

Задача А. Кружок английского (только для 7-8 классов!)

Имя входного файла: `english.in`
Имя выходного файла: `english.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Марина Андреевна ведет кружок английского языка у 7Б класса. Сегодня на занятие пришли Аня, Борис, Сергей и Диана. Для выполнения очередного упражнения Марина Андреевна хочет разбить учеников на пары.

Но сделать это нужно не абы как! Дело в том, что у всех учеников разный уровень знания английского языка. Марина Андреевна оценивает уровень Ани в a баллов, уровень Бориса в b баллов, Сережи — c баллов, а Дианы — d баллов. Упражнение принесет пользу только в том случае, когда суммарный уровень учеников в паре не меньше, чем L .

Марина Андреевна хочет, чтобы занятие принесло детям как можно больше знаний. Помогите ей и определите какое наибольшее количество «эффективных» пар она может получить.

Формат входных данных

В первой строке входного файла `english.in` даны 5 целых чисел, разделенных одним пробелом — a , b , c , d и L . Все числа от 1 до 10.

Формат выходных данных

В выходной файл `english.out` выведите одно число — наибольшее число «эффективных» пар: 0, 1 или 2.

Примеры

<code>english.in</code>	<code>english.out</code>
5 5 2 3 7	2
3 4 1 2 8	0

Замечание

В первом примере ответ можно получить если разбить детей на пары, например, следующим образом: Аня с Сережей, а Борис с Дианой.

Задача В. Игра со строкой

Имя входного файла: `abc.in`
Имя выходного файла: `abc.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Маленькая Элизабет любит играть со строками. У нее есть строка длины n , полностью состоящая из букв «а». Она делает со строкой следующие действия:

- Если первая буква в строке «а», то она дописывает в конец строки «bc», после чего удаляет из строки первые 2 буквы.
- Если первая буква в строке «b», то она дописывает в конец строки «a», после чего также удаляет из строки первые 2 буквы.
- Если первая буква в строке «с», то она дописывает в конец строки «aa», после чего опять удаляет из строки первые 2 буквы.

Элизабет останавливается после того, как у нее получится строка, состоящая из одной буквы «а». Например, при $n = 4$, для достижения цели ей требуется 6 шагов:

`aaaa` → `aabc` → `bcbc` → `bca` → `aa` → `bc` → `a`

Элизабет проверила все значения n от 2 до 100 и у нее всегда в итоге получалась одна буква «а». Теперь она хочет узнать сколько она потратила шагов для каждого из n , но выполнять все операции заново для нее слишком утомительно, поэтому она попросила Вас написать для этого программу.

Формат входных данных

В единственной строке входного файла `abc.in` дано одно целое число n ($2 \leq n \leq 100$).

Формат выходных данных

В выходной файл `abc.out` выведите одно число — количество требуемых шагов.

Примеры

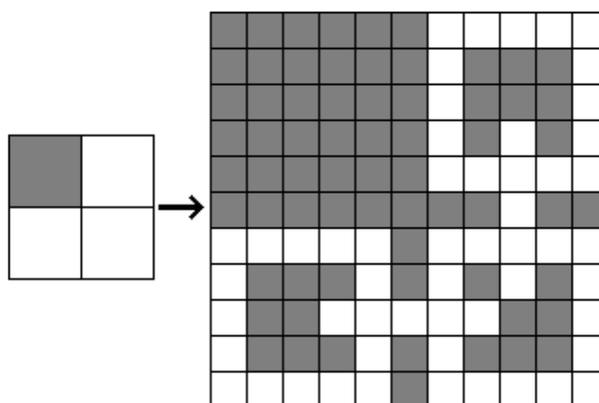
<code>abc.in</code>	<code>abc.out</code>
4	6
3	24

Задача С. Генератор кроссвордов

Имя входного файла: crossword.in
Имя выходного файла: crossword.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Ишполит, полиглот и эрудит, хочет составить самый большой в мире кроссворд из пятибуквенных слов, чтобы попасть в книгу рекордов Гиннеса. Он уже придумал, каким образом будет устроена сетка его кроссворда. А именно:

Ишполит составил *сжатое* описание сетки кроссворда — таблицу размера $n \times m$, состоящую из черных и белых клеток. В полной версии кроссворда каждая белая клетка сжатого описания должна быть заменена на 4 пятибуквенных слова, образующих квадрат 5×5 . Черные клетки не создают пятибуквенных слов. В самом конце все квадраты 5×5 , которые в сжатом описании имели общее ребро, должны быть соединены дополнительным пятибуквенным словом. Пример подобного «разжатия» кроссворда можно видеть на рисунке:



Напишите для Ишполита программу, которая по сжатому описанию построит полную версию сетки кроссворда, чтобы он наконец смог заняться ее заполнением.

Формат входных данных

В первой строке входного файла `crossword.in` даны два целых числа n и m ($1 \leq n, m \leq 40$).

В следующих n строках содержится ровно по m символов в каждой — сжатое описание сетки кроссворда. При этом «.» означает белую клетку, а «#» — черную.

Формат выходных данных

В выходной файл `crossword.out` выведите $6n - 1$ строк по $6m - 1$ символов в каждой — итоговую сетку кроссворда. Для белых клеток используйте «.», а для черных — «#». Смотрите пример для лучшего понимания.

Примеры

crossword.in	crossword.out
2 2	#####.....
#.	#####.###.
..	#####.###.
	#####.#.#.
	#####.....
	#####.##
#.....
	.###.#.#.#.
	.##.....##.
	.###.#.###.
#.....

