

2015

Национальный исследовательский университет
Мордовский государственный университет им. Н. П. Огарева

[ОТКРЫТАЯ ОЛИМПИАДА ПО ПРОГРАММИРОВАНИЮ]

А. Скоро Новый год

Входные данные: файл **input.txt**

Выходные данные: файл **output.txt**

Ограничение по времени: 3с

Ограничение по памяти: 256Мб

У Деда Мороза очень ответственная работа. Нужно подготовить столько подарков. И чтобы все детишки были довольны. Поэтому он начинает подготовку заранее. В этом году ему нужно поздравить $1 \leq n \leq 100$ детей. У него в распоряжении $1 \leq m \leq 10^9$ видов конфет. Каждый год его помощники собирают статистические данные. Поэтому Деду Морозу заранее известно, какие виды конфет предпочитает каждый ребенок. Но составлять индивидуальные подарки слишком трудоемко, поэтому Дед Мороз хочет составить такой набор конфет, чтобы каждый ребенок получил только те, которые ему нравятся. При этом он хочет использовать в подарке максимально возможное количество видов конфет. Помогите Деду Морозу составить подходящий набор.

Входные данные: первая строка входного файла содержит два целых числа n и m . Далее следует n строк. i -я строка содержит целое неотрицательное число k —количество статистических данных по i -му ребенку. Далее через пробел перечисляются номера видов конфет, которые предпочитает i -й ребенок. Нумерация видов конфет начинается с 1. Гарантируется, что суммарное количество уникальных чисел во входном файле не превышает 10^6+2 , а общее количество чисел во входном файле не превышает 10^7+2 .

Выходные данные: в первую строку выходного файла поместите число k - количество уникальных видов конфет; во вторую строку поместите k чисел - номера видов конфет, которые войдут в подарочный набор, отсортированные по возрастанию. Если невозможно составить подарок, удовлетворяющий условиям, выведите единственное число -1 .

Пример #1

Файл input.txt

```
3 2
2 1 2
2 2 2
2 2 1
```

Файл output.txt

```
1
2
```

Пример #2

Файл input.txt

```
3 1000
3 1 1 1
0
1 1000
```

Файл output.txt

```
-1
```

В. Заповедник

Входные данные: файл **input.txt**

Выходные данные: файл **output.txt**

Ограничение по времени: 1с

Ограничение по памяти: 256Мб

Большой Берляндский Бесконечный Лес был посажен много веков назад первыми поселенцами Берляндской земли. И поскольку они были приверженцами точных наук, они посадили деревья ровными рядами. Расстояние между соседними деревьями в ряду и между соседними рядами составляет l метров.

Правительство Берляндии старается заботиться о флоре и фауне. Особенно о фауне. Поэтому на территории Большого Берляндского Бесконечного Леса был построен заповедник для редчайшего вида белок – берлок. Для комфортного существования одной берлке требуется k свободных деревьев. Заповедник был построен с расчетом размещения в нем ровно m пар берлок. Известно, что граница заповедника проходит через n растущих в лесу деревьев, которые используются как опорные столбы, таким образом, что заповедник представляет собой произвольный многоугольник.

Ваша задача - определить площадь построенного заповедника.

Входные данные: первая строка входного файла содержит целые числа $1 \leq l \leq 100$, $1 \leq k \leq 10^6$, $3 \leq n \leq 10^6$, $1 \leq m \leq 10^6$. Корректность входных данных и существование заповедника, удовлетворяющего заданным условиям, гарантируются.

Выходные данные: в выходной файл поместить единственное число – площадь заповедника в квадратных метрах с точностью до 6 знаков после запятой.

