



## ADN Marțian

Rusia este cunoscută prin succesele sale în explorarea spațiului. Recent, oamenii de știință ruși au analizat niște mostre de sol Marțian și au găsit niște molecule ciudate, care, cred ei, ar putea fi un fel de ADN. Spre deosebire de ADN-ul obișnuit, această moleculă are două elemente de bază în loc de patru. Astfel, întreaga moleculă poate fi descrisă ca un șir de zerouri și unități.

Oamenii de știință au calculat lungimea moleculei, aceasta are  $n$  elemente de bază. Acum ei doresc să determine structura sa, adică să găsească șirul de zerouri și unități  $S$ , care codifică elementele ADN-ului. În scopul aceasta, ei pot face teste într-un analizator special de teste ADN. În fiecare test ei stabilesc o secvență de elemente, codificată prin șirul  $P$ , și analizatorul verifică dacă această secvență apare în ADN, adică dacă șirul  $P$  este subșir al lui  $S$ .

Mostra este foarte mică, astfel cercetătorii vor putea realiza doar  $t$  teste. Ajută-i să facă teste corecte pentru a determina structura ADN-ului.

### Detalii de implementare

Trebuie să implementați funcția (metoda):

- `string analyse(int n, int t)` Această funcție trebuie să facă teste folosind funcția din bibliotecă (metoda) `make_test` și să rezolve ADN-ul.
- $n$ : lungimea ADN-ului,
- $t$ : numărul de teste permise.
  - funcția trebuie să returneze șirul soluție  $S$  care descrie ADN-ul.

### Funcții de bibliotecă

- `make_test(string p)`. Această funcție determină dacă șirul  $P$  este subșir al lui  $S$ .
  - $p$ : subșir pentru testare.
  - funcția returnează `true` dacă  $P$  este subșir al lui  $S$ , și `false` în caz contrar.

### Exemplu

Grader-ul face următorul apel:

- `analyse(3, 7)`. Lungimea șirului  $S$  este  $3$ , aveți permise  $7$  teste.

Programul vostru va face următoarele apeluri de funcții:

- `make_test("00")` returnează `false`.
- `make_test("01")` returnează `true`.
- `make_test("10")` returnează `true`.
- `make_test("11")` returnează `false`.
- `make_test("010")` returnează `false`.

Acum singurul șir posibil este "101", astfel funcția `analyse` returnează "101".

## Subcerințe

1. (11 puncte)  $n \leq 5, t = 31$ ,
2. (25 puncte)  $n \leq 100, t = 256$ ,
3. (64 puncte)  $n \leq 1000, t = 1024$ .

## Grader simplu

Grader-ul citește datele de intrare în următorul format:

- linia 1: string  $S$ ,
- linia 2: integer  $t$ .

## Notițe de limbaje

Vă rugăm să folosiți fișierele șablon furnizate de limbajul de programare utilizat.