

Задача 1. Задачи по информатике

(идея задачи с codeforces.ru)

Входной файл	input.txt
Выходной файл	output.txt
Ограничение времени (сек/тест)	2
Максимальный балл	25+10 ^(*)

Условие задачи

В рамках подготовки к олимпиаде по информатике и программированию учитель составил для Вовочки список задач, которые Вовочка должен был решить во время каникул. Однако нашему герою решить все задачи не удалось. К задачам прилагались правильные ответы, поэтому Вовочка точно знает, какие задачи он смог решить, а какие нет.

С первого дня очередной четверти Вовочке предстоит отчитаться о проделанной работе перед учителем. Вовочка точно знает, что учитель будет строго последовательно проверять решения (по составленному списку задач). Если решение верное, то проверка такого решения не отнимает у учителя много времени (т.е. в один день учитель может проверить произвольное количество правильно решенных задач). Если же решение неверное или решения нет, в этом случае учитель вместе с Вовочкой разбирают задачу, на что тратится значительное время. Известно, что в один день учитель может разобрать с Вовочкой не более 2-х нерешенных задач.

Т.к. разбор задач проходит после уроков, Вовочка хочет потратить на это как можно меньше учебных дней. Помогите Вовочке распределить задачи для проверки учителем по дням таким образом, что количество затраченных дней было минимально.

Входные данные

В первой строке входного файла записано число N ($1 \leq N \leq 1000$) – число задач в списке. На следующей строке через пробел записано N чисел 0 или 1. Каждое i -ое число соответствует результату решения i -ой задачи: 1 – задача решена, 0 – задача не решена или решена неверно.

Выходные данные

В первую строку выходного файла требуется вывести целое число K – минимальное количество учебных дней, которые вынужден будет потратить Вовочка. Во второй строке выходного файла должны быть записаны K целых чисел – число задач, которые будет показывать Вовочка учителю в очередной день. Т.е. i -ое число соответствует количеству задач, по которым Вовочка сможет отчитаться перед учителем в i -ый день.

Если существует несколько одинаковых по эффективности (количеству потраченных дней) решений, то выведите любое из них.

Пример входного файла (input.txt)	Пример выходного файла (output.txt)
12 1 1 1 0 0 0 1 0 0 0 1 1	3 5 3 4
7 1 1 1 0 0 1 0	2 4 3

(*) – дополнительные 10 баллов начисляются за абсолютно правильное решение задачи (если решение проходит все тесты)

Задача 2. Скобочные последовательности

(идея задачи с codeforces.ru)

Входной файл	input.txt
Выходной файл	output.txt
Ограничение времени (сек/тест)	2
Максимальный балл	25+10

Условие задачи

Одной из задач, которую Вовочка не смог решить, была следующая задача.

Назовем строку, составленную только из символов '(' и ')', скобочной последовательностью. Будем считать, что такая последовательность правильная, если из нее путем вставки символов математических операций и цифр можно получить корректное математическое выражение. Например, последовательности "()()", "()" и "((()()())" – правильные, в то время как ")(", "())" и "((((((((" – нет.

Помогите Вовочке ответить на вопрос, какое минимальное количество скобок нужно удалить из произвольной скобочной последовательности, чтобы сделать ее правильной.

Входные данные

В единственной строке входного файла записана непустая скобочная последовательность. Длина скобочной последовательности не превосходит 10^6 символов.

Выходные данные

В выходной файл требуется вывести единственное число – минимальное кол-во скобок, которые нужно удалить из исходной последовательности, чтобы сделать ее правильной. Если последовательность уже является правильной, то ничего удалять не нужно, соответственно вывести надо число 0. Если из исходной последовательности удалением скобок невозможно сделать правильную скобочную последовательность, то в файл надо записать число -1.

