

I split U in 3 (abc)

Oh crazy, Stoff hat *rein zufälligerweise* gerade T Strings U_0, \dots, U_{T-1} am Boden gefunden, wobei der i -te String aus N_i Buchstaben des kleinen lateinischen Alphabets besteht.

Weil Stoff ein sehr neugieriger Bünzli ist, fragt er dich für jeden String U_i wobei $0 \leq i < T$, wie viele verschiedene Möglichkeiten man hat, ihn in 3 (möglicherweise leere) Strings A, B, C zu unterteilen, sodass $U_i = A + B + C$ ist, unter den Einschränkungen jedes der folgenden Szenarien:

- Szenario abc: die Unterteilung erfüllt $A \preceq B \preceq C$;
- Szenario acb: die Unterteilung erfüllt $A \preceq C \preceq B$;
- Szenario bac: die Unterteilung erfüllt $B \preceq A \preceq C$;
- Szenario bca: die Unterteilung erfüllt $B \preceq C \preceq A$;
- Szenario cab: die Unterteilung erfüllt $C \preceq A \preceq B$;
- Szenario cba: die Unterteilung erfüllt $C \preceq B \preceq A$.

Die Operation $+$ bezeichnet die Verkettung und \preceq das lexikografische Kleiner gleich.¹

Implementierung

Du musst eine Datei mit der Endung `.cpp` einreichen.

⇐ In den Anhängen der Aufgabe ist eine Vorlage `abc.cpp` mit einer Beispielimplementierung.



Die Eingabe enthält mehrere Testfälle! Stelle sicher, alle globalen Variablen zwischen jedem Durchlauf zurückzusetzen!

Folgende Funktion muss implementiert werden:

C++	<pre>void split(int N, string U, long long &abc, long long &acb, long long &bac, long long &bca, long long &cab, long long &cba);</pre>
-----	---

- Die Zahl N ist die Länge des Strings U .
- Der String U ist einer der Strings, den Stoff *auf mysteriöse Art und Weise* gefunden hat.
- Die Funktion soll jedes Szenario beantworten, in dem sie den Wert des jeweiligen Parameters modifiziert.
- Diese Funktion wird während der Ausführung deines Programms T mal aufgerufen.

Der Grader wird die Funktion ausführen und die ausgegebenen Werte in die Ausgabe weiterleiten.

¹Stoffs formale Definition: Für zwei Strings S und T ist $S \preceq T$ genau dann wenn eine der folgenden Bedingungen wahr ist:

- S ist leer.
- Weder S noch T ist leer, und der erste Buchstabe von S kommt im lateinischen Alphabet vor dem ersten Buchstaben in T .
- Weder S noch T ist leer, und der erste Buchstabe von S und T ist gleich und $S' \preceq T'$, wobei S' und T' die Strings sind, die man durch Entfernen des ersten Buchstabens in jeweils S und T erhält.

Beispielgrader

Eine vereinfachte Version des Graders ist unter dem Verzeichnis dieser Aufgabe verfügbar. Du kannst sie verwenden, um deine Lösungen lokal zu prüfen. Dieser Beispielgrader liest die Eingabe von `stdin`, ruft die Funktion auf, die du implementieren sollst, und gibt die Ausgabe in `stdout` aus.

Die Eingabe besteht aus $T + 1$ Zeilen, wobei T die Anzahl der Testfälle ist.

- Zeile 1: Die Ganzzahl T .
- Zeile $2 + i$ ($0 \leq i < T$): Ein String U_i .

Die Ausgabe besteht aus T Zeilen mit folgendem Aufbau:

- Zeile $1 + i$ ($0 \leq i < T$): Enthält die 6 Antworten des Programms für den i -ten Testfall in der gleichen Reihenfolge in der Aufgabenbeschreibung.

Einschränkungen

- Die Gesamtlänge der Strings in einem Testfall ist maximal 400 000.
- Jeder String ist nicht leer und besteht aus kleinen Lateinbuchstaben.

Punktevergabe

Dein Programm wird auf mehreren Testfällen geprüft, welche in verschiedene Teilaufgaben unterteilt sind. Die Punktzahl einer Teilaufgabe ist gleich der tiefsten Punktzahl aller Testfälle, multipliziert mit dem Wert der Teilaufgabe.

Die Punktzahl hängt davon ab, wie viele der sechs Szenarien richtig beantwortet wurden, und zwar gemäss folgender Tabelle:

Gelöste Szenarien	0	1	2	3	4	5	6
Punkte	0	0.3	0.5	0.7	0.8	0.9	1

- **Teilaufgabe 0 [0 Punkte]**: Beispiele.
- **Teilaufgabe 1 [10 Punkte]**: Die Strings enthalten nur den Buchstaben a.
- **Teilaufgabe 2 [10 Punkte]**: Die Gesamtlänge der Strings ist maximal 300.
- **Teilaufgabe 3 [20 Punkte]**: Die Gesamtlänge der Strings ist maximal 15 000.
- **Teilaufgabe 4 [60 Punkte]**: Keine weiteren Einschränkungen.

Beispiele

stdin	stdout
3	4 2 5 2 3 2
cafj	8 8 8 8 8 8
aaaaaaa	21 10 9 1 8 1
aabyx1l	

Erklärung

Im **ersten Testfall** sind die möglichen Unterteilungen:

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

Von diesen zählen für jedes Szenarien die folgenden:

- Szenario **abc**: Unterteilungen 1, 3, 4 und 11.
- Szenario **acb**: Unterteilungen 2 und 5.
- Szenario **bac**: Unterteilungen 1, 7, 8, 10 und 13.
- Szenario **bca**: Unterteilungen 6 und 15.
- Szenario **cab**: Unterteilungen 5, 12 und 14.
- Szenario **cba**: Unterteilungen 9 und 15.