# Знакомство с циклом for

# Факториал

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Напишите программу, которая считывает целое неотрицательное число *n* и выводит его факториал. Факториал числа *n* обозначается *n*! и равен произведению всех натуральных чисел от 1 до *n*:

https://contest.yandex.ru/testsys/tex/render/biEgPSAxXHRpbWVzMlx0aW1lc1xsZG90c1x0aW1lcyBu.png

Факториал нуля принят равным 1.

## Формат ввода

Число n.

## Формат вывода

Одно число — https://contest.yandex.ru/testsys/tex/render/biE=.png .

## Пример

| Ввод | Вывод |
| --- | --- |
| 4 | 24 |

# Ждём потепления

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Программа считывает максимальную дневную температуру в градусах за первый, второй и т. д. дни наблюдений до тех пор, пока не встретится температура не меньше 22.0℃. Выводится количество полных недель, которое пришлось прождать до этого дня. Сам этот день в период ожидания не включается.

## Формат ввода

В первой строке записана максимальная температура в первый день наблюдений, во второй строке — температура во второй день наблюдений и т.д.

## Формат вывода

Количество полных недель, прошедших до дня потепления.

## Пример

| Ввод | Вывод |
| --- | --- |
| 15.9  16.3  16.8  19.1  20.5  20.1  21.8  22.4 | 1 |

# Делите ли

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Простыми называются такие натуральные числа, у которых ровно два делителя: 1 и само это число (они должны отличаться, то есть 1 — не простое число). Простые числа и вообще разложение чисел на множители долгое время были предметом абстрактной игры ума математиков, но в настоящее время некоторые математические понятия, связанные с разложением на множители, играют важнейшую роль в шифровании трафика в интернете.

Напишите программу, которая считывает одно натуральное число и выводит на первой строке все делители этого числа в порядке возрастания, разделённые пробелами, а на второй — «ПРОСТОЕ» или «НЕТ» в зависимости от того, простым было введённое число или нет.

## Формат ввода

Одно натуральное число.

## Формат вывода

В первой строке — все делители введенного числа, записанные через пробел (перед первым и после последнего делителя пробелов нет).  
Во второй — сообщение «ПРОСТОЕ» или «НЕТ».

## Пример

| Ввод | Вывод |
| --- | --- |
| 12 | 1 2 3 4 6 12  НЕТ |

# Пирамида

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Выведите пирамиду из символов «\*» заданной высоты.

## Формат ввода

Одно целое число — высота пирамиды.

## Формат вывода

Пирамида в формате, аналогичном показанному в примере.

## Пример

| Ввод | Вывод |
| --- | --- |
| 3 | \*  \*\*\*  \*\*\*\*\* |

# Обратный отсчёт

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Любопытно, что впервые обратный отсчёт перед запуском ракеты был использован в немом научно-фантастическом фильме «Женщина на Луне». Драматический приём оказался настолько удачным, что прижился в реальной практике космонавтики.

Напишите программу, которая ведёт обратный отсчёт.

## Формат ввода

Вводится одно целое число n — количество секунд, оставшееся до запуска.

## Формат вывода

Для каждой секунды от n-й до нулевой последовательно выведите: «Осталось секунд: <количество оставшихся секунд>». После этого выведите: «Пуск».

Если n<0, то это значит, что с пуском мы опаздываем: в этом случае выводите «Пуск» немедленно, не тратя времени на обратный отсчёт.

## Пример

| Ввод | Вывод |
| --- | --- |
| 5 | Осталось секунд: 5  Осталось секунд: 4  Осталось секунд: 3  Осталось секунд: 2  Осталось секунд: 1  Осталось секунд: 0  Пуск |

# Дзета-функция Римана

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Сосчитайте сумму обратных квадратов (обратный квадрат числа k — это число 1/k2) первых n натуральных чисел (n вводит пользователь). Поделите квадрат числа https://contest.yandex.ru/testsys/tex/render/XHBp.png (где https://contest.yandex.ru/testsys/tex/render/XHBp.png — отношение длины окружности к её диаметру) на эту сумму и выведите результат. Точность ответа должна быть не менее 10 знаков после запятой.

Взгляните, к какому числу приближается результат с ростом n.

## Формат ввода

Вводится одно натуральное число n, n≤1300000.

## Формат вывода

Выводится одно действительное число, согласно условию.

## Пример

| Ввод | Вывод |
| --- | --- |
| 1000 | 6.003647956020601 |

## Примечания

В качестве числа https://contest.yandex.ru/testsys/tex/render/XHBp.png возьмите значение 3.141592653589793 или еще более точное.

# Шварценеггер против Годзиллы

Kлассная работа

макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Арнольд Шварценеггер стреляет в ужасного монстра Годзиллу из дробовика. Нужно найти общую величину урона, нанесённого Годзилле выстрелом.

Подсказка: возможно, для быстрой работы программы вам пригодится алгоритм Евклида.

## Формат ввода

Сначала вводится количество дробинок.  
Затем урон от каждой дробинки. Урон от каждой дробинки выражается простой дробью, её числитель и знаменатель вводятся на отдельных строках.

## Формат вывода

Суммарный урон, выраженный простой несократимой дробью с дробной чертой между числителем и знаменателем.

## Пример

| Ввод | Вывод |
| --- | --- |
| 3  1  60  1  30  1  100 | 3/50 |