Пример входного файла (input.txt)	Пример выходного файла (output.txt)
(() ()) ()	2
() (()) ()	0
))) (-1

Задача 3. Восстановление IPv6-адреса

(задача с codeforces.ru)

Входной файл	input.txt
Выходной файл	output.txt
Ограничение времени (сек/тест)	2
Максимальный балл	25+10

Условие задачи

IPv6-адрес – это некоторое 128-битное число. Для удобства это число записывают блоками по 16 бит в шестнадцатеричной системе счисления, разделяя блоки двоеточиями — всего 8 блоков, в каждом из которых по 4 шестнадцатеричные цифры. Вот пример корректной записи адреса IPv6: “0235:a034:20cd:cdef:0235:8765:55c0:fedc”. Будем называть такой формат записи IPv6-адреса полным.

Помимо полного формата записи IPv6-адреса, существует сокращенный формат записи. Запись IPv6-адреса может быть сокращена путем удаления одного или нескольких ведущих нулей в начале каждого блока. Однако каждый из блоков в итоге должен содержать хотя бы одну цифру. Например, ведущие нули можно сократить следующим образом: “a33e:00d4:0000:0248:0007:ffea:1000:0000” → “a33e:d4:0:0248:07:ffea:1000:00”. Существуют и другие способы сократить нули в этом IPv6-адресе.

Некоторые IPv6-адреса содержат длинные последовательности нулей. Сплошные последовательности из нулевых 16-битных блоков могут быть сокращены до “::”. Последовательность должна состоять из одного или нескольких последовательных блоков, все 16 бит в которых равны 0.

Примеры сокращения последовательностей нулевых блоков можно видеть ниже:

```
a65f:00b3:0000:0248:0007:0000:0000:0000 → a65f:00b3:0000:0248:0007::
a65f:0000:0000:0248:0007:0000:5678:0ff0 → a65f::0248:0007:0000:5678:0ff0
a65f:0000:0000:0000:0007:0000:5678:0ff0 → a65f:0000::0000:0007:0000:5678:0ff0
a65f:00b3:0000:0248:0007:0000:0000:0000 → a65f:00b3:0000:0248:0007::0000
0000:0000:0000:0000:0000:0000:0000:0000 → ::
```

Сокращение нулевых блоков в адресе разрешается использовать не более одного раза. Это означает, что в сокращенной записи последовательность символов “::” может встретиться не более одного раза. В противном случае иногда будет невозможно определить число нулевых блоков, представленных каждым двойным двоеточием.

Требуется написать программу, которая будет восстанавливать полную форму записи из сокращенной.

Входные данные

В единственной строке входного файла записан IPv6-адрес в сокращенной или полной форме.

Выходные данные

В единственную строку выходного файла требуется записать полную форму заданного во входной файле IPv6-адреса.

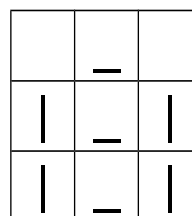
Пример входного файла (input.txt)	Пример выходного файла (output.txt)
a65f:00b3:0000:0248:0007::	a65f:00b3:0000:0248:0007:0000:0000:0000
::	0000:0000:0000:0000:0000:0000:0000:0000
0de::8a:e4:777:0	00de:0000:0000:0000:008a:00e4:0777:0000

Задача 4. Сложение чисел ++

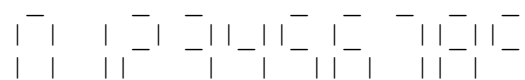
Входной файл	input.txt
Выходной файл	output.txt
Ограничение времени (сек/тест)	2
Максимальный балл	25+10

Условие задачи

Как-то раз в первом класса на уроке математики Вовочка, вместо того, чтобы решать примеры, рисовал в тетради в клеточку цифры. Все нарисованные цифры выглядели как на дисплее калькулятора (см. рисунок), т.е. одна цифра занимала ровно 9 клеточек. Цифры располагались в несколько рядов (строк).



– представление
цифры 8



Учитель заметил, что Вовочка занимается ерундой, и в качестве наказания (иначе обещал поставить двойку) предложил Вовочке сосчитать сумму чисел, которые образовывали цифры в каждом ряду. Помогите Вовочке справиться с этим заданием.

