Mã bài: permrle

(Mã hóa hoán vị RLE)

(Google Code Jam 2008, Round 2)

*Time limit: 2s/test*

*Đoạn đầu tiên dùng để mô tả về phép mã hóa Run-Length Encoding (RLE). Nếu các bạn đã biết về nó, bạn có thể bỏ qua đoạn này.*

Mã hóa RLE là một phép nén xâu cơ bản, giúp thu được một xâu có độ dài nhỏ hơn và vẫn đảm bảo có thể khôi phục nguyên vẹn lại xâu gốc. Giả sử bạn cần mã hóa một xâu kí tự S:

* Chia S thành các xâu con, mỗi xâu chỉ chứa các kí tự giống nhau. Các xâu càng dài càng tốt (luôn mở rộng đến khi gặp một kí tự khác)
* Với mỗi xâu con độ dài k, mã hóa nó dưới dạng “X + c”, với X là số lần xuất hiện của kí tự c trong xâu. Ví dụ, một xâu con ‘aaaa’ có thể nén thành ‘4a’

Ví dụ, S = ‘aabcaaaa’ có thể được mã hóa thành RLE(S) = ‘2a1b1c4a’. Trong trường hợp này, S được chia thành 4 phần (4 xâu con).

Trong bài toán này, bạn áp dụng một biến thể của RLE có tên là permRLE. Cách làm như sau:

* Bạn được cho trước một số nguyên K, và dữ liệu đảm bảo độ dài xâu đã cho chia hết cho K
* Chọn một hoán vị P của {1,2,…,K}
* Chia S thành các xâu liên tiếp với độ dài K. Hoán vị từng xâu con với hoán vị P. Chẳng hạn, với xâu con ‘abcd’ và hoán vị {3,1,4,2}, bạn sẽ có xâu ‘cadb’
* Nối các xâu con lại với nhau theo đúng thứ tự chia ra. Chẳng hạn, với S = abcdefghijkl, bạn sẽ được xâu S’ = cadbgehfkilj
* Áp dụng phép mã hóa RLE trên xâu S’

Hãy tìm phương án hoán vị sao cho phép mã hóa RLE chia xâu S’ thành ít phần nhất.

Input:

* Dòng 1: 1 số nguyên K
* Dòng 2: xâu kí tự S, gồm N kí tự in thường. N chia hết cho K

Output:

* 1 số nguyên đưa ra số phần tối thiểu cần chia ra trong phép mã hóa tối ưu

Ràng buộc:

* Dễ: 2 <= K <= 5; 0 < N <= 1000
* Khó: 2 <= K <= 16; 0 < N <= 50000

Example

|  |  |
| --- | --- |
| Input | Output |
| 4 abcabcabcabc | 7 |
| 3 abcabcabcabc | 12 |