



Combo

אתם משחקים במשחק מחשב. לשלט יש 4 כפתורים, A, B, X ו-Y. במשחק זה, ניתן להשיג מטבעות באמצעות מהלכי קומבו. ניתן לבצע מהלך קומבו באמצעות לחיצה על סדרת כפתורים.

במשחק יש סדרה סודית של כפתורים, אשר מיוצגת על ידי מחרוזת S של 4 התווים לעיל. אינכם יודעים את המחרוזת S, אבל אתם יודעים את אורכה N.

אתם גם יודעים שהתו הראשון של S אינו מופיע בה שוב. לדוגמה, S יכולה להיות "ABXYY" או "XYAAA", אך היא לא יכולה להיות "AAAAA" או "BXYBX".

ניתן ללחוץ על סדרה של לכל היותר $4N$ כפתורים כדי לבצע מהלך קומבו. נגדיר את p להיות המחרוזת שמייצגת את סדרת הכפתורים שלחצתם. מספר המטבעות שתקבלו על מהלך זה מחושב בתור אורך הרישא הכי ארוכה של S שהיא תת מחרוזת של p . תת מחרוזת של מחרוזת t היא סדרה רצופה (אולי ריקה) של תווים ב- t . רישא של t היא תת מחרוזת של t שהיא ריקה או מכילה את התו הראשון של t .

לדוגמה, אם S היא "ABXYY" ו- p היא "XXYYABYABXAY", תקבלו 3 מטבעות כי "ABX" היא הרישא הארוכה ביותר של S שהיא גם תת מחרוזת של p .

משימתכם היא לגלות את המחרוזת הסודית S על ידי ביצוע מעט מהלכי קומבו.

פרטי מימוש

עליכם לממש את הפונקציה הבאה:

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

• N : אורך המחרוזת S.

• פונקציה זו נקראת בדיוק פעם אחת בכל test case.

• פונקציה זו צריכה להחזיר את המחרוזת S.

תוכניתכם יכולה לקרוא לפונקציה הבאה:

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

• p : סדרת כפתורים שעליהם אתם לוחצים.

• צריכה להיות מחרוזת שאורכה בין 0 ל- $4N$, כולל. כל תו של p צריך להיות A, B, X או Y.

• אינכם רשאים לקרוא לפונקציה זו יותר מ-8000 פעמים באף test case.

• פונקציה זו מחזירה את מספר המטבעות שמקבלים כשלוחצים על סדרת הכפתורים ש- p מייצגת.

אם אחד או יותר מהתנאים האלה לא מתקיים, הפידבק על תוכניתכם יהיה Wrong Answer. אחרת, הפידבק יהיה Accepted והניקוד שלכם יחושב לפי מספר הקריאות ל-press (ראו תת משימות).

דוגמה

נגדיר את S להיות "ABXY". הגריידר קורא ל-`guess_sequence(5)`. דוגמה לתקשורת מופיעה להלן.

קריאה	ערך החזרה
<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

בקריאה הראשונה ל-`press`, "ABX" מופיעה בתוך "XXYYABYABXAY" כתת מחרוזת אך "ABXY" אינה מופיעה כתת מחרוזת, לכן ערך החזרה הוא 3.

בקריאה השלישית ל-`press`, "ABXY" מופיעה בתוך "ABXYABXY" כתת מחרוזת, לכן ערך החזרה הוא 5. בקריאה השישית ל-`press`, אין רישא של "ABXY" שמופיעה בתוך "BXY" כתת מחרוזת למעט המחרוזת הריקה, לכן ערך החזרה הוא 0.

לבסוף, `guess_sequence(5)` צריכה להחזיר "ABXY".

הקובץ `sample-01-in.txt` שנמצא ב-`zip` המצורף מתאים לדוגמה זו.

מגבלות

- $1 \leq N \leq 2000$
- כל תו של המחרוזת S הוא A, B, X או Y .
- התו הראשון ב- S אינו מופיע שוב ב- S .

בבעיה זו, הגריידר אינו אדפטיבי. כלומר S נקבעת בתחילת הריצה של הגריידר ואינה תלויה בשאלות שאתם מבצעים.

תת משימות

1. (5 נקודות) $N = 3$
2. (95 נקודות) ללא מגבלות נוספות. בתת משימה זו, הניקוד לכל `test case` מחושב בצורה הבאה. נגדיר את q להיות מספר הקריאות ל-`press`.

- אם $q \leq N + 2$, הניקוד הוא 95.
- אם $N + 2 < q \leq N + 10$, הניקוד הוא $95 - 3(q - N - 2)$.
- אם $N + 10 < q \leq 2N + 1$, הניקוד הוא 25.

- אם $\max\{N + 10, 2N + 1\} < q \leq 4N$, הניקוד הוא 5.
- אחרת, הניקוד הוא 0.

שימו לב שהניקוד של כל תת משימה הוא הניקוד המינימלי מתוך כל ה-test cases שבתוכה.

גריידר לדוגמה (sample grader)

הגריידר לדוגמה קורא קלט בפורמט הבא:

- שורה 1: S

אם הפידבק על תוכניתכם הוא Accepted, הגריידר לדוגמה מדפיס q Accepted: כאשר q הוא מספר הקריאות לפונקציה press.

אם הפידבק על תוכניתכם הוא Wrong Answer, הגריידר לדוגמה מדפיס Wrong Answer: MSG כאשר MSG הוא:

- invalid press: אחד מערכי p שניתן ל-press אינו תקין. כלומר, האורך של p אינו בין 0 ל- $4N$, כולל, או אחד התווים של p אינו A, B, X או Y.
- too many moves: הפונקציה press נקראה יותר מ-8000 פעמים.
- wrong guess: ערך החזרה של guess_sequence אינו S .