Входные данные

Входной файл состоит из нескольких строк одинаковой длины (в символах) – для записи расположения чисел, получившегося у Вовочки. При этом вертикальные отрезки нарисованных цифр записываются символом вертикальной черты – '|', а горизонтальные – символом подчеркивания; пустое место обозначается пробелом. Любая цифра состоит из 9 символов, на рисунке показана запись цифры 8 в таком формате. Остальные цифры записываются аналогично – см. рисунок. Максимальное количество цифр в каждом числе – 9. Между цифрами могут быть произвольные расстояния, которые описываются одинаковым количеством пробелов во всех 3-х строках конкретного ряда цифр. При этом длина строк вместе с пробелами, включая конечные, – не более 100 символов каждая. Между рядами цифр также может быть произвольное число строк, состоящих из одних пробелов. Общее количество строк, включая состоящие из одних пробелов, также не более 100.

Выходные данные

В выходной файл требуется вывести единственное число – сумму чисел, составленных из цифр каждого ряда. Известно, что полученная сумма не более 10^9 .

Пример входного файла (input.txt)	Пример выходного файла (output.txt)
<pre> _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ </pre>	1511
<pre> _ _ _ _ </pre>	2

Задача 5. Телепередачи

Входной файл	input.txt
Выходной файл	output.txt
Ограничение времени (сек/тест)	2
Максимальный балл	25+10

Условие задачи

Вовочка, вместо того чтобы учить уроки, решил весь воскресный день смотреть телевизор. Для этого он в субботу вечером просмотрел программу телепередач и составил пронумерованный список из N ($1 \leq N \leq 1000$) телепередач, которые он хотел бы увидеть. Так как в одно и то же время могут идти несколько телепередач (если они транслируются на нескольких каналах), то Вовочка понял, что все их просмотреть он не сможет. Помогите ему составить план просмотра так, чтобы он увидел как можно больше телепередач из составленного списка, при условии, что любая телепередача будет им просматриваться от начала до конца (это также означает, что в одно и то же время нельзя Вовочке смотреть две телепередачи, но вполне допустимо, чтобы в какое-то время он ничего не смотрел).

Входные данные

Во входном файле в первой строке содержится число N – количество телепередач в списке. Каждая следующая из N строк содержит информацию об одной телепередаче. Т.е. в $i+1$ строке файла описывается i -ая телепередача, для которой задается время ее начала и время окончания, разделенные ровно одним пробелом. Время записывается в формате **чч:мм** и изменяется в диапазоне от 08:00 до 22:00.

Выходные данные

В первую строку выходного файла необходимо вывести число телепередач в составленном плане (напоминаем, что оно должно быть максимально возможным). Во второй строчке должны быть записаны через пробел номера телепередач в составленном плане, которые Вовочка будет последовательно просматривать.

Пример входного файла (input.txt)	Пример выходного файла (output.txt)
5 11:20 13:00 08:00 11:30 09:00 11:20 14:00 16:30 10:00 11:00	3 3 1 4

Задача 6. Выравнивание по ширине

Входной файл	input.txt
Выходной файл	output.txt
Ограничение времени (сек/тест)	2
Максимальный балл	25+10

Условие задачи

Как известно, в прошлом у видеокарт было мало памяти и поэтому большинство программ работало в текстовом режиме 80 x 24 (24 строки текста по 80 символов). Прошло время, видеокарты стали на порядки производительнее, однако консольные приложения никуда не исчезли и по-прежнему по умолчанию работают в текстовом режиме 80 x 24.

У Вовочки первый компьютер появился очень-очень давно, еще во времена операционной системы MS-DOS (Windows тогда еще не было), поэтому консольные приложения для него милы и понятны. Любимая программа Вовочки сейчас – Far Manager и конечно же Вовочка предпочитает текстовый режим 80 x 24. В данной программе есть встроенный просмотрщик / редактор текстовых файлов (также работает в текстовой режиме).

