



Martian DNA

Russia is known for its success in the field of space exploration. Recently Russian scientists analysed the samples of Martian soil, and found some strange molecule, which they believe can be some kind of DNA. This molecule has only two base elements instead of the four bases in the normal DNA. So the whole molecule can be described as string of zeroes and ones.

The scientists computed the length of the molecule, it is n base elements. Now they want to determine its structure, i.e. find the string of ones and zeroes S , that encodes the elements of the DNA molecule. In order to do it, they can make tests in a special DNA analyser. In each test they set a sequence of elements, encoded by string P , and the analyser checks if this sequence appears in the molecule, i.e. if the string P is a substring of S .

The sample is very fragile, so the scientists will be able to make only t tests. Help them make correct tests to determine the structure of the DNA sequence.

Implementation details

You should implement one function (method):

- `string analyse(int n, int t)` This function should make the tests using the library function (method) `make_test` and resolve the DNA.
 - n : length of the DNA sequence.
 - t : number of tests allowed.
 - The function should return the string S describing the DNA sequence.

Library functions

- `bool make_test(string p)`. This function checks whether the string P is a substring of S .
 - p : substring to test.
 - Returns `true` if P is a substring of S , `false` otherwise.

Please use the provided template files for details of implementation in your programming language.

Example

The grader makes the following function call:

- `analyse(3, 7)`. The length of string S is 3 , you are allowed to make 7 tests.

The contestant's program makes the following function calls:

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

Now the only possible string is "101", so the function `analyse` returns "101".

Subtasks

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

Sample Grader

The sample grader reads the input in the following format:

- Line 1: String S .
- Line 2: Integer t .