コンボ

あなたはテレビゲームで遊んでいる. ゲームのコントローラには 4 種類のボタン A, B, X, Y がついている. このゲームではボタンを連続して押すことでコンボを繰り出すことができ, コンボを繰り出すとコインがもらえる.

このゲームには, 秘密のコマンドが決められている. 秘密のコマンドはボタンに対応する 4 種類の文字から構成される文字列 S で表される. あなたは S そのものは知らないが, その長さ N は知っている.

また, S の先頭文字が先頭以外の場所では現れないこともわかっている. 例えば, "ABXYY" や "XYYAA" はこの条件を満たすが, "AAAAA" や "BXYBX" はこの条件を満たさない.

1 回のコンボの中でボタンを押す回数は $4N$ 以下でなければならない. ここで押したボタンに対応する文字列を p とする. このコンボでもらえるコインの枚数は, S の接頭辞であり, かつ p の部分文字列であるような文字列のうち, 最も長いものの長さと一致する. ここで, 文字列 t の部分文字列とは, t の中に現れる連続した (空でもよい) 文字列である. また, 文字列 t の接頭辞とは, t の部分文字列のうち, 空であるか t の先頭文字を含むものである.

例えば, S が "ABXYY" であり, p が "XXYYABYABXAY" であるとすると, S の接頭辞であり p の部分文字列である文字列のうち最長のものは "ABX" であるため, あなたは 3 枚のコインをもらえる.

あなたの仕事は, 十分少ない回数のコンボで, 秘密のコマンド S を決定することである.

実装の詳細

あなたは, 以下の関数を実装する必要がある:

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

- N: 文字列 S の長さである.
- この関数は, それぞれのテストケースに対してちょうど 1 回だけ呼び出される.
- この関数は, 文字列 S を返さなければならない.

提出するプログラムでは, 以下の関数を呼び出すことができる:

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

- p: 押すボタンの列である.
- p の長さは 0 以上 $4N$ 以下でなければならない. p は A, B, X, Y の 4 種類の文字のみから構成されなければならない.
- それぞれのテストケースに対して, この関数を $8\,000$ 回を超えて呼び出すことはできない.

- この関数は, p で表されるコンボを繰り出したときにもらえるコインの枚数を返す.

もし, 上に挙げた条件のいずれかが満たされていない場合, 提出したプログラムは **Wrong Answer** と判定される. 条件がすべて満たされている場合, 提出したプログラムは **Accepted** と判定され, press の呼び出し回数に応じて得点が計算される ("小課題" の節を参照).

入出力例

S が "ABXYY" であるとする. 採点プログラムは guess_sequence(5) を呼び出す. やり取りの例を以下に示す.

呼び出し	返り値
press("XXYYABYABXAY")	3
press("ABXYY")	5
press("ABXYYABXYY")	5
press("")	0
press("X")	0
press("BXYY")	0
press("YYXBA")	1
press("AY")	1

1 回目の press の呼び出しでは, "ABX" は "XXYYABYABXAY" の部分文字列だが, "ABXY" はそうではないので, 3 が返される.

3 回目の press の呼び出しでは, "ABXYY" 自体が "ABXYYABXYY" の部分文字列であるので, 5 が返される.

6 回目の press の呼び出しでは, "ABXYY" の空文字列以外の接頭辞は "BXYY" の部分文字列でないので, 0 が返される.

最終的に, guess_sequence(5) は "ABXYY" を返さなければならない.

zip 圧縮された添付パッケージ (attachment package) に入ったファイル sample-01-in.txt がこの例に対応している。

制約

- $1 \leq N \leq 2\,000$.
- S は A, B, X, Y の 4 種類の文字のみから構成される.
- S の先頭文字は先頭以外では現れない.

この課題では, 採点プログラムは adaptive ではない. つまり S は, 採点プログラムの実行開始時に固定さ

れており, 提出したプログラムの press の呼び出しの内容に応じて変化することはない.

小課題

1. (5 点) $N = 3$.
2. (95 点) 追加の制約はない. この小課題では, それぞれのテストケースに対する得点が以下のように計算される. このテストケースでの press の呼び出し回数を q としたとき,
   - もし $q \leqq N + 2$ならば, 得点は 95 点となる.
   - もし $N + 2 < q \leqq N + 10$ ならば, 得点は $95 - 3(q - N - 2)$ 点となる.
   - もし $N + 10 < q \leqq 2N + 1$ ならば, 得点は 25 点となる.
   - もし $\max\{N + 10, 2N + 1\} < q \leqq 4N$ ならば, 得点は 5 点となる.
   - 上のいずれにも当てはまらない場合, 得点は 0 点となる.

各小課題の最終的な得点は, その小課題のそれぞれのテストケースでの得点の最小値となることに注意せよ.

採点プログラムのサンプル

採点プログラムのサンプルの入力形式は以下の通りである.

- 1 行目: S

もし提出したプログラムが **Accepted** と判定されるならば, 採点プログラムのサンプルは, press を呼び出した回数を q として, `Accepted: q` と出力する.

もし提出したプログラムが **Wrong Answer** と判定されるならば, 採点プログラムは `Wrong Answer: MSG` と出力する. MSG の部分には状況に応じて以下のいずれかのメッセージが入る.

- `invalid press`: 関数 press を呼び出すときの引数 p が無効である. つまり, p の文字数が 0 以上 $4N$ 以下ではない, もしくは, p が A, B, X, Y 以外の文字を含んでいる, ということである.
- `too many moves`: 関数 press が $8\,000$ 回を超えて呼び出されている.
- `wrong guess`: 関数 guess_sequence の返り値が S と一致しない.