



# Combo

Vous jouez à un jeu vidéo d'action. La manette possède 4 boutons : A, B, X, et Y. Dans ce jeu, vous obtenez des jetons en effectuant des combos. Vous pouvez effectuer un combo en appuyant successivement sur certains boutons.

Le jeu possède une séquence secrète de boutons, qu'on représentera par une chaîne  $S$  des 4 caractères précités. Vous ne connaissez pas la chaîne  $S$ , mais vous connaissez sa longueur  $N$ .

**Vous savez également que le premier caractère de  $S$  n'apparaît qu'une seule fois.** Par exemple,  $S$  peut être "ABXYY" ou "XYAA", mais pas "AAAAA" ou "BXYBX".

Vous pouvez appuyer sur une séquence d'au plus  $4N$  boutons pour faire un combo. Soit  $p$  la chaîne de caractères qui représente la séquence de boutons sur lesquels vous avez appuyé. Le nombre de jetons que vous obtenez en effectuant ce combo est égal à la longueur du plus long préfixe de  $S$  qui est une sous-chaîne de  $p$ . Une sous-chaîne d'une autre chaîne de caractères  $t$  est une suite continue de caractères de  $t$  (peut être vide). Un préfixe de  $t$  est une sous-chaîne de  $t$  qui est vide ou qui contient le premier caractère de  $t$ .

Par exemple, si  $S$  est "ABXYY" et  $p$  est "XXYYABYABXAY", vous obtenez 3 jetons car "ABX" est le plus long préfixe de  $S$  qui est aussi une sous-chaîne de  $p$ .

Votre mission est de déterminer la chaîne de caractères secrète  $S$  en effectuant peu de combos.

## Détails d'implémentation

Vous devez implémenter la fonction suivante :

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

- $N$ : La longueur de la chaîne de caractères  $S$ .
- Cette fonction est appelée exactement une fois pour chaque test.
- Cette fonction doit retourner la chaîne de caractères  $S$ .

Votre programme peut appeler la fonction suivante :

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

- $p$ : La séquence de boutons sur lesquels vous appuyez.
- La longueur de  $p$  doit être comprise entre 0 et  $4N$  inclus. Chaque caractère de  $p$  doit être soit A, B, X, ou Y.
- Vous ne pouvez pas appeler cette fonction plus de 8 000 fois pour chaque cas de test.
- Cette fonction retourne le nombre de jetons que vous obtenez lorsque vous entrez la séquence  $p$ .

Si l'une des conditions ci-avant n'est pas satisfaite, votre programme est jugé comme **Wrong Answer**. Sinon, il est jugé **Accepted** et votre score est calculé en fonction du nombre de fois où vous appelez `press` (regarder les sous-tâches).

## Exemple

Soit  $S$  égale à "ABXY". L'évaluateur appelle `guess_sequence(5)`. Un exemple de communication est affiché ci-dessous :

Appel	Retour
<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

Pour le premier appel de `press`, "ABX" apparaît dans "XXYYABYABXAY" comme sous-chaine mais pas "ABXY", la valeur retournée est donc 3.

Pour le troisième appel de `press`, "ABXY" est une sous-chaine de "ABXYABXY", la valeur retournée est donc 5.

Pour le sixième appel de `press`, aucun préfixe de "ABXY" n'apparaît dans "BXY" sauf la chaîne vide, donc la valeur retournée est 0.

Et finalement, `guess_sequence(5)` doit retourner "ABXY".

Le fichier `sample-01-in.txt` dans l'archive du problème correspond à cet exemple.

## Contraintes

- $1 \leq N \leq 2\,000$
- Chaque caractère de  $S$  est égal à A, B, X, ou Y.
- Le premier caractère de  $S$  ne réapparaît jamais dans  $S$ .

Dans ce problème, l'évaluateur n'est PAS adaptatif. Cela signifie que  $S$  est fixé au début de l'exécution de l'évaluateur et ne dépend pas des requêtes effectuées par votre programme.

## Sous-tâches

1. (5 points)  $N = 3$
2. (95 points) Pas de contrainte additionnelle. Pour cette sous-tâche, votre score pour chaque fichier test est calculé de la manière suivante: Soit  $q$  le nombre de fois où vous appelez `press`.
  - Si  $q \leq N + 2$ , votre score sera 95.
  - Si  $N + 2 < q \leq N + 10$ , votre score sera  $95 - 3(q - N - 2)$ .
  - Si  $N + 10 < q \leq 2N + 1$ , votre score sera 25.
  - Si  $\max\{N + 10, 2N + 1\} < q \leq 4N$ , votre score sera 5.
  - Dans les autres cas, votre score sera 0.

Veuillez noter que votre score de chaque sous-tâche est le score minimum parmi les fichiers tests de la sous-tâche.

## Évaluateur d'exemple (Sample grader)

L'évaluateur lit l'entrée dans le format suivant:

- ligne 1 :  $S$

Si votre programme est jugé **Accepted**, L'évaluateur affiche "Accepted: q" avec q le nombre d'appels à la fonction `press`.

Si votre programme est jugé **Wrong Answer**, il affiche "Wrong Answer: MSG". La signification de MSG est la suivante :

- `invalid press` : la valeur de `p` envoyée à `press` est invalide. Potentiellement, la longueur de `p` n'est pas entre 0 et  $4N$  inclus, ou quelques caractères de `p` diffèrent de A, B, X, ou Y.
- `too many moves` : La fonction `press` est appelée plus de 8 000 fois.
- `wrong guess` : La chaîne de caractères retournée par `guess_sequence` est différente de  $S$ .