Пример #1

Файлinput.txt

10 3 4 1

Файлoutput.txt

300.000000

С. Парковка

Входные данные: файл **input.txt**

Выходные данные: файл **output.txt**

Ограничение по времени: 4с

Ограничение по памяти: 256Мб

Компания МегаХард любит проверять смекалку своих сотрудников прямо на рабочем месте в процессе обыденных, казалось бы, занятий. Вот и теперь специалисты отдела обеспечения качества кадров придумали для своих сотрудников «развлечение» – необычную парковку. Новая двухуровневая парковка была построена по следующему принципу. Все машины заезжают на верхний уровень, а затем с помощью лифта автоматически спускаются на нижний уровень и ставятся на некоторое парковочное место, постоянное для каждого сотрудника. Нижний уровень парковки состоит из $n \times m$ ячеек и спроектирован в виде лабиринта. Перемещаться по парковке можно только от ячейки к соседним ячейкам в любом направлении (по диагонали перемещаться нельзя). Кабинет каждого сотрудника оборудован специальным транспортером, который настраивается на определенное парковочное место. Если сотрудник хочет забрать свою машину, транспортер доставляет его прямо к ней на нижний уровень парковки. Ежемесячно компания устраивает квест. Каждой ячейке парковки присваивается стоимость. При переходе от одной ячейки к другой месячная премия сотрудника уменьшается на величину абсолютной разницы стоимостей этих ячеек. Итак, вам известно ваше парковочное место и описание лабиринта. Найдите по этим данным выход с парковки, такой, что ваша месячная премия будет максимально возможной.

Входные данные: первая строка входного файла содержит целые числа $1 \leq n \leq 1000$ и $1 \leq m \leq 1000$ – размеры парковки. Вторая строка содержит два целых числа $1 \leq i \leq n$ и $1 \leq j \leq m$ – номера строки и столбца ячейки, соответствующей парковочному месту.

Далее следует n строк по m целых неотрицательных чисел в каждой – стоимости ячеек парковки. Величина стоимости одной ячейки $0 \leq s_{ij} \leq 32766$.

Далее следует $n+1$ строк – описание ячеек лабиринта парковки. Каждая ячейка может иметь верхнюю, нижнюю, левую или правую границу (любое их сочетание или ни одну из них). Если ячейка имеет левую или правую границу и не имеет нижней границы, то ее описание представляет собой символ пробела, а слева или справа от нее ставится символ | (вертикальная черта). Разумеется, левая граница текущей ячейки является одновременно и правой границей предыдущей ячейки той же строки. То же справедливо и для правой границы текущей ячейки – она одновременно является левой границей следующей ячейки той же строки. Если ячейка имеет нижнюю границу, то ее описание – это символ _ (нижнее подчеркивание). Этот же символ является верхней границей ячейки следующей строки, расположенной под текущей ячейкой. То есть, чтобы текущая ячейка имела верхнюю границу, ячейка над ней должна иметь нижнюю границу. Кроме того, для удобочитаемости в описание лабиринта были добавлены дополнительные символы пробелов (см. пример входных данных).

Выход из лабиринта обозначен как отсутствие символа _ или | вместо одной из границ лабиринта.

ВНИМАНИЕ! В примере входных данных символы пробелов были заменены на символы s. Но тесты к задаче содержат именно символы пробелов!

Выходные данные: в первую строку выходного файла поместить целое число s – длину искомого пути; во вторую строку поместить величину, на которую сократится ваша премия по итогам прохода лабиринта; затем вывести s строк: в каждой строке вывести

пару целых чисел через пробел – номера строки и столбца ячейки, принадлежащей найденному пути. Путь должен начинаться с координат парковочного места и заканчиваться координатами ячейки с выходом. Если решений несколько, то выведите любое из них. Гарантируется, что выход из лабиринта и путь к нему существуют.

Пример #1**Файл input.txt**

```
3 3
1 1
1 2 3
4 5 6
7 8 9
s_s_s_s
|s|_ss|
|s|s|s|
|_s_ss|
```

Файл output.txt

```
5
8
1 1
2 1
3 1
3 2
3 3
```

Пример #2**Файл input.txt**

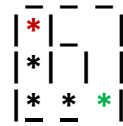
```
3 3
1 1
1 2 3
4 5 6
7 8 9
s_s_s_s
|sssss|
|sssss|
|_s_ss|
```

Файл output.txt

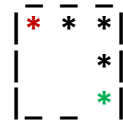
```
5
8
1 1
1 2
1 3
2 3
3 3
```

Примечание.

Для примера 1 имеем следующее решение (красной звездочкой обозначено парковочное место, зеленой – ячейка с выходом, черными – ячейки пути):



Одно из возможных решений примера 2:



D. Чемпионат по нему

Входные данные: файл **input.txt**

Выходные данные: файл **output.txt**

Ограничение по времени: 1с

Ограничение по памяти: 256Мб

Васе дали очень важное поручение. Его попросили организовать Чемпионат школы по нему. Им – это игра, суть которой заключается в следующем. Имеется некоторое количество кучек камней. Игроки по очереди берут ненулевое количество камней из любой кучки. Проигрывает тот, кто не может сделать ход.

У Васи есть камни n различных цветов. Для Чемпионата он хочет составить одинаковые кучки размером в m камней. В каждой кучке не должно быть двух камней одного цвета.

