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 $\{1, 2, \ldots, N\}$, 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 wygerować 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 $\{1, 2, \ldots, N\}$, po jednej w wierszu. Każda permutacja powinna być wypisana za pomocą N parami różnych liczb od 1 do N włącznie, poodzielanych 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	Wyjście	Wyjaśnienie
3	1 2 3 1 3 2 3 1 2 3 2 1 2 3 1 2 1 3	Możliwe są różne poprawne wyjścia. Podajemy tutaj jedynie przykładowe poprawne rozwiązanie.