



Bogate łańcuchy



Limit czasu: 1 s

Limit pamięci: 256 MB

W tym problemie zajmiemy się bogactwem łańcuchów znaków, składających się z małych liter alfabetu łacińskiego. Przez *bogactwo łańcucha* s będziemy rozumieć ilość różnych znaków, które w nim choć raz występują – oznaczmy to przez $B(s)$. Na przykład $B(\text{"xx"}) = 1$, zaś $B(\text{"xyzxyz"}) = 4$.

Podciągiem łańcucha znaków s nazywamy jego dowolny niepusty spójny fragment. Ponieważ znaki w łańcuchu są numerowane od 0 do $\text{długość} - 1$, zatem podciąg jest jednoznacznie wyznaczony przez podanie numeru jego początkowego znaku (p) oraz numeru końcowego znaku (k), co będziemy oznaczać przez $s[p, k]$. Na przykład dla $s = \text{"bogactwo"}$ podciąg $s[0, 2]$ to "bog", natomiast podciąg $s[3, 5]$ to "act".

Rozważmy wszystkie podciągi łańcucha s . Oczywiście, dowolny taki podciąg posiada bogactwo z zakresu od 1 do $B(s)$. Dla każdej liczby n z tego zakresu należy znaleźć, ile podciągów posiada bogactwo dokładnie równe n .

Dane wejściowe

Jedyny wiersz danych wejściowych zawiera ciąg małych liter alfabetu łacińskiego s o długości z zakresu od jeden do trzystu tysięcy.

Wynik programu

Twój program powinien wypisać wiersz tekstu zawierający wartość $B(s)$, a następnie w kolejnych wierszach powinien wypisać ilość podciągów o danym bogactwie (kolejno od 1 do $B(s)$).

Przykład

Dla danych wejściowych:

alfa

program powinien wypisać:

3
4
3
3

Wartość $B(s)$ wynosi 3. Istnieją 4 podciągi o bogactwie 1:

$$s[0, 0] = "a", s[1, 1] = "l", s[2, 2] = "f", s[3, 3] = "a",$$

3 podciągi o bogactwie 2:

$$s[0, 1] = "al", s[1, 2] = "lf", s[2, 3] = "fa",$$

oraz 3 podciągi o bogactwie 3:

$$s[0, 2] = "alf", s[1, 3] = "lfa", s[0, 3] = "alfa".$$