I split U in 3 (abc)

Valerio tocmai a găsit T șiruri de caractere $U_0, ..., U_{T-1}$, al i-lea dintre acestea fiind format din N_i litere mici latine.

Deoarece Valerio este foarte curios, el te întreabă, pentru fiecare $0 \leq i < T$, în câte moduri se poate împărți U_i în 3 șiruri (posibil goale) A, B, C astfel încât $U_i = A + B + C$, respectând condițiile din fiecare dintre următoarele scenarii:

- Scenariul `abc`: împărțirile trebuie să satisfacă $A \preceq B \preceq C$;
- Scenariul `acb`: împărțirile trebuie să satisfacă $A \preceq C \preceq B$;
- Scenariul `bac`: împărțirile trebuie să satisfacă $B \preceq A \preceq C$;
- Scenariul `bca`: împărțirile trebuie să satisfacă $B \preceq C \preceq A$;
- Scenariul `cab`: împărțirile trebuie să satisfacă $C \preceq A \preceq B$;
- Scenariul `cba`: împărțirile trebuie să satisfacă $C \preceq B \preceq A$.

unde $+$ denotă concatenarea de șiruri, iar $\preceq$ este relația de ordine lexicografică „mai mic sau egal”.

[1]

Implementare

Trebuie să trimiți un singur fișier cu extensia `.cpp`.

> ⇦ Printre fișierele atașate acestei probleme vei găsi un șablon `abc.cpp` cu o implementare exemplu.

> ⚠ Un singur fișier de intrare poate conține mai multe teste! Asigură-te că resetezi variabilele globale între apeluri.

Trebuie să implementezi următoarea funcție:

C++

```
void split(int N, string U,
           long long &abc, long long &acb, long long &bac,
           long long &bca, long long &cab, long long &cba);
```

- Întregul N reprezintă lungimea șirului U.
- Șirul U este unul dintre șirurile găsite de Valerio.
- Funcția trebuie să răspundă pentru fiecare scenariu, atribuind valori parametrilor corespunzători.
- Această funcție este apelată de T ori pe parcursul execuției programului.

Grader-ul va apela funcțiile și va afișa valorile returnate în fișierul de ieșire.

[1] Formal, date fiind două șiruri S și T, avem $S \preceq T$ dacă și numai dacă este adevărată una dintre următoarele:
- S este șirul vid;
- Niciunul dintre șiruri nu este vid, iar primul caracter din S apare înaintea primului caracter din T în alfabetul latin.
- Niciunul dintre șiruri nu este vid, primele caractere din cele două șiruri sunt identice și $S' \preceq T'$, unde S' și T' sunt șirurile obținute prin eliminarea primului caracter din S respectiv T.

Grader de probă

O versiune simplificată a grader-ului folosit la corectare este disponibilă în directorul asociat acestei probleme. O poți folosi pentru a-ți testa soluțiile local. Grader-ul dat ca exemplu citește datele de intrare din `stdin`, apelează funcția pe care trebuie să o implementezi și scrie în `stdout` în următorul format.

Fișierul de intrare este compus din $T + 1$ linii, unde T este numărul de teste, astfel:

- Linia 1: un întreg T.
- Linia $2 + i$ $(0 \leq i < T)$: un șir U_i.

Fișierul de ieșire este compus din T linii, conținând:

- Linia $1 + i$ $(0 \leq i < T)$: cele 6 răspunsuri date de program pentru testul i, în aceeași ordine în care sunt prezentate în enunț.

Constrângeri

- Lungimea totală a șirurilor dintr-un caz de test este cel mult 400 000.
- Fiecare dintre șiruri este nevid și format doar din caractere mici latine.

Punctaj

Programul tău va fi testat pe mai multe cazuri de test grupate pe subtasks. Scorul pentru un subtask este egal cu cel mai mic scor obținut pe unul dintre cazurile sale, înmulțit cu valoarea subtask-ului.

Scorul pentru un caz de test depinde de câte dintre cele șase scenarii sunt rezolvate corect, conform tabelului următor:

Scenarii rezolvate	0	1	2	3	4	5	6
Puncte	0	0.3	0.5	0.7	0.8	0.9	1

- **Subtask-ul 0 [0 puncte]**: Exemple.
- **Subtask-ul 1 [10 puncte]**: Singurul caracter din șir este `a`.
- **Subtask-ul 2 [10 puncte]**: Lungimea totală a șirurilor dintr-un test este cel mult 300.
- **Subtask-ul 3 [20 puncte]**: Lungimea totală a șirurilor dintr-un test este cel mult 15 000.
- **Subtask-ul 4 [60 puncte]**: Fără alte restricții suplimentare.

Exemple de intrare/ieșire

stdin	stdout
3 cafj aaaaaaa aabyyxll	4 2 5 2 3 2 8 8 8 8 8 8 21 10 9 1 8 1

Explicație

În **primul exemplu** împărțirile sunt:

1. $A =$ `""`; $B =$ `""`; $C =$ `"cafj"`.
2. $A =$ `""`; $B =$ `"c"`; $C =$ `"afj"`.
3. $A =$ `""`; $B =$ `"ca"`; $C =$ `"fj"`.
4. $A =$ `""`; $B =$ `"caf"`; $C =$ `"j"`.
5. $A =$ `""`; $B =$ `"cafj"`; $C =$ `""`.
6. $A =$ `"c"`; $B =$ `""`; $C =$ `"afj"`.
7. $A =$ `"c"`; $B =$ `"a"`; $C =$ `"fj"`.
8. $A =$ `"c"`; $B =$ `"af"`; $C =$ `"j"`.
9. $A =$ `"c"`; $B =$ `"afj"`; $C =$ `""`.
10. $A =$ `"ca"`; $B =$ `""`; $C =$ `"fj"`.
11. $A =$ `"ca"`; $B =$ `"f"`; $C =$ `"j"`.
12. $A =$ `"ca"`; $B =$ `"fj"`; $C =$ `""`.
13. $A =$ `"caf"`; $B =$ `""`; $C =$ `"j"`.
14. $A =$ `"caf"`; $B =$ `"j"`; $C =$ `""`.
15. $A =$ `"cafj"`; $B =$ `""`; $C =$ `""`.

Dintre acestea, următoarele sunt numărate pentru fiecare scenariu:

- Scenariul `abc`: împărțirile 1, 3, 4 și 11.
- Scenariul `acb`: împărțirile 2 și 5.
- Scenariul `bac`: împărțirile 1, 7, 8, 10 și 13.
- Scenariul `bca`: împărțirile 6 și 15.
- Scenariul `cab`: împărțirile 5, 12 și 14.
- Scenariul `cba`: împărțirile 9 și 15.