

Kontrprzykład (kontrprzykład)

Limit pamięci: 128 MB

Limit czasu: 1.00 s

Podciągami słowa A nazwiemy każde słowo, które możemy uzyskać usuwając niektóre litery z A i odczytując pozostałe nie zmieniając ich kolejności. Dla przykładu, `bef` jest podciągami słowa `abcdefg`, zaś `gf` nie jest.

Jasio na lekcji informatyki dowiedział się jak rozwiązać problem znalezienia najdłuższego wspólnego podciągu dwóch słów, czyli najdłuższego słowa, które jest podciągami obu danych słów. Po powrocie do domu chłopak bez chwili zastanowienia zaczął rozwiązywać zadanie domowe, polegające na zaimplementowaniu poznanego algorytmu. Udało mu się napisać poprawny kod, chociaż, co później wypomniała mu nauczycielka, nieoptymalny czasowo ¹.

```
string dp[N+1][N+1];
string lcs(string a, string b) {
    for (int i = 1; i <= a.size(); ++i) {
        for (int j = 1; j <= b.size(); ++j) {
            if (a[i-1] == b[j-1]) dp[i][j] = dp[i-1][j-1] + a[i-1];
            else if (dp[i-1][j].size() >= dp[i][j-1].size()) dp[i][j] = dp[i-1][j];
            else dp[i][j] = dp[i][j-1];
        }
    }
    return dp[a.size()][b.size()];
}
```

Na liście zadań znajdowało się również zadanie bonusowe, które polegało na zaimplementowaniu algorytmu znajdującego najdłuższy wspólny podciąg trzech ciągów. Jasio stwierdził, że to nic trudnego – w końcu może wykorzystać swoją poprzednią implementację i najpierw znaleźć najdłuższy wspólny podciąg dwóch pierwszych słów, a następnie najdłuższy wspólny podciąg znalezionej słowa oraz słowa trzeciego.

```
string lcs3(string a, string b, string c) {
    return lcs(lcs(a, b), c);
}
```

Testy w tym zadaniu były dość słabe i okazało się, że (oczywiście błędne) rozwiązanie Jasia przeszło wszystkie testy. Chłopak chwali się, że jest mistrzem algorytmów tekstowych. Jako osoba z klasy Jasia, która poprawnie rozwiązała zadanie bonusowe, czujesz, że trzeba go sprowadzić na ziemię.

Napisz program, który dla danych liczb N , L , J , wyznaczy trzy słowa o długości N , składające się z małych liter alfabetu angielskiego, których najdłuższy wspólny podciąg ma L liter, a program Jasia zwróci słowo o długości J , lub stwierdź, że nie istnieją takie trzy słowa.

Wejście

W pierwszym (jedynym) wierszu wejścia znajdują się trzy liczby całkowite N , L , J , pooddzielane pojedynczymi odstępami.

Wyjście

Jeżeli istnieje kontrprzykład zgodny z treścią zadania, wypisz trzy słowa w trzech wierszach. Kolejność ma znaczenie! W przeciwnym wypadku wypisz jeden wiersz zawierający słowo `ACCEPTED`.

Jeżeli istnieje wiele możliwych poprawnych odpowiedzi, Twój program może wypisać dowolną z nich.

¹Implementacja Jasia działa w złożoności $O(n^3)$ zamiast $O(n^2)$ ze względu na kopiowanie słów. Dla odpowiednio dobranych danych może się zdarzyć, że tablica `dp` będzie wypełniona $\Omega(n^2)$ słowami o długości $\Omega(n)$.

Ograniczenia

$1 \leq N \leq 100, 0 \leq J \leq L \leq N.$

Przykład

Wejście

10 5 3

Wyjście

eckoyzleak
kozalkoeox
kixdoezakw

Wyjaśnienie

Program Jasia obliczy, że najdłuższy podciąg dwóch pierwszych słów to kozle, a następnie, że najdłuższy wspólny podciąg kozle i trzeciego słowa to koz. Najdłuższy wspólny podciąg ma 5 liter, na przykład kozak.

Wejście

10 10 0

Wyjście

ACCEPTED

Ciąg permutacji (permutacje)

Limit pamięci: 64 MB

Limit czasu: 1.00 s

Jasio przygotowuje się do zawodów Bajtockiej Olimpiady Informatycznej Juniorów. Ostatnio nauczył się funkcji `next_permutation`, która generuje następną leksykograficznie permutację zadanego ciągu obiektów. W tym zadaniu będziemy rozpatrywali jedynie permutacje zbioru $\{1, 2, \dots, N\}$, a więc dowolne ustawienie elementów (każdy element występuje dokładnie raz) tego zbioru w ciąg.

