CyberChallenge.IT 2024
Programming Test

Pattern Recognition [100 points]

Problem Statement

Charlie Let me get this straight: in binary exploitation, we chuck massive strings into program input fields, and if we spot our string in memory where it shouldn’t be, it’s a problem, right?

Alan Spot on, Charlie! And there’s more: sometimes, we have to delicately craft these strings to precisely pinpoint our location within them.

Bob Can’t I just smash the keyboard randomly?

Alan And what if you need thousands of characters?

Bob Easy! I’ll make them all the same and randomly tweak a few at the end!

Alan You could end up in a tight spot with that, Bob... Typically, we resort to de Bruijn sequences, but that’s a tale for another time!

Bob I’m not interested in that. My method always works in practice! I can prove it!

Alan Alright, Bob, let’s play a game: I’ll give you a string $S$. How many strings $R$ exist such that you can cover all of $S$ using only copies of $R$?

Bob The problem does not even make sense, what do you mean by cover?

Alan I mean that I can recreate the string $S$ using copies of $R$, possibly overlapping them. For example, I can cover the string “$xyxyxy$” with “$xy$”, “$xyxy$” and, of course, “$xyxyxy$” itself. Is it clear now?

Bob Uhm, yes, it makes sense...

Alan takes a breath, hoping this will bring a momentary pause to Bob’s enthusiasm...

Problem Details

Input

The input consists of $3T + 1$ lines:

- Line 1: the number $T$ of testcases you would need to answer
- Lines 2, . . . , $3T + 1$: every group of 3 lines is formatted as follows
  - Line 1: two space separated integers, $N$ and $M$, respectively the length of the alphabet from which the string $S$ is sampled, and the length of the string $S$ itself
  - Line 2: a string of length $N$, representing the alphabet
  - Line 3: a string of length $M$, the actual string $S$

Output

The output consists of $T$ lines, each representing the answer to the corresponding testcase.

Scoring

Your program will be tested on a number of testcases grouped in subtasks. In order to obtain the score associated to a subtask, you need to correctly solve all its testcases.

- **Subtask 1** [20 points]: $1 \leq T \leq 100, N = 2, 1 \leq M \leq 12$
• **Subtask 2** [50 points]: \(1 \leq T \leq 100, 1 \leq N \leq 12, 1 \leq M \leq 500\)

• **Subtask 3** [30 points]: \(1 \leq T \leq 100, 1 \leq N \leq 20, 1 \leq M \leq 20000\)

**Examples**

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2 11</td>
<td></td>
</tr>
<tr>
<td>SG</td>
<td></td>
</tr>
<tr>
<td>GGGSGGSGGG</td>
<td></td>
</tr>
<tr>
<td>2 4</td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td></td>
</tr>
<tr>
<td>CCC</td>
<td></td>
</tr>
<tr>
<td>2 6</td>
<td></td>
</tr>
<tr>
<td>HK</td>
<td></td>
</tr>
<tr>
<td>HKHKHK</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Explanation**

The given input contains 3 different testcases:

- The first one, the string GGGSGGSGGG, can only be covered with the full string itself.
- The second one, CCCC, can be covered either with C, CC, CCC or CCCC.
- The third one, HKHKHK, can be covered with HK, HKHK or HKHKHK.