11308. ICPC строки

Определим ICPC-строку как строку, составленную из букв *I*, *C*, *P*, *C*, такую, что каждые 4 подряд идущие символа содержат ровно две буквы *C*, одну букву *I* и одну букву *P*.

Для заданного *n* найдите количество различных ICPC-строк длины *n*.

**Вход.** Ввод содержит одно целое число *n* (4 ≤ *n* ≤ 1000) – длину строки.

**Выход.** Выведите одно целое число – количество ICPC-строк длины *n*.

|  |  |
| --- | --- |
| **Пример входа** | **Пример выхода** |
| 5 | 12 |

## РЕШЕНИЕ

**комбинаторика**

# Анализ алгоритма

Число различных анаграмм слова “ICPC” равно 4! / (2! \* 2!) = 12.

Пятый символ ICPC-строки однозначно определяется первыми четырьмя. Действительно, каждые 4 подряд идущие буквы должны содержать две буквы *C*, одну *I* и одну *P*. Следовательно, пятый символ обязан совпадать с первым. По той же причине шестой символ совпадает со вторым, седьмой – с третьим и так далее.

Таким образом, любая ICPC-строка полностью задаётся первыми четырьмя символами. Следовательно, количество ICPC-строк длины *n* не зависит от *n* и равно числу анаграмм слова “ICPC”, то есть 12.

**Пример**

Рассмотрим примеры ICPC-строк:

* **ICPC**ICPCICPCICPCICPC…
* **CCPI**CCPICCPICCPICCPI…
* **PCIC**PCICPCICPCICPCIC…

**Реализация алгоритма**

Для решения задачи достаточно вывести одно число 12.

printf("12\n");