

# Słowa FibOli (słowa-fiboli)

Memory limit: 64 MB

Time limit: 1.00 s

Oli ostatnio spodobało się przetwarzanie słów. Przetwarzanie słowa polega na tym, że Ola przegląda je literka po literce i do nowego słowa dokłada kolejne kawałki: jeśli zobaczyła literkę  $a$  to do nowego słowa dokłada  $ab$ , a jeśli  $b$  to dokłada  $a$ . Nowe słowo na początku jest puste. Na przykład ze słowa  $abba$  otrzymałaby  $abaaab$ . Pierwszym słowem, od którego Ola zaczyna jest  $b$ , więc potem kolejno otrzyma  $a$ ,  $ab$ ,  $aba$ ,  $abaab$  itd.

Dzisiaj Olę zaciekało  $N$ -te otrzymane w ten sposób słowo. Wypisała więc wszystkie jego sufiksy na kartce i teraz się zastanawia, co by było gdyby je wszystkie posortowała. Na przykład gdyby sortowała sufiksy  $abaab$  to by otrzymała:  $aab$ ,  $ab$ ,  $abaab$ ,  $b$ ,  $baab$ .

Oczywiście, nie interesuje jej cała lista posortowana, lecz jedynie parę konkretnych pozycji. Pomóż Oli i powiedz jej, jak długie będzie słowo na danej pozycji – skoro ona ma kartkę z całym słowem, to już sobie dalej poradzi.

## Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne  $N$  i  $Q$  – odpowiednio numer słowa, które Olę interesuje oraz liczba jej pytań  $Q$ . W kolejnych  $Q$  wierszach znajdują się pytania Oli – liczby naturalne  $T_i$  określające zapytanie o  $T_i$ -ty leksykograficznie sufiks słowa Oli.

Możesz założyć, że ciąg operacji jest poprawny tzn. że Ola nie będzie pytać o pozycję większą niż długość rozważanego słowa.

## Wyjście

Twój program powinien wypisać na wyjście  $Q$  wierszy. W  $i$ -tym wierszu powinna się znaleźć jedna liczba całkowita będąca odpowiedzią na pytanie Oli.

## Ograniczenia

$$1 \leq N \leq 90, 1 \leq Q \leq 100.$$

## Przykład

Input	Output
5 3	3
1	5
3	4
5	