Сколькими способами Вася может составить кучки, удовлетворяющие заданным условиям?

Входные данные: входной файл содержит одну строку, в которой последовательно через пробел записаны числа $1 \leq n \leq 20, 1 \leq m \leq n$.

Выходные данные: в первую строку выходного файла поместить одно целое число: количество способов составления нужного набора камней.

Пример #1

Файл input.txt

3 2

Файл output.txt

3

Е. Военно-морской парад

Входные данные: файл **input.txt**

Выходные данные: файл **output.txt**

Ограничение по времени: 1с

Ограничение по памяти: 256Мб

В Берляндии к юбилею знаменитого адмирала Командиркина решили устроить военно-морской парад. Но знаменательное событие уже совсем скоро... Поэтому осталось совсем мало времени для репетиций, и командование решило доверить управление кораблями суперкомпьютеру.

Для того чтобы корабли шли красиво ровным строем, необходимо написать программу, вычисляющую координаты четвертого корабля, по координатам трех других, предполагая, что они расположены в вершинах прямоугольника.

Входные данные: входной файл содержит три строки, в каждой из которых последовательно через пробел записаны координаты расположения кораблей x_i и y_i ($0 \leq x_i, y_i \leq 10^5$) в двоичной системе счисления. Координаты всех кораблей – целые неотрицательные числа.

Выходные данные: в единственную строку выходного файла поместить два числа, записанных в двоичной системе счисления, - целые части координат четвертого корабля или ERR, если корабли не могут образовать прямоугольник.

Пример #1

Файл input.txt

```
0 100
100 0
0 0
```

Файл output.txt

```
100 100
```

Пример #2

Файл input.txt

```
1 1
10 1
11 101
```

Файл output.txt

```
ERR
```


F. МУУУУ!

Входные данные: файл **input.txt**

Выходные данные: файл **output.txt**

Ограничение по времени: 3с

Ограничение по памяти: 256Мб

С этого года в Берляндии принята новая программа по ликвидации компьютерной безграмотности среди широких кругов населения. И если обучение мартышек программированию большой проблемы не представляет, то с буренками не все так просто! Они очень капризны и абсолютно отказываются понимать программирование на C++. Берляндская академия наук предложила разработать новый язык программирования, специально предназначенного для использования буренками. Язык этот очень прост, и состоит только из слов, употребляемых в языке буренок. Все инструкции языка представляют собой вариацию слова «тоо», единственного слова, которое они понимают. Все остальные слова и символы не являются инструкциями и полностью игнорируются. Пробелы, табуляции, переносы строк могут как использоваться, так и не использоваться. То же и с другими символами. Все инструкции являются регистрозависимыми.

В языке используются следующие команды:

- 0 тоо Команда связана с командой MOO. При нормальном исполнении, просматривает программный код в обратном порядке в поисках ближайшей команды MOO и продолжает выполнение программы с нее. При поиске пропускается команда, непосредственно предшествующая тоо (см. пример MOO). Эта команда всегда должна встречаться в паре с MOO.
- 1 mOo Перемещает текущую позицию на предыдущую ячейку.
- 2 moO Перемещает текущую позицию на следующую ячейку.
- 3 mOO Считывает значение из текущей ячейки и исполняет его как команду с соответствующим опкодом (например, если в текущей ячейке содержится значение 2, исполняется команда moO). Некорректные опкоды и код 3 приводят к завершению программы.
- 4 Моо Если в текущей ячейке содержится 0, то из STDIN считывается один ASCII символ и сохраняется в текущей ячейке. Если в текущей ячейке содержится не 0, то выполняется печать символа по его ASCII коду, содержащемуся в этой ячейке, в STDOUT.
- 5 MOo Уменьшает значение в текущей ячейке на 1.
- 6 MoO Увеличивает значение в текущей ячейке на 1
- 7 MOO Если значение в текущей ячейке равно 0, пропускает следующую команду и продолжает исполнение программы после ближайшей команды тоо. Если значение в текущей ячейке отлично от 0, то исполняется следующая команда. Пропуск команды важен! Например, в следующем коде будет использована вторая тоо: **OOOMOotoo**
- 8 OOO Устанавливает значение в текущей ячейке равным 0.
- 9 MMM Если в регистр не загружено значение, то копирует в него значение из текущей

ячейки. Если в регистре есть значение, то копирует это значение в текущую ячейку и очищает регистр.

