Combo

Du spielst ein Action Video Game. Der Gamecontroller hat die 4 Buttons `A`, `B`, `X` und `Y`. In diesem Spiel kannst du mit Kombo-Moves Münzen gewinnen. Ein Kombo-Move ist eine Sequenz von gedrückten Buttons.

Dieses Spiel hat eine geheime Sequenz von Buttons, die als String S mit den entsprechenden 4 Zeichen dargestellt wird. Du kennst den String S nicht, aber du kennst seine Länge N.

Du weißt auch, dass das erste Zeichen von S kein weiteres Mal in S vorkommt. Z.B. kann S "ABXYY" oder "XYYAA" sein, aber keinesfalls "AAAAA" oder "BXYBX".

Für einen Kombo-Move kannst du eine Sequenz von bis zu $4N$ Buttons drücken. Wir bezeichnen mit p den String, der einer solchen Sequenz von gedrückten Buttons entspricht. Die Länge des längsten Präfix von S, der auch ein Substring von p ist, ist die Anzahl der Münzen, die du für den Kombo-Move p bekommst. Ein Substring eines Strings t ist eine zusammenhängende (möglicherweise leere) Sequenz in t. Ein Präfix von t ist ein Substring von t, der entweder leer ist oder das erste Zeichen von t enthält.

Zum Beispiel, wenn S "ABXYY" ist und p "XXYYABYABXAY", dann bekommst du 3 Münzen, weil "ABX" der längste Präfix von S ist, der auch ein Substring von p ist.

Du hast die Aufgabe den geheimen String S mit wenigen Kombo-Moves zu bestimmen.

Implementierungshinweise

Du musst die folgende Funktion implementieren:

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

- `N`: die Länge des Strings S.
- Diese Funktion wird für jeden Testcase genau einmal aufgerufen.
- Diese Funktion muss den String S zurückgeben.

Dein Programm kann die folgende Funktion aufrufen:

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

- `p`: eine Sequenz von deinen gedrückten Buttons.

- p muss eine Länge zwischen 0 und $4N$ (inklusive) haben. Jedes Zeichen in p muss entweder A, B, X oder Y sein.
- Du kannst diese Funktion höchstens 8 000 Mal pro Testcase aufrufen.
- Diese Funktion liefert dir die Anzahl der Münzen, die du durch den Kombo-Move p bekommst.

Wenn mindestens eine der obigen Bedingungen nicht erfüllt ist, wird dein Programm mit **`Wrong Answer`** beurteilt. Andersfalls wird dein Programm mit **`Accepted`** beurteilt und deine Punkteanzahl wird abhängig von der Anzahl deiner `press` Aufrufe berechnet (siehe Teilaufgaben).

Beispiel

Sei `S` gleich "ABXYY". Der Grader ruft `guess_sequence(5)` auf. Eine Beispielinteraktion wird in folgender Tabelle demonstriert:

Aufruf	Rückgabewert
`press("XXYYABYABXAY")`	3
`press("ABXYY")`	5
`press("ABXYYABXYY")`	5
`press("")`	0
`press("X")`	0
`press("BXYY")`	0
`press("YYXBA")`	1
`press("AY")`	1

Im ersten Aufruf von `press` kommt "ABX" als Substring von "XXYYABYABXAY" vor, "ABXY" hingegen nicht, weshalb 3 zurückgegeben wird.

Im dritten Aufruf von `press` kommt "ABXYY" als Substring von "ABXYYABXYY" vor, also wird 5 zurückgegeben.

Im sechsten Aufruf von `press` ist der leere String der einzige Präfix von "ABXYY", der auch ein Substring von "BXYY" ist. `press` liefert somit 0.

In diesem Beispiel soll `guess_sequence(5)` "ABXYY" zurückgeben.

Die Datei `sample-01-in.txt` des Zip Archives entspricht diesem Beispiel.

Einschränkungen

- $1 \leq N \leq 2000$

- Jedes Zeichen des Strings S ist entweder `A`, `B`, `X` oder `Y`.
- Das erste Zeichen von S kommt kein zweites Mal in S vor.

Der Grader dieser Aufgabe passt sich nicht an dein Programm an. Das bedeutet, dass S von Beginn an festgelegt wird und nicht von den Abfragen abhängt.

Teilaufgaben

1. (5 Punkte) $N = 3$
2. (95 Punkte) Keine zusätzlichen Einschränkungen. Bei diesem Subtask werden deine Punkte wie folgt berechnet. Sei q die Anzahl an Aufrufen von `press`.
   - Wenn $q \leq N + 2$, erhältst du 95 Punkte.
   - Wenn $N + 2 < q \leq N + 10$, erhältst du $95 - 3(q - N - 2)$ Punkte.
   - Wenn $N + 10 < q \leq 2N + 1$, erhältst du 25 Punkte.
   - Wenn $\max\{N + 10, 2N + 1\} < q \leq 4N$, erhältst du 5 Punkte.
   - Ansonsten erhältst du 0 Punkte.

Beachte, dass sich die Punkteanzahl jeder Teilaufgabe als Minimum der Punkte über die Testcases dieser Teilaufgabe ergibt.

Beispielgrader

Der Beispielgrader liest die Eingabe in folgendem Format ein:

- Zeile 1: S

Wenn dein Programm die richtige Antwort liefert, gibt der Beispielgrader `Accepted: q` aus, wobei `q` die Anzahl an Aufrufen von `press` sind.

Falls dein Programm die falsche Antwort liefert, gibt er `Wrong Answer: MSG` aus, wobei `MSG` einen der folgenden Werte hat:

- `invalid press`: Der Parameter `p` eines Aufrufs von `press` entspricht nicht den Anforderungen. Das heißt, dass die Länge von `p` nicht zwischen 0 und $4N$ (inklusive) ist, oder eines der Zeichen von `p` nicht `A`, `B`, `X` oder `Y`.
- `too many moves`: Die Funktion `press` wurde mehr als $8\,000$ Mal aufgerufen.
- `wrong guess`: Der Rückgabewert von `guess_sequence` ist nicht `S`.