Недавно на просторах Интернета Вовочка нашел серию рассказов своего любимого писателя. Оказалось, что тексты рассказов отформатированы непонятно как и читать их в Far'e неудобно. Самым удобным режимом для чтения Вовочка считает режим с выравниванием текста по ширине. Помогите Вовочке отформатировать текст, как ему нравится.

Напомним, что в текстовом режиме, в котором работает Вовочка, в одну строку помещается 80 символов. При выравнивании по ширине выполняются следующие условия:

- Каждая строка начинается со слова или знака препинания, если абзац начинается со знака препинания (1-ый символ – буква или знак препинания). Исключение составляют только строки, с которых начинается очередной абзац. В начале каждого абзаца должны стоять 4-е пробела.
- Каждая строка должна заканчиваться словом или знаком препинания (80-ый символ – буква или знак препинания). Исключение составляют только строки, которыми заканчивается абзац, а также строки, состоящие из одного слова. Такие строки выравниваются влево, после последнего значащего символа пробелов нет.
- Знаками препинания считаются 5 символов – (; , . ! ?) . Все остальные символы, за исключением пробела и символов новой строки, считаются буквами, из которых могут состоять слова. Гарантируется, что в тексте отсутствуют символы табуляции и другие управляющие символы.
- Строки друг от друга отделяют 2 символа с кодами 13 и 10 – «перевод строки» и «возврат каретки». Последняя строка также должна заканчиваться данными символами.
- После каждого слова должен обязательно стоять один или более пробелов. Если после слова стоит знак (знаки) препинания, то перед ними пробел не ставится, а после них ставится (т.е. знак препинания «прилипает» к слову, после которого он стоит). Если непосредственно после знаков препинания (без пробелов между) стоит кавычка, одинарная или двойная (' "), или одна из 3-х закрывающихся скобок () }]) то она также не должна отделяться от слова со знаками препинания. Слово со знаками препинания, кавычками и закрывающимися скобками в конце можно считать неделимым словом.
- Пробелы не ставятся после точки, если она непосредственно окружена цифрами.

- Об исходном форматировании текста известно следующее:

- Для понимания остальных особенностей форматирования текста по ширине – см. примеры входных и выходных данных.

Текст записан во входном файле в виде множества строк. Обращаем ваше внимание, что строка (абзац) может оказаться длиной более 255 символов.

В выходной файл необходимо вывести выровненный по ширине (согласно описанным выше правилам) исходный текст.

Пример входного файла (input.txt)

```
Формулировать условия задач по программированию сложно (я бы даже сказал "
очень-очень").Здесь важно постараться_недопустить_неоднозначности_толкования_у
словия. Что_не_всегда_удается_сделать. И это только одна из сложностей.¶
Дальше идет просто текст,для условия; ; Задачи-задачи:¶
?? вы еще здесь?странно! ... (Очень ) . ¶
wefh 892/, 234798798,ewjfk,,324.890/.;/324789234 ijkdwjn7892 m234 78d u234 bjhbj
ehwb hbj iuniuen uniuwe iuej ; i i i i i i i i i i i i i i i iiiiiiiiiiiiiiiiii
iiiiiiiiiiiiiiiiiiiiiiiiiiiiii.¶
¶
end.¶
```

Пример выходного файла (output.txt)

```
Формулировать условия задач по программированию сложно (я бы даже сказал "очень-очень"). Здесь важно постараться не допустить неоднозначности толкования условия. Что не всегда удается сделать. И это только одна из сложностей.
    Далее идет просто текст, для условия;; Задачи-задачи:
        ?? вы еще здесь? странно!... (Очень ).
        wefh 892/, 234798798, ewjfk,, 324.890/. /; 324789234 ijkdwjn7892 m234 78d
u234 bjhbje hwb hbj iuniuen uniuwe iuej; i i i i i i i i i i i i i i i i
iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii.
end.
```

Символов ¶ в текстах нет, данный символ используется, чтобы показать концы строк.