

Zadanie: PER

Permutacje antyarytmetyczne



III OI, etap trzeci. Plik źródłowy per.* Dostępna pamięć: 32 MB.

Permutację p_0, p_1, \dots, p_{n-1} liczb naturalnych $0, 1, \dots, n-1$ nazywamy permutacją antyarytmetyczną, jeżeli nie występuje w niej żaden trzywyrazowy ciąg arytmetyczny, tj. nie istnieją takie indeksy $i < j < k$, że liczby p_i, p_j, p_k (w tej kolejności) tworzą ciąg arytmetyczny. Na przykład ciąg $3, 1, 0, 4, 2$ jest permutacją antyarytmetyczną liczb $0, 1, 2, 3, 4$. Natomiast ciąg $0, 5, 4, 3, 1, 2$ nie jest permutacją antyarytmetyczną, ponieważ jej wyrazy – pierwszy, piąty i szósty $(0, 1, 2)$ tworzą ciąg arytmetyczny (także wyrazy drugi, czwarty i piąty – $5, 3, 1$ oraz drugi, trzeci i czwarty – $5, 4, 3$ tworzą ciągi arytmetyczne).

Napisz program, który znajdzie dowolną permutację arytmetyczną liczb $0, 1, \dots, n-1$.

Wejście

W pierwszym i jedynym wierszu standardowego wejścia znajduje się pojedyncza liczba całkowita n ($3 \leq n \leq 100\,000$).

Wyjście

W n kolejnych wierszach standardowego wyjścia należy wypisać n różnych liczb ze zbioru $\{0, 1, \dots, n-1\}$, każdą w osobnym wierszu. Liczby te powinny tworzyć permutację antysymetryczną liczb $0, 1, \dots, n-1$.

Przykład

Dla danych wejściowych:

5

poprawnym wynikiem jest:

3
1
0
4
2