To *bardzo fajnie*, pomyślał Jasio – będzie można łatwo napisać rozwiązanie naiwne testujące wszystkie możliwości kolejności wykonania jakichś akcji. Jasiowi jednak nie do końca podoba się kolejność generowanych permutacji. Przykładowo: permutacja $(1, 5, 4, 3, 2)$ bezpośrednio poprzedza $(2, 1, 3, 4, 5)$. Jasio wolałby, aby sąsiednie dwie permutacje różniły się możliwie mało, najlepiej zamianą dokładnie dwóch sąsiednich elementów. Wtedy w jego naiwnych rozwiązaniach często możliwa jest tylko drobna aktualizacja wyniku poprzedniej permutacji, żeby uzyskać wynik dla następnej, zamiast obliczać ów wynik od nowa. Czy pomożesz mu dobrać lepszą kolejność, tak żeby wygenerować wszystkie permutacje, każdą dokładnie jeden raz, ale żeby każde dwie sąsiednie permutacje różniły się jedynie zamianą pewnych dwóch sąsiednich elementów?

Napisz program, który wczyta liczbę naturalną N , wygeneruje wszystkie $N!$ różnych permutacji w odpowiedniej kolejności i wypisze wynik na standardowe wyjście.

Wejście

W pierwszym (jedynym) wierszu wejścia znajduje się jedna liczba naturalna N , określająca długość każdej permutacji.

Wyjście

Twój program powinien wypisać na wyjście dokładnie $N!$ permutacji zbioru $\{1, 2, \dots, N\}$, po jednej w wierszu. Każda permutacja powinna być wypisana za pomocą N parami różnych liczb od 1 do N włącznie, podzielanych pojedynczymi odstępami.

Wypisane permutacje mają być parami różne, a każde dwie sąsiednie wypisane permutacje mają się różnić pozycją dokładnie dwóch sąsiednich elementów.

Jeżeli istnieje wiele możliwych odpowiedzi, Twój program może wypisać dowolną z nich.

Ograniczenia

$$2 \leq N \leq 9.$$

Przykład

Wejście

3

Wyjście

1 2 3
1 3 2
3 1 2
3 2 1
2 3 1
2 1 3

Wyjaśnienie

Możliwe są różne poprawne wyjścia. Podajemy tutaj jedynie przykładowe poprawne rozwiązanie.

Suma median (suma-median)

Limit pamięci: 64 MB

Limit czasu: 1.00 s

Jasio nauczył się ostatnio algorytmu *magicznych piątek*. Jest to algorytm pozwalający w czasie liniowym, bez użycia typowego sortowania, znaleźć k -ty najmniejszy element w nieposortowanym ciągu obiektów (problem selekcji).

– *Imponujące!* – pomyślał Jasio, gdy zrozumiał ideę algorytmu.

Nie będziemy tutaj przytaczać dokładnie całego algorytmu, ale z grubsza polega on na podzieleniu elementów ciągu długości N na $\frac{N}{5}$ grup po pięć elementów. W każdej z tych grup obliczamy medianę. Potem należy zrobić coś jeszcze, ale nie tego dotyczy zadanie.

Jasio zafascynował się konceptem dzielenia na grupy po pięć obiektów i liczenia z nich mediany do tego stopnia, że zupełnie zapomniał o problemie selekcji, który początkowo chciał rozwiązać. Chciałby tak pogrupować elementy swojego ciągu (umieścić każdy element w jakiejś pięcioelementowej grupie), żeby suma median w grupach była największa możliwa. Pomożesz?

Napisz program, który: wczyta elementy ciągu Jasia, wyznaczy optymalny podział na pięcioelementowe grupy, który maksymalizuje sumę median grup i wypisze wynik na standardowe wyjście.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N , określająca liczbę elementów ciągu. W drugim (ostatnim) wierszu wejścia znajduje się ciąg N liczb naturalnych A_i , pooddzielanych pojedynczymi odstępami. Są to elementy ciągu Jasia.

Wyjście

W pierwszym (jedynym) wierszu wyjścia powinna się znaleźć jedna liczba całkowita – największa możliwa suma median w wybranych grupach pięcioelementowych.

Ograniczenia

$5 \leq N \leq 500\,000$, N jest postaci $5k$ dla pewnego $k \in \mathbb{N}$, $1 \leq A_i \leq 10^9$.

Przykład

Wejście

```
10
6 1 5 20 9 7 1 3 4 6
```

Wyjście

```
12
```

Wyjaśnienie

W tym przypadku optymalnie byłoby na przykład umieścić elementy o wartościach (3, 4, 5, 6, 6) w jednej grupie, zaś pozostałe elementy w drugiej grupie. Wtedy mediana jednej grupy to 5, a drugiej grupy to 7.