja  $p$  a string que representa a sequência de botões que você pressionou. O número de moedas obtidas com esta combinação é calculado como sendo o comprimento do maior prefixo de  $S$  que também é uma substring de  $p$ . Uma substring de  $t$  é uma subsequência contígua de caracteres (possivelmente vazia) de  $t$ . Um prefixo de  $t$  é uma substring de  $t$  que é vazia ou contém o primeiro caracter de  $t$ .

Por exemplo, se  $S$  for "ABXYY" e  $p$  for "XXYYABYABXAY", você irá obter 3 moedas, porque "ABX" é o maior prefixo de  $S$  que também é uma substring de  $p$ .

A sua tarefa é determinar a string secreta  $S$  usando poucas combinações de movimentos.

## Detalhes de implementação

Você deve implementar a seguinte função:

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

- $N$ : o comprimento da string  $S$ .
- Esta função é chamada exatamente uma vez para cada caso de teste.
- Esta função deve devolver a string  $S$ .

O seu programa pode chamar a seguinte função:

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

- $p$ : a sequência de botões que decidiu pressionar.
- $p$  deve ser uma string com comprimento entre 0 e  $4N$ , inclusive. Cada caracter de  $p$  deve ser A, B, X, or Y.
- Não pode chamar esta função mais do que 8 000 vezes para cada caso de teste.
- Esta função retorna o número de moedas obtidas ao pressionar a sequência de caracteres representada por  $p$ .

Se alguma das condições de cima não for satisfeita, o seu programa será avaliado com **Wrong Answer**. Caso contrário, o programa será avaliado com **Accepted** e a sua pontuação é calculada tendo em conta o número de chamadas a `press` (ver Subtarefas).

## Exemplo

Seja  $S$  igual a "ABXY". O avaliador chama `guess_sequence(5)`. Um exemplo de comunicação é mostrado a seguir:

Chamada	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("BXY")</code>	0
<code>press("YYXBA")</code>	1
<code>press("AY")</code>	1

Na primeira chamada a `press`, "ABX" aparece como substring de "XXYYABYABXAY", mas "ABXY" não, e por isso é retornado 3.

Na terceira chamada a `press`, "ABXY" aparece completamente como substring de "ABXYABXY", e por isso é retornado 5.

Na sexta chamada a `press`, nenhum prefixo de "ABXY", a não ser a string vazia, aparece como substring de "BXY", e por isso é retornado 0.

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

O arquivo `sample-01-in.txt` no arquivo zip em anexo corresponde ao exemplo.

## Restrições.

- $1 \leq N \leq 2000$
- Cada caracter da string  $S$  é A, B, X, ou Y.
- O primeiro caracter de  $S$  nunca volta a aparecer em  $S$ .

Neste problema o corretor NÃO é adaptativo. Isto significa que  $S$  é fixada no início da execução do corretor e não depende das perguntas feitas pela sua solução.

## Subtarefas

1. (5 pontos)  $N = 3$
2. (95 pontos) Nenhuma restrição adicional. Para esta sub tarefa, a sua pontuação para cada caso de teste é calculada da seguinte maneira:

Seja  $q$  o número de chamadas a `press`.

- Se  $q \leq N + 2$ , a sua pontuação é 95.
- Se  $N + 2 < q \leq N + 10$ , a sua pontuação é  $95 - 3(q - N - 2)$ .
- Se  $N + 10 < q \leq 2N + 1$ , a sua pontuação é 25.
- Se  $\max\{N + 10, 2N + 1\} < q \leq 4N$ , a sua pontuação é 5.
- Caso contrário, a sua pontuação é 0.

Note que a sua pontuação para cada sub tarefa é o mínimo das pontuações para cada um dos casos de testes dessa sub tarefa.

## Corretor de exemplo

O corretor de exemplo lê input no seguinte formato:

- linha 1:  $S$

Se o seu programa é avaliado como **Accepted**, o avaliador de exemplo imprime `Accepted: q` com  $q$  sendo o número de chamadas à função `press`.

Se o seu programa for avaliado como **Wrong Answer**, é imprimido `Wrong Answer: MSG`. O significado de `MSG` é o seguinte:

- `invalid press`: o valor de  $p$  enviado para `press` é inválido, ou seja, o comprimento de  $p$  não está entre 0 e  $4N$ , inclusive, ou algum caracter de  $p$  não é A, B, X, ou Y.
- `too many moves`: a função `press` foi chamada mais do que 8000 vezes.
- `wrong guess`: o valor de retorno de `guess_sequence` não é  $S$ .