Zadanie: OCS
Ostatnia cyfra silni

Podstawy C++, ILO Białystok. Dostępna pamięć: 32 MB. *01.09.2019*

Niech n będzie nieujemną liczbą całkowitą. Liczbę $n!$ (czytaj n – *silnia*) definiuje się następująco: Jeśli $n \leq 1$, to $n! = 1$. Dla $n > 1$, $n!$ jest równe iloczynowi wszystkich liczb od 1 do n, czyli $n! = 1 * 2 * \ldots * n$. Na przykład $4! = 1 * 2 * 3 * 4 = 24$.

Liczba $n!$ bardzo szyblo rośnie: dla $n = 200$ $n! =$ 7886578673647905035523632139321850622951359776871732632
947425332443594499634033429203042840119846239041772121389196388302576427902426371050619266249528
299311134628572707633172373969889439224456214516642402540332918641312274282948532775242424075739 0
32403212574055795686602260319041703240623517008587961789222227896237038973747200000000000000000000
00000000000000000000000000000.

Na szczęście nie musisz jej wyliczać. Wystarczy, że dla danego n podasz ostatnią cyfrę wartości $n!$.

Napisz program który:

- wczyta ze standardowego wejścia nieujemną liczbę całkowitą n,
- policzy cyfrę jedności w zapisie dziesiętnym liczby n!,
- wypisze wynik na standardowe wyjście.

Wejście

Pierwszy i jedyny wiersz standardowego wejścia zawiera dokładnie jedną nieujemną liczbę całkowitą n, $(0 \leq n \leq 10^9)$.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia Twój program powinien zapisać dokładnie jedną cyfrę równą cyfrze jedności w zapisie dziesiętnym liczby $n!$.

Przykład 1

Dla danych wejściowych:
200

poprawnym wynikiem jest:
0

Przykład 2

Dla danych wejściowych:
3

poprawnym wynikiem jest:
6

Przykład 3

Dla danych wejściowych:
2

poprawnym wynikiem jest:
2