

I split U in 3 (abc)

Valerio je pronašao T stringova U_0, \dots, U_{T-1} , od kojih se i -ti sastoji od N_i latiničnih malih slova.

Pošto je Valerio jako znatiželjan, pita vas, za svako $0 \leq i < T$, na koliko načina se U_i može podeliti na 3 potencijalno prazna stringa A, B, C tako da važi $U_i = A + B + C$ pod ograničenjima svakog od sledećih scenarija:

- Scenario abc: podele moraju zadovoljiti $A \preceq B \preceq C$;
- Scenario acb: podele moraju zadovoljiti $A \preceq C \preceq B$;
- Scenario bac: podele moraju zadovoljiti $B \preceq A \preceq C$;
- Scenario bca: podele moraju zadovoljiti $B \preceq C \preceq A$;
- Scenario cab: podele moraju zadovoljiti $C \preceq A \preceq B$;
- Scenario cba: podele moraju zadovoljiti $C \preceq B \preceq A$.

gde $+$ označava spajanje (konkatenaciju) stringova, a \preceq je leksikografski manje-od-ili-jednako.¹

Имплементација

Moraš poslati samo jedan fajl sa ekstenzijom `.cpp`.

↩ Među priložima za ovaj zadatak, naći ćeš šablon `abc.cpp` sa primerom implementacije.



Jedan ulazni fajl može sadržati više test primera! Obavezno resetuj globalne promenljive između različitih pokretanja.

Moraš implementirati sledeću funkciju:

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);
```

- Ceo broj N predstavlja dužinu stringa U .
- String U je jedan od stringova koje je Valerio pronašao.
- Funkcija treba da odgovori na svaki scenario dodeljujući vrednosti odgovarajućem parametru.
- Ova funkcija se poziva T puta tokom izvršavanja tvog programa.

Grader će pozvati funkcije i ispisati vraćene vrednosti u izlazni fajl.

Пример грејдера

Pojednostavljena verzija gradera koja se koristi tokom ispravljanja dostupna je u direktorijumu vezanom za ovaj problem. Možeš je koristiti za lokalno testiranje svojih rešenja. Sample grader čita ulazne podatke iz `stdin`, poziva funkciju koju treba da implementiraš i ispisuje u `stdout` u sledećem formatu.

¹Formalno, za data dva stringa S i T , važi $S \preceq T$ ako i samo ako je jedno od sledećeg tačno:

- S je prazan string;
- Nijedan string nije prazan, i prvo slovo stringa S dolazi pre prvog slova stringa T u latiničnom alfabetu.
- Nijedan string nije prazan, prva slova oba stringa su ista i $S' \preceq T'$ gde su S' i T' stringovi dobijeni uklanjanjem prvog slova iz S odnosno T .

Ulaz se sastoji od $T + 1$ linija, gde je T broj test primera, i sadrži:

- Linija 1: ceo broj T .
- Linija $2 + i$ ($0 \leq i < T$): string U_i .

Izlaz se sastoji od T linija, i sadrži:

- Linija $1 + i$ ($0 \leq i < T$): 6 odgovora koje je tvoj program dao za i -ti test primer, u istom redosledu kako su predstavljani u opisu zadatka.

Ограничења

- Ukupna dužina stringova u jednom ulaznom slučaju je najviše 400 000.
- Svaki od stringova je neprazan i sastoji se od malih latiničnih slova.

Бодовање

Tvoj program će biti testiran na nekoliko test primera grupisanih u podzadatke. Bodovi za podzadatak su jednaki najgorim bodovima dobijenim na jednom od njegovih test primera, pomnoženim sa vrednošću podzadatka.

Bodovi za test primer zavise od toga koliko od šest scenarija rešiš tačno, prema sledećoj tabeli:

Rešenih scenarija	0	1	2	3	4	5	6
Bodova	0	0.3	0.5	0.7	0.8	0.9	1

- **Подзатак 0 [0 поена]:** Primeri.
- **Подзатак 1 [10 поена]:** Jedini karakter u stringu je a.
- **Подзатак 2 [10 поена]:** Ukupna dužina stringova u jednom ulaznom slučaju je najviše 300.
- **Подзатак 3 [20 поена]:** Ukupna dužina stringova u jednom ulaznom slučaju je najviše 15 000.
- **Подзатак 4 [60 поена]:** Bez dodatnih ograničenja.

Примери улаза/излаза

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

Објашњење

U prvom test primeru podele su:

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

Od ovih, sledeće se broje za svaki scenario:

- Scenario abc: podele 1, 3, 4 i 11.
- Scenario acb: podele 2 i 5.
- Scenario bac: podele 1, 7, 8, 10 i 13.
- Scenario bca: podele 6 i 15.
- Scenario cab: podele 5, 12 i 14.
- Scenario cba: podele 9 i 15.