



# Combo

Vă jucați un joc de acțiune. Controlerul jocului are 4 butoane, A, B, X, și Y. În acest joc acumulați monede cu diferite combo-uri. Puteți obține un combo apăsând o secvență de butoane.

Jocul are o secvență secretă a butoanelor, care poate fi reprezentată ca o secvență de caractere  $S$  formată cu acele 4 caractere. Nu cunoașteți secvența  $S$ , dar îi știți lungimea  $N$ .

**Știți de asemenea că primul caracter al lui  $S$  nu reapare nicăieri în acesta.** De exemplu,  $S$  poate fi "ABXYY" sau "XYAA", dar nu poate fi "AAAAA" sau "BXYBX".

Puteți apăsa o secvență de cel mult  $4N$  butoane pentru a forma un combo. Fie  $p$  secvența de caractere care reprezintă secvența de butoane apăstate. Numărul de monede pe care le veți primi pentru acest combo este calculat ca lungimea celui mai lung prefix al lui  $S$  care este și subsecvență a lui  $p$ . O subsecvență a unei secvențe de caractere  $t$  este o secvență (posibil vidă) de caractere aflate pe poziții consecutive în  $t$ . Un prefix al lui  $t$  este o subsecvență a lui  $t$  care fie este vid, fie conține primul caracter din  $t$ .

De exemplu, dacă  $S$  este "ABXYY" și  $p$  este "XXYYABYABXAY", veți primi 3 monede deoarece "ABX" este cel mai lung prefix al lui  $S$  care este și subsecvență a lui  $p$ .

Sarcina voastră este să determinați secvența secretă de caractere  $S$  utilizând puține combo-uri.

## Detalii de implementare

Trebuie să implementați următoarea funcție:

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

- $N$ : lungimea secvenței  $S$ .
- Această funcție este apelată exact o dată pentru fiecare test.
- Această funcție trebuie să întoarcă secvența  $S$ .

Programul vostru poate apela următoarea funcție:

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

- $p$ : o secvență de butoane pe care le apăsați.
- $p$  trebuie să fie o secvență de caractere cu lungimea cuprinsă între 0 și  $4N$  inclusiv. Fiecare caracter din  $p$  trebuie să fie ori A, ori B, ori X ori Y.
- Nu puteți apela această funcție de mai mult de 8 000 de ori pentru fiecare test.
- Această funcție va întoarce numărul de monede pe care le veți primi dacă apăsați secvența de butoane reprezentată de secvența  $p$ .

Dacă vreuna din condițiile de mai sus nu este satisfăcută, programul vostru va fi evaluat ca **Wrong answer**. Altfel, programul vostru va fi evaluat ca **Accepted**, iar punctajul vostru va fi calculat după numărul de apeluri ale funcției `press` (vezi Subtask-uri).

## Exemplu

Fie secvența  $S$  egală cu "ABXY". Grader-ul apelează `guess_sequence(5)`. Un exemplu de comunicare este prezentat mai jos.

Apel	Întoarcere
<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

Pentru primul apel al funcției `press`, "ABX" apare în "XXYYABYABXAY" ca subsecvență, dar "ABXY" nu apare, deci se va întoarce 3.

Pentru al treilea apel al funcției `press`, "ABXY" apare în "ABXYABXY" ca subsecvență, deci se va întoarce 5.

Pentru al șaselea apel al funcției `press`, niciun prefix de-al secvenței "ABXY" în afara de subsecvența vidă nu apare în "BXY" ca subsecvență, deci se va întoarce 0.

La sfârșit, `guess_sequence(5)` ar trebui să întoarcă "ABXY".

Fișierul `sample-01-in.txt` din pachetul arhivat anexat corespunde acestui exemplu.

## Constrângeri

- $1 \leq N \leq 2\,000$
- Fiecare caracter al secvenței  $S$  este ori A, ori B, ori X ori Y.
- Primul caracter al lui  $S$  nu reapare în  $S$ .

În această problemă, grader-ul NU este adaptiv. Asta înseamnă că  $S$  este fixat la începutul rulării grader-ului și nu depinde de întrebările puse de programul voastră.

## Subtask-uri

1. (5 puncte)  $N = 3$
2. (95 de puncte) Fără constrângeri adiționale. Pentru acest subtask punctajul vostru este calculat după cum urmează. Fie  $q$  numărul de apeluri ale funcției `press`.
  - Dacă  $q \leq N + 2$ , punctajul vostru este 95.
  - Dacă  $N + 2 < q \leq N + 10$ , punctajul vostru este  $95 - 3(q - N - 2)$ .
  - Dacă  $N + 10 < q \leq 2N + 1$ , punctajul vostru este 25.
  - Dacă  $\max\{N + 10, 2N + 1\} < q \leq 4N$ , punctajul vostru este 5.
  - Altfel, punctajul vostru este 0.

Luați la cunoștință ca punctajul pentru un subtask este minimul dintre punctajele testelor care alcătuiesc acel subtask.

## Grader local

Grader-ul local citește datele de intrare în următoarea formă:

- linia 1:  $S$

Dacă programul vostru este evaluat ca **Accepted**, atunci grader-ul local va afișa `Accepted: q` unde  $q$  va reprezenta numărul de apeluri ale funcției `press`.

Dacă programul vostru este evaluat ca **Wrong answer**, atunci grader-ul local va afișa `Wrong answer: MSG`. Semnificația lui `MSG` este după cum urmează:

- `invalid press`: O valoare greșită a lui  $p$  a fost folosită în apelul funcției `press`. Mai exact, lungimea lui  $p$  nu este între 0 și  $4N$  inclusiv, sau un caracter din  $p$  nu este ori A, ori B, ori X ori Y.
- `too many moves`: Funcția `press` a fost apelată de mai mult de 8 000 de ori.
- `wrong guess`: Valoarea întoarsă de funcția `guess_sequence` nu este  $S$ .