Caracatiţă

Observam ca sirurile a, abb, cdd, opp, xzzyyy, qppaaammmm au ceva in comun. Mai exact, primul caracter se repeta o data, al doilea de 2 ori, al treilea de 3 ori, etc. Vom numi aceste siruri vrednice.

Se dau N siruri vrednice de caractere.

**Cerinta**

Se dau M query-uri. Fiecare query este un sir de caractere. Vrem sa aflam de cate ori fiecare query se “potriveste” in cele N siruri date.

Consideram ca un query q se potriveste intr-un sir s daca:

* q este subsir al lui s (abc este subsir al lui axbxcx)
* daca asupra lui q si s se aplica o operatie care restrange toate caracterele adiacente egale intr-unul singur, q este subsecventa a lui s (abc este subsecventa a lui xabcx) (q se potriveste in s de cate ori este subsecventa q restrans in s restrans)

De exemplu, abc se potriveste in abbccc, xaabbbccc o data, in abbcccaaaabbbbbccccc de 2 ori, si in abbdddcccc, axxbbbccc, abb niciodata.

**Date de intrare**

Pe prima linie a fisierului **caracatita.in** se afla numarul N. Urmeaza N linii care contin fiecare cate un sir vrednic de caractere, continand doar caracterele a-z.

Pe urmatoarea linie se afla numarul M. Urmeaza M linii care contin fiecare cate un sir de caractere a-z.

**Date de iesire**

Pe linia i din fisierul de iesire **caracatita.out** se va afla numarul de potriviri al query-ului i.

**Restrictii si precizari**

* 1 <= N <= 500
* 1 <= Lungimea unui sir vrednic <= 5050
* 1 <= M <= 100 000
* 2 <= Lungimea unui sir query <= 1000
* Pentru 20 % din teste, N <= 50
* Pentru alte 20 % din teste, M <= 1000

**Exemplu**

|  |  |
| --- | --- |
| caracatita.in | caracatita.out |
| 1  abbcccddd  4  abc  abcc  abccc  abcccc | 1  1  1  0 |
| 2  axxyyyzzzzpppp  xyy  5  axp  axz  xxyyyzzz  xyy  z | 0  0  1  2  1 |

Timp executie/test = 3s. Memorie disponibila/test = 512 MB