



# Kombo

Po soutěži si můžete zahrát akční hru. Herní konzole má čtyři tlačítka: A, B, X a Y. Ve hře se získávají peníze tak, že děláte komba. Každé kombo se dělá tak, že postupně zmáčknete nějakou posloupnost tlačítek.

Existuje tajná posloupnost tlačítek, kterou můžeme reprezentovat jako řetězec  $S$  složený ze zmíněných čtyř písmen. Tento řetězec sice neznáte, ale zato víte jeho délku  $N$ .

**Také víte, že první znak řetězce  $S$  se už v něm nikdy znovu neobjeví.** Mezi možné řetězce  $S$  tedy patří kupříkladu "ABXY" nebo "XYAA", ale nepatří mezi ně řetězce "AAAA" nebo "BXYB".

Každé kombo se sestává maximálně ze  $4N$  stisků tlačítek. Nechtě  $p$  je řetězec reprezentující posloupnost tlačítek, která jste stiskli. Počet mincí, které za toto kombo dostanete, se spočítá jako délka nejdelšího prefixu  $S$ , který je zároveň podřetězcem  $p$ . Podřetězcem řetězce  $t$  myslíme souvislou (klidně prázdnou) podposloupnost znaků obsažených v  $t$ . Prefix řetězce  $t$  je podřetězec  $t$ , který je buď prázdný, nebo začíná na prvním písmenu  $t$ .

Je-li kupříkladu tajný řetězec  $S$  roven "ABXY" a je-li posloupnost stisknutých znaků  $p$  rovna "XXYYABYABXAY", dostanete 3 mince, neboť "ABX" je nejdelší prefix  $S$ , který je zároveň podřetězcem  $p$ .

Váš úkolem je najít tajný řetězec  $S$  pomocí co nejmenšího počtu kombo řetězců.

## Implementační detaily

Váš úkolem je implementovat následující funkci:

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

- Parametr  $N$  je délka řetězce  $S$ .
- Tato funkce je zavolána právě jednou pro každý testovací vstup.
- Tato funkce má vrátit tajný řetězec  $S$ .

Váš program může volat následující funkci:

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

- Parametr  $p$  je řetězec udávající posloupnost tlačítek, která stisknete.
- Délka řetězce  $p$  musí být mezi 0 a  $4N$  včetně. Každý znak  $p$  musí být jedním ze znaků A, B, X, Y.
- Pro žádný vstup nesmí počet volání této funkce přesáhnout 8 000.
- Tato funkce vrací počet mincí, jež dostanete po stisknutí tlačítek odpovídajících řetězci  $p$ .

Jestliže některé z těchto podmínek nebudou splněny, vyhodnocovací systém odpoví **Wrong answer**. Jinak vyhodnocovací systém odpoví **Accepted** a vaše skóre bude spočteno dle počtu volání funkce `press` (viz sekce Podúlohy).

## Příklad

Předpokládejme, že  $S$  je "ABXYY". Program testující vaše řešení zavolá funkci `guess_sequence(5)`. Následuje příklad možné komunikace.

Volání	Vrácená hodnota
<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

Při prvním zavolání funkce `press` obsahuje řetězec "XXYYABYABXAY" jako podřetězec "ABX", ale neobsahuje "ABXY", takže vrácená hodnota je 3.

Při třetím zavolání funkce `press` se celý řetězec "ABXYY" objeví jako podřetězec řetězce "ABXYYABXYY", takže funkce vrátí hodnotu 5.

Při šestém volání funkce `press` je prázdný řetězec jediným prefixem "ABXYY", který se objeví v řetězci "BXYY". Vrácená hodnota je proto 0.

Funkce `guess_sequence(5)` by měla vrátit řetězec "ABXYY".

Soubor `sample-01-in.txt` v zazipovaném archívu odpovídá tomuto příkladu.

## Omezení

- $1 \leq N \leq 2\,000$
- Každý znak řetězce  $S$  je A, B, X, nebo Y.
- První znak řetězce  $S$  se v něm již nikdy znovu neobjeví.

V tomto problému vyhodnocovač testující vaše řešení NENÍ adaptivní. To znamená, že  $S$  je zafixováno na začátku běhu programu a nezávisí tedy na dotazech, které vaše řešení klade.

## Podúlohy

1. (5 bodů)  $N = 3$
2. (95 bodů) Žádná další omezení. Počet bodů za tuto podúlohu je pro každý testovací vstup spočítán následujícím způsobem. Necht'  $q$  je počet volání funkce `press`.
  - Jestliže  $q \leq N + 2$ , získáte 95 bodů.
  - Jestliže  $N + 2 < q \leq N + 10$ , získáte  $95 - 3(q - N - 2)$  bodů.
  - Jestliže  $N + 10 < q \leq 2N + 1$ , získáte 25 bodů.
  - Jestliže  $\max\{N + 10, 2N + 1\} < q \leq 4N$ , získáte 5 bodů.
  - Jinak dostanete 0 bodů.

Celkový získaný počet bodů za každou podúlohu je minimum z počtu bodů přes všechny testovací vstupy v podúloze.

## Ukázkový testovač (grader)

Ukázkový testovač očekává vstup v následujícím formátu:

řádek 1:  $S$

Jestliže je váš program vyhodnocen jako **Accepted**, ukázkový testovač vytiskne `Accepted: q`, kde  $q$  je počet volání funkce `press`.

Jestliže je váš program vyhodnocen jako **Wrong answer**, ukázkový testovač vytiskne `Wrong answer: MSG`. Řetězec `MSG` může být jeden z následujících:

- `invalid press`: Řetězec  $p$ , který dostala funkce `press` jako parametr, nesplňuje omezení ze zadání. To znamená, že buď délka řetězce  $p$  není mezi 0 a  $4N$  včetně, nebo některý ze znaků v  $p$  není A, B, X, ani Y.
- `too many moves`: Počet volání funkce `press` přesáhl 8 000.
- `wrong guess`: Návrátová hodnota funkce `guess_sequence` není rovna  $S$ .