

Uwaga. Prosimy zwrócić uwagę na bardzo niski limit pamięci! W tym zadaniu wymagany jest algorytm działający w pamięci liniowej.

W pewnym spokojnym mieście otwierane jest nowe centrum handlowe. Klientów witać ma wielki neon, widoczny nawet z odległych krańców miasta. Niestety, coś poszło nie tak, pomyłki się spiętrzyły i na neonie znajduje się zupełnie inny napis niż powinien. Przyciąga, i owszem, rzesze ludzi, ale ci zamiast robić zakupy, śmieją się z pomyłki bliźnich.

Tak oczywiście pozostać nie może – neon trzeba przerobić na właściwy, minimalnym kosztem. Ile liter można pozostawić z obecnej wersji? Nie możemy zmieniać kolejności liter. Musimy zatem wybrać najdłuższy możliwy ciąg liter z oryginalnego napisu, który jest ciągiem w docelowym napisie.

Wejście

W pierwszym i jedynym wierszu standardowego wejścia znajdują się dwa ciągi znaków alfanumerycznych (małych i wielkich liter alfabetu angielskiego oraz cyfry), oddzielone od siebie pojedynczą spacją – obecny oraz docelowy napis. Długość każdego z ciągów nie przekracza 4000.

Wyjście

Dla każdego zestawu danych wypisz w osobnej linii maksymalną liczbę znaków, które można pozostawić.

Ocenianie

Możesz rozwiązać zadanie w kilku prostszych wariantach – niektóre grupy testów spełniają pewne dodatkowe ograniczenia. Poniższa tabela pokazuje, ile punktów otrzyma Twój program, jeśli przejdzie testy z takim ograniczeniem.

Dodatkowe ograniczenia	Liczba punktów
długość ciągów nie przekracza 40	25
długość ciągów nie przekracza 200	75

Przykłady

Wejście dla testu neo0a:

STOP SOTP

Wyjście dla testu neo0a:

3

Wyjaśnienie do przykładu: Możemy zostawić litery STP, bądź SOP.

Wejście dla testu neo0b:

YGU FY

Wyjście dla testu neo0b:

1

Wejście dla testu neo0c:

jd67H 8j9hy67h

Wyjście dla testu neo0c:

3

Wejście dla testu neo0d:

a b

Wyjście dla testu neo0d:

0

Pozostałe testy przykładowe

- test neo0e: Napisy na wejściu mają po 4000 znaków. Obecny napis to abcabcabc...a, natomiast docelowy to cbacbacba...c.