Задача D. Хитрая перестановка

Имя входного файла: `tricky.in`
Имя выходного файла: `tricky.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Перестановкой длины n называется последовательность из n чисел, в которой каждое из чисел от 1 до n встречается ровно 1 раз. Например, $(3, 4, 5, 1, 2)$ и $(1, 2)$ — перестановки, а $(1, 4, 3)$ и $(2, 1, 3, 2)$ — нет.

Назовем перестановку *хитрой*, если для любых двух соседних чисел в перестановке разница между ними больше или равна минимальному из них. Например, перестановка

$(3, 1, 2, 4)$

является хитрой, поскольку $|3 - 1| \geq \min(3, 1)$, $|1 - 2| \geq \min(1, 2)$ и $|2 - 4| \geq \min(2, 4)$.

Дано число n , найдите хитрую перестановку длины n .

Формат входных данных

В единственной строке входного файла `tricky.in` дано одно целое число n ($1 \leq n \leq 100$).

Формат выходных данных

В выходной файл `tricky.out` выведите n чисел — искомую перестановку. Если решений несколько — выведите любое. Гарантируется, что ответ всегда существует.

Примеры

<code>tricky.in</code>	<code>tricky.out</code>
4	3 1 2 4

Задача Е. Двухъядерный процессор

Имя входного файла: `coreduo.in`
Имя выходного файла: `coreduo.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Петя получил стипендию и на радостях купил для своего компьютера новый процессор. Да не простой, а двухъядерный! «Теперь все программы будут работать в 2 раза быстрее», — подумал Петя. Но не тут то было!

Как оказалось, программы быстрее выполняться не стали, зато появилась возможность запустить две программы одновременно. Поэтому, например, две программы, которые на обычном процессоре выполняются 10 и 20 секунд соответственно — на новом двухъядерном процессоре можно выполнить за 20 секунд. Процессор устроен таким образом, что запущенную на одном ядре программу нельзя прерывать (и возобновлять выполнение через некоторое время) или переносить вычисления на другое ядро. То есть, программа выполняется на ядре все свое время работы одним отрезком времени, а следующую программу на данном ядре можно запустить только после окончания выполнения предыдущей.

У Пети есть n программ, для каждой из них известно время выполнения t_i . Петя хочет распределить программы между ядрами процессора так, чтобы все программы были выполнены как можно быстрее. Программы можно выполнять в любом порядке.

Например, программы с временами выполнения 5, 3, 7, 9, 12 можно выполнить за 19 секунд, если на первом ядре выполнять первую, вторую и четвертую программы, а на втором — остальные две. При этом нет способа распределить программы так, чтобы успеть их выполнить за 18 секунд.

Помогите Пете решить его проблему!

Формат входных данных

В первой строке входного файла `coreduo.in` дано число n ($1 \leq n \leq 100$). Во второй строке даны n чисел, разделенных пробелом. i -ое из них равно времени выполнения i -ой программы — t_i ($1 \leq t_i \leq 1000, 1 \leq i \leq n$).

Формат выходных данных

В выходной файл `coreduo.out` выведите одно число — наименьшее время выполнения всех программ.

Примеры

<code>coreduo.in</code>	<code>coreduo.out</code>
5	19
5 3 7 9 12	

Замечание

Решения, решающие задачу для $n \leq 20$ будут оценены в 50 баллов из 100.

Решения, решающие задачу для $n \leq 40$ будут оценены в 70 баллов из 100.

Задача F. Список файлов (только для 9-11 классов!)

Имя входного файла: list.in
Имя выходного файла: list.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В задаче описано упрощенное поведение программы — в реальности все может быть немного по-другому. Используйте условие задачи как формальный документ.

В Unix-подобных операционных системах (таких как, например, Linux) команда «ls -C» выводит на экран список всех файлов текущего каталога в многоколоночном режиме. При этом список всех файлов в каталоге сортируется в алфавитном порядке, после этого разбивается на несколько (скажем, k) интервалов одинаковой длины (кроме, возможно, последнего) и выводится на экран в k колонок. Ширина колонки равна максимальной длине имени файла, входящего в соответствующий интервал. Колонки отделяются друг от друга одним столбцом из пробелов. Пример подобного вывода:

```
a      accd e t
aba    b   f wtrt
abacaba db   k
```

Ширина вывода для k колонок — сумма ширин всех колонок плюс $(k - 1)$. В приведенном выше примере ширина вывода равна 19. Команда «ls -C» разбивает список файлов на интервалы таким образом, чтобы итоговая ширина вывода не превышала ширины экрана w . При этом из всех возможных вариантов разбиения выбирается такой, чтобы длина интервалов (что то же самое, что число строк в выводе или высота колонок) была минимальной.

По списку длин имен файлов и ширине экрана определите количество строк, которые выведет команда «ls -C».

Формат входных данных

В первой строке входного файла list.in даны два целых числа n и w — количество файлов в списке и ширина экрана ($1 \leq n \leq 10^5$, $1 \leq w \leq 10^9$).

Во второй строке даны n чисел. i -ое из них равно длине имени i -го файла из списка, который уже упорядочен в алфавитном порядке.

Гарантируется, что длина имени любого файла — целое число от 1 до w .

Формат выходных данных

В выходной файл list.out выведите одно число — количество строк, которые выведет команда «ls -C».

Примеры

list.in	list.out
11 20 1 3 7 4 1 2 1 1 1 1 4	3

Замечание

Решения, решающие задачу для $n \leq 10^4$ оцениваются в 50 баллов из 100.