- 10 OOM Печатает целочисленное значение текущей ячейки в STDOUT на отдельной строке
- 11 oom Считывает целочисленное значение из STDIN и помещает его в текущую ячейку.

Для исполнения программ используется компьютер, состоящий из $5 \cdot 10^4$ целочисленных ячеек памяти, одного целочисленного регистра (который может быть либо пуст, либо содержать определенное значение) и регистра счетчика команд.

Помогите Берляндской академии наук разработать интерпретатор языка МУ!

Входные данные: исходный файл содержит программу на языке МУ! и набор входных данных, считываемых программой из stdin. Первые строки входного файла содержат текст программы. Максимальная длина программы – 10^4 инструкций. Затем на отдельной строке располагается строка-разделитель <stdin>. После чего располагаются строки входных данных, которые могут быть символом латинского алфавита, цифрой или целым числом, по одному элементу на строку.

Выходные данные: в выходной файл поместите вывод программы при заданном наборе входных данных. Если программа ничего не выводит, в выходной файл поместите строку EMPTY. Если программа завершилась досрочно или ее не удастся скомпилировать, в выходной файл поместите строку BOOM.

Пример #1

Файл input.txt

```
oom
MoO
OOM
```

```
<stdin>
3
```

Файл output.txt

```
4
```

Пример #2

Файл input.txt

```
om
```

```
<stdin>
```

Файл output.txt

```
EMPTY
```

G. Система поиска

Входные данные: файл **input.txt**

Выходные данные: файл **output.txt**

Ограничение по времени: 3с

Ограничение по памяти: 256Мб

В Берляндии планируется открыть новую библиотеку, все книги в которой будут храниться в электронном суперкомпьютере. Естественно, все будет оборудовано по последнему слову техники и науки, и для хранения и поиска книг планируется использовать самые совершенные алгоритмы! Чтобы пользователи могли легко и быстро находить нужные им книги, Берляндская академия наук планирует разработать и внедрить совершенно новую систему поиска, которая будет находить книги и слова в книгах, даже если пользователи ввели запросы в систему с опечатками.

Помогите академии наук разработать такой алгоритм.

Входные данные: первая строка входного файла содержит поисковый запрос – слово, которое необходимо найти, длина которого не превышает 100 символов, и число k – максимальное количество ошибок ($0 \leq k \leq 5$). Ошибкой считается вставка, удаление или замена символа в некоторой позиции. Вторая строка содержит единственное число N ($1 \leq N \leq 5 \cdot 10^3$) – количество слов в словаре. Последующие N строк входного файла содержат слова – последовательности букв латинского алфавита в верхнем и нижнем регистре. Одна строка содержит только одно слово.

Выходные данные: в выходной файл поместите список всех слов, удовлетворяющих критерию поиска без учета регистра, в алфавитном порядке, приведенных к нижнему регистру. Если по критерию поиска совпадений нет, то в выходной файл выведите NOTFOUND.

Пример #1

Файл input.txt

```
hello 1
3
Hello
hell
bell
```

Файл output.txt

```
hell
hello
```

Пример #2

Файл input.txt

```
hello 1
1
ding
```

Файл output.txt

```
NOTFOUND
```

Н. Сломанный модуль

Входные данные: файл **input.txt**

Выходные данные: файл **output.txt**

Ограничение по времени: 9с

Ограничение по памяти: 256Мб

У Алекса случилась беда... Во время выполнения сверхсекретного задания он повредил свой дешифратор и у него сломался один из модулей. Теперь Алекс не может получать секретную информацию от Юстаса. Для успешного завершения сверхсекретной миссии ему необходимо срочно починить модуль. Механические повреждения удалось восстановить, и теперь требуется восстановить алгоритм, реализуемый этим модулем. Но Алекс совершенно не смыслит в программировании, и ждет помощи.

Функция сломанного модуля заключается в нахождении количества решений уравнения вида:

$$a \bmod x = b,$$

где a и b – параметры, подаваемые на вход модуля, \bmod – операция нахождения остатка от деления. Модуль поддерживает только арифметику с целыми неотрицательными числами.

Входные данные: входной файл содержит одну строку, в которой последовательно через пробел записаны два числа a и b ($0 \leq b \leq a \leq 10^9$).

Выходные данные: в единственную строку выходного файла поместить одно целое число, равное количеству решений или INF, если решений бесконечно много.

Пример #1

Файл input.txt

2 2

Файл output.txt

INF

Пример #2

Файл input.txt

4214412 56

Файл output.txt

3