



Kombo

Grasz właśnie w grę. Twój kontroler ma 4 przyciski: A, B, X, oraz Y. Podczas gry zdobywasz monety przy użyciu pewnych kombinacji ruchów (komb). Komba wykonujesz poprzez wciskanie przycisków w odpowiedniej kolejności.

Gra ma pewną tajną kombinację ruchów, która może być reprezentowana jako słowo S złożone z tych 4 znaków. Nie znasz tego słowa, ale znasz jego długość N .

Wiesz także, że pierwszy znak w S nigdy się w nim nie pojawia ponownie. Dla przykładu, ciągiem S może być "ABXYY" lub "XYAA", ale nie "AAAA" ani też "BXYBX".

Możesz nacisnąć $4N$ przycisków aby wykonać ruch kombo. Niech p oznacza słowo, które reprezentuje ciąg przycisków, które nacisnąłeś. Liczba monet, która otrzymasz za ten ruch to długość najdłuższego prefiksu S , który jest także pod słowem p . Pod słowo słowa t to spójny (być może pusty) ciąg znaków występujący w t . Prefiksem t jest pod słowo t , które jest puste, bądź zawiera pierwszy znak słowa t .

Dla przykładu, gdy S to "ABXYY", a p to "XXYYABYABXAY", wówczas otrzymasz 3 monety, jako że "ABX" jest najdłuższym prefiksem S , który jest także pod słowem p .

Twoim zadaniem jest ustalenie słowa S przy użyciu niewielkiej liczby ruchów kombo.

Szczegóły implementacji

Powinieneś zaimplementować następującą funkcję:

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

- N : długość ciągu S .
- Funkcja ta wywoływana jest dokładnie raz dla każdego przypadku testowego.
- Funkcja ta powinna zwracać słowo S .

Twój program może wywoływać następującą funkcję:

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

- p : ciąg przycisków, które naciskasz.
- p musi być słowem o długości pomiędzy 0 a $4N$, włącznie. Każdy znak p musi być A, B, X lub Y.

- Nie możesz użyć tej funkcji więcej niż 8000 razy dla każdego przypadku testowego.
- Funkcja ta zwraca liczbę monet, które uzyskujesz po wcisnięciu kombinacji ruchów odpowiadających p .

Jeżeli któryś z wymienionych wyżej warunków nie jest spełniony, Twój program zostanie oceniony jako **Wrong Answer**. W przeciwnym wypadku Twój program zostanie oceniony jako **Accepted**, a Twój wynik będzie zależał od liczby wywołań funkcji `press` (zobacz sekcję Podzadania).

Przykład

Niech S będzie słowem "ABXY". Program sprawdzający wywołuje funkcję `guess_sequence(5)`. Przykład komunikacji został przedstawiony poniżej.

Wywołanie	Zwrócona wartość
<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

Dla pierwszego wywołania funkcji `press`, słowo "ABX" występuje jako pod słowo w słowie "XXYYABYABXAY", ale "ABXY" już nie, zatem zwrócona zostaje liczba 3.

Dla trzeciego wywołania funkcji `press`, samo "ABXY" pojawia się w A jako pod słowo, stąd zwrócona zostaje liczba 5.

Dla szóstego wywołania funkcji `press`, żaden prefiks "ABXY" poza pustym nie pojawia się w "BXY" jako pod słowo, dlatego zwracane jest 0.

Finalnie, `guess_sequence(5)` powinno zwrócić "ABXY".

Plik `sample-01-in.txt` w załączonym pakiecie odpowiada temu przykładowi.

Ograniczenia

- $1 \leq N \leq 2000$
- Każdy znak w słowie S to A, B, X lub Y.

- Pierwszy znak S występuje w słowie S jedynie raz.

W tym zadaniu sprawdzaczka NIE jest adaptacyjna. To znaczy, że S jest ustalone przy uruchomieniu programu sprawdzającego i nie zależy od zapytań zadanych przez Twój program.

Podzadania

1. (5 punktów) $N = 3$
2. (95 punktów) Brak dodatkowych ograniczeń. Dla tego podzadania Twój wynik dla każdego testu jest obliczany w następujący sposób. Niech q będzie liczbą wywołań funkcji `press`.
 - Dla $q \leq N + 2$, Twój wynik to 95.
 - Dla $N + 2 < q \leq N + 10$, Twój wynik to $95 - 3(q - N - 2)$.
 - Dla $N + 10 < q \leq 2N + 1$, Twój wynik to 25.
 - Dla $\max\{N + 10, 2N + 1\} < q \leq 4N$, Twój wynik to 5.
 - W przeciwnym wypadku, Twój wynik to 0.

Zwróć uwagę, że wynik dla każdego podzadania jest równy minimalnemu wynikowi ze wszystkich testów w danym podzadaniu.

Przykładowa sprawdzaczka

Przykładowa sprawdzaczka wczytuje wejście w następującym formacie:

- wiersz 1: S

Jeżeli Twój program został oceniony jako **Accepted**, sprawdzaczka wypisze `Accepted:` q gdzie q oznacza liczbę wywołań funkcji `press`.

Jeżeli Twój program został oceniony jako **Wrong Answer**, sprawdzaczka wypisze `Wrong Answer: MSG`. Znaczenie `MSG` jest następujące:

- `invalid press`: Argument `p` przekazany funkcji `press` jest nieprawidłowy. Oznacza to, że długość `p` nie jest pomiędzy 0 a $4N$ (włącznie) lub jakiś znak `p` nie jest `A`, `B`, `X`, albo `Y`.
- `too many moves`: Funkcja `press` została wywołana więcej niż 8 000 razy.
- `wrong guess`: Wartość zwrócona przez `guess_sequence` nie jest równa S .