



Repeated String ☆

Problem

Submissions

Leaderboard

Discussions

Editorial

Editorial by [tunyash](#)

Let's calculate the number of occurrences of the whole string s in the first n letters of Lilah's infinite string, which is $\lfloor \frac{n}{|s|} \rfloor$.

We denote the number of occurrences of a in s to be A . Then the number of occurrences of a to the first $\lfloor \frac{n}{|s|} \rfloor \cdot |s|$ is $\lfloor \frac{n}{|s|} \rfloor \cdot A$. We've just calculated the number of a's in the first $n - (n \bmod |s|)$ letters of s . Now all we have to do is to notice that the letters with 0-indices $n - (n \bmod |s|), n - (n \bmod |s|) + 1, \dots, n - 1$ are respectively equal to letters with indexes $0, 1, \dots, (n \bmod |s|) - 1$. Then we can enumerate these indices and calculate B as the number of a's among them. Finally, the answer is $\lfloor \frac{n}{|s|} \rfloor \cdot A + B$.

In short, the number of a's in s * the number of whole occurrences of s plus the number of a's in the substring of s that remains gives the answer. The number of occurrences of $s = \text{int}(n/|s|)$. The length of the remaining substring is $n \% |s|$.

Set by [tunyash](#)

Problem Setter's code:

C++

```
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

int main()
{
    string s;
    cin >> s;
    ll n;
    cin >> n;
    ll ans = 0;
    int A = 0;
    int B = 0;
    for (int i = 0; i < (int)s.size(); i++)
    {
        if (s[i] == 'a') A++;
        if ((ll)i < n % (ll)s.size() && s[i] == 'a') B++;
    }

    cout << n / (ll)s.size() * (ll)A + (ll) B << endl;
    return 0;
}
```

STATISTICS

Difficulty: EasyTime Complexity: $O(|s|)$

Required Knowledge: strings

Publish Date: Sep 03 2016

Originally featured in [HourRank 12](#)

NEED HELP?

[View discussions](#)[View top submissions](#)

Feedback

Was this editorial helpful?

Yes

No