

I split U in 3 (abc)

Valerio tocmai a găsit T șiruri de caractere U_0, \dots, U_{T-1} , al i -lea dintre care este format din N_i caractere latine minuscule.

Cum Valerio e foarte curios, te roagă, pentru fiecare $0 \leq i < T$, să îi spui în câte moduri poate fi împărțit U_i în 3 șiruri de caractere posibil goale A, B, C astfel încât $U_i = A + B + C$, respectând constrângerile fiecăruia dintre următoarele scenarii:

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

unde $+$ înseamnă concatenarea șirurilor, iar \preceq este relația lexicografică de mai mic sau egal.¹

Implementare

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



Printre atașamentele acestei probleme, vei găsi un șablon `abc.cpp` cu o implementare exemplu.



Un singur fișier de intrare poate conține mai multe cazuri de test! Asigură-te că resetezi variabilele globale între diferite execuții.

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 la fiecare scenariu prin atribuirea valorilor parametrului corespunzător.
- Această funcție este apelată de T ori pe parcursul execuției programului tău.

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

¹Formal, date două șiruri S și T , avem $S \preceq T$ dacă și numai dacă una dintre următoarele afirmații este adevărată:

- S este șirul vid;
- Niciunul dintre șiruri nu este vid, iar primul caracter al lui S apare înaintea primului caracter al lui T în alfabetul latin.
- Niciunul dintre șiruri nu este vid, primele caractere ale celor 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 aferent acestei probleme. O poți folosi pentru a-ți testa local soluțiile. Grader-ul 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 format din $T + 1$ linii, unde T este numărul de cazuri de test, conținând:

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

Fișierul de ieșire este format din T linii, conținând:

- Linia $1 + i$ ($0 \leq i < T$): cele 6 răspunsuri date de programul tău pentru al i -lea caz de test, în aceeași ordine în care sunt prezentate în enunț.

Constrângeri

- Lungimea totală a șirurilor dintr-un caz de intrare este de maxim 400 000.
- Fiecare dintre șiruri este nevid și format din caractere latine minuscule.

Punctaj

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

Scorul pentru un caz de test depinde de câte dintre cele șase scenarii le rezolvi corect, conform următorului tabel:

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]:** Cazuri exemplu.
- Subtask-ul 1 [10 puncte]:** Singurul caracter din șir este `a`.
- Subtask-ul 2 [10 puncte]:** Lungimea totală a șirurilor dintr-un caz de intrare este de maxim 300.
- Subtask-ul 3 [20 puncte]:** Lungimea totală a șirurilor dintr-un caz de intrare este de maxim 15 000.
- Subtask-ul 4 [60 puncte]:** Fără constrângeri suplimentare.

Exemple de intrare/ieșire

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

Explicație

În primul caz de test, împărțirile sunt:

- | | | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|---------------------|
| 1. $A = ""$; | $B = ""$; | $C = \text{"cafj"}$. | 7. $A = \text{"c"}$; | $B = \text{"a"}$; | $C = \text{"fj"}$. |
| 2. $A = ""$; | $B = \text{"c"}$; | $C = \text{"afj"}$. | 8. $A = \text{"c"}$; | $B = \text{"af"}$; | $C = \text{"j"}$. |
| 3. $A = ""$; | $B = \text{"ca"}$; | $C = \text{"fj"}$. | | | |
| 4. $A = ""$; | $B = \text{"caf"}$; | $C = \text{"j"}$. | | | |
| 5. $A = ""$; | $B = \text{"cafj"}$; | $C = ""$. | | | |
| 6. $A = \text{"c"}$; | $B = ""$; | $C = \text{"afj"}$. | | | |

9. $A = "c"; \quad B = "afj"; \quad C = ""$.
10. $A = "ca"; \quad B = ""; \quad C = "fj"$.
11. $A = "ca"; \quad B = "f"; \quad C = "j"$.
12. $A = "ca"; \quad B = "fj"; \quad C = ""$.
13. $A = "caf"; \quad B = ""; \quad C = "j"$.
14. $A = "caf"; \quad B = "j"; \quad C = ""$.
15. $A = "cafj"; \quad B = ""; \quad C = ""$.

Dintre acestea, următoarele contează 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.