콤보

당신은 액션 비디오 게임을 하고 있다. 게임 컨트롤러는 4개의 버튼 A, B, X, Y를 가지고 있다. 게임에서 당신은 콤보 동작을 통해 동전을 얻을 수 있다. 당신은 버튼을 연속해서 눌러서 콤보 동작을 만들 수 있다.

이 게임에는 4개 문자 A,B,X,Y로 이루어진 문자열 S로 나타내는 버튼들의 비밀 순서가 있다. 당신은 문자열 S을 알지 못하지만 그것의 길이 N은 알고 있다.

당신은 S의 첫 번째 문자가 S안에서 다시 나타나지 않는다는 것도 알고 있다.

예를 들어, S는"ABXYY" 또는 "XYYAA"일 수 있지만 "AAAAA" 또는 "BXYBX"가 될 수는 없다.

콤보 동작을 위해서는 $4N$개 이하의 버튼들을 누른다. p가 당신이 누르는 버튼들의 순서를 나타내는 문자열이라고 하자. 콤보 동작으로 얻는 동전의 수는 p의 부분문자열이 되는 S의 가장 긴 접두사의 길이로 계산된다. 문자열 t의 부분문자열은 t안의 연속된 문자들의 순서(길이가 0인 문자열 가능)이다. t의 접두사는 길이가 0인 문자열 이거나 t의 첫 번째 문자를 포함하는 t의 부분문자열이다.

예를 들어, S가 "ABXYY"이고 p가 "XXYYABYABXAY"이면, "ABX"가 p의 부분문자열이면서 S의 가장 긴 접두사이기 때문에 당신은 3개의 동전을 얻는다.

당신은 적은 콤보 동작을 사용해서 비밀 문자열 S를 찾아야 한다.

Implementation details

다음 함수를 구현해야 한다:

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

- N: 문자열 S의 길이.
- 이 함수는 각 테스트 케이스 마다 정확히 한번 호출된다.
- 이 함수는 문자열 S를 반환해야 한다.

프로그램은 다음 함수를 호출할 수 있다:

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

- p: 당신이 누르는 버튼들의 순서.
- p는 0이상 $4N$이하 길이의 문자열이여야 한다. p의 각 문자는 A, B, X, Y 중 하나여야 한다.
- 각 테스트 케이스에 대해 이 함수를 8 000번을 초과해서 호출할 수 없다.
- 이 함수는 p로 표현된 버튼들의 순서을 눌렸을 때 얻을 수 있는 동전의 수를 반환한다.

위 조건들 중 하나라도 만족하지 않으면, 프로그램은 **`Wrong Answer`** 로 판정된다. 그렇지 않으면, 프로그램은 **`Accepted`** 로 판정되고 점수는 함수 `press`의 호출 수로 계산된다(Subtasks를 보시오).

Example

S를 'ABXYY"라고 하자. 그레이더가 `guess_sequence(5)`를 호출한다. 게임 수행의 예가 아래에 보여진다.

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

`press`의 첫 번째 호출에서 "ABX"는 "XXYYABYABXAY"의 부분문자열이지만 "ABXY"는 그렇지 않다. 따라서 3이 반환된다.

`press`의 세 번째 호출에서 "ABXYY"는 "ABXYYABXYY"의 부분문자열이다. 따라서 5가 반환된다.

`press`의 여섯 번째 호출에서 길이가 0인 문자열을 제외하면 "ABXYY"의 어떤 접두사도 "BXYY"의 부분문자열이 아니다. 따라서 0이 반환된다.

결과적으로 `guess_sequence(5)`는 "ABXYY"를 반환해야 한다.

압축된 첨부 패키지 파일의 `sample-01-in.txt` 는 이 예제에 대응한다.

Constraints

- $1 \leq N \leq 2\,000$
- 문자열 S의 각 문자는 A, B, X, Y 중 하나이다.
- S의 첫 번째 문자는 이후에 S에서 결코 다시 나타나지 않는다.

이 문제에서 그레이더는 적응적이지 않다(NOT adaptive). 이것은 S가 그레이더의 수행 초기에 고정되어서 당신이 요청하는 쿼리에 따라 바뀌지 않는다는 것을 의미한다.

Subtasks

1. (5 points) $N = 3$
2. (95 points) 추가적인 제약조건이 없다. 이 subtask에서 각 테스트 케이스의 점수는 다음과 같이 계산된다. q가 `press`의 호출 횟수라고 하자.
   - $q \leq N + 2$ 이면, 점수는 95 이다.
   - $N + 2 < q \leq N + 10$ 이면, 점수는 $95 - 3(q - N - 2)$ 이다.
   - $N + 10 < q \leq 2N + 1$ 이면, 점수는 25 이다.
   - $\max\{N + 10, 2N + 1\} < q \leq 4N$ 이면, 점수는 5 이다.
   - 이외의 모든 경우에 점수는 0 이다.

각 subtask의 점수는 그 subtask의 테스트 케이스들에 대한 점수 중 최소값임에 주의하자.

Sample grader

샘플 그레이더는 다음 형식으로 입력을 받는다:

- line 1: S

프로그램이 **Accepted**로 판정되면, 샘플 그레이더는 `Accepted : q` 를 출력한다. 여기서, `q`는 함수 `press`의 호출 횟수이다.

프로그램이 **Wrong Answer**로 판정되면, 샘플 그레이더는 `Wrong : MSG` 로 출력한다. 여기서, `MSG`는 다음 중 하나이다.

- `invalid press` : `press`에 주어진 `p`의 값이 유효하지 않다. 다시 말해서, `p`의 길이가 0이상 $4N$ 이하가 아니거나 `p`의 어떤 문자가 `A`, `B`, `X`, `Y` 중 하나가 아니다.
- `too many moves` : 함수 `press`가 $8\,000$ 번을 초과해서 호출된다.
- `wrong guess` : `guess_sequence`의 반환 값이 S가 아니다.