

## Отборочный этап. Второй тур (приведен один из вариантов заданий)

### 1. Электронные таблицы. Адресация ячеек и вычисления (2 балла)

[Что и по какому модулю]

В ячейки таблицы B1 и A2 записали некоторые формулы:

	A	B
1		=A\$1+1
2	=POW(MOD(A1; XX); 2)	
3		

В формуле в ячейке A2 XX обозначает некоторое (необязательно двузначное) число. После чего формулу из ячейки B1 скопировали во все ячейки диапазона B1:M1, а формулу из ячейки A2 во все ячейки диапазона A2:M20. В ячейку A1 написали некоторое число, не кратное числу XX. В результате было обнаружено, что в диапазоне E2:E20 число, отличное от 1, содержится только в ячейке E2. Определите наименьшее возможное значение для XX и для него наименьшее возможное число в ячейке A1, если в ячейке E2 содержится число 900. В ответ запишите через пробел 2 числа – сначала XX, затем число из ячейки A1.

*Примечание: таблица соответствия имён используемых функций.*

	Excel	Google Sheets	LibreOffice
Возведение в степень	СТЕПЕНЬ	POW	POWER
Остаток от деления	ОСТАТ	MOD	MOD

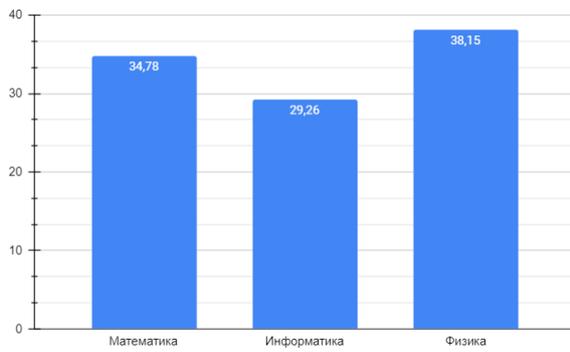
Ответ: 31 26

### 2. Электронные таблицы. Графики и диаграммы (1 балл)

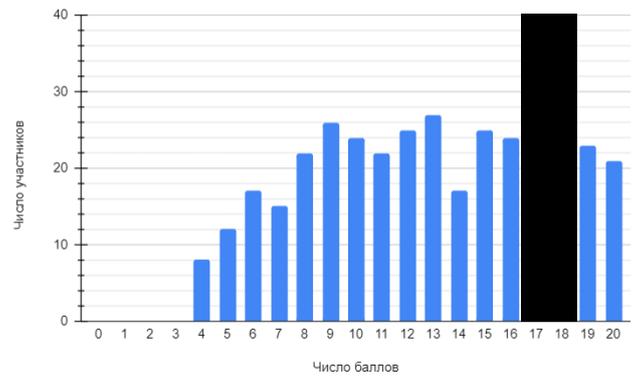
[Данные удалены]

Были проведены олимпиады школьников по трём предметам – информатике, математике и физике. В каждой олимпиаде можно было получить от 0 до 20 баллов, чтобы стать призёром нужно было набрать хотя бы 16. По итогам олимпиад организаторы составили две диаграммы: процент призёров среди участников олимпиады и распределение участников олимпиады по математике по числу баллов. Однако часть второй диаграммы была удалена.

Процент призов среди участников



Распределение участников олимпиады по математике по числу баллов



Определите число участников олимпиады по математике. В ответ запишите одно целое число.

Примечание. Цифры на графике округлены до второго знака после запятой.

Ответ: 368

### 3. Сортировка и фильтрация данных (3 балла)

#### [Шаги к успеху]

Примечание:

*Сортировка пузырьком:* выполняется некоторое количество проходов по массиву — начиная от начала массива, перебираются пары соседних элементов массива. Если 1-й элемент пары больше (в случае сортировки по возрастанию; в случае сортировки по убыванию — меньше) 2-го, элементы переставляются (выполняется обмен). Пары элементов массива перебираются (проходы по массиву повторяются) либо  $n-1$  раз (где  $n$  — число элементов массива), либо до тех пор, пока на очередном проходе не обнаружится, что более не требуется выполнять перестановки (обмены) (массив отсортирован). При каждом проходе алгоритма по внутреннему циклу очередной наибольший (в случае сортировки по возрастанию; в случае сортировки по убыванию — наименьший) элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим элементом» (в случае сортировки по возрастанию; в случае сортировки по убыванию — наименьшим), а наименьший (в случае сортировки по возрастанию; в случае сортировки по убыванию — наибольший) элемент перемещается на одну позицию к началу массива.

*Сортировка слиянием работает по следующему принципу:*

1. Если в рассматриваемом массиве один элемент, то он уже отсортирован — алгоритм завершает работу.
2. Иначе массив разбивается на две части, которые сортируются рекурсивно (т.е. на каждой из частей выполняется описанный алгоритм сортировки слиянием).
3. После сортировки двух частей массива к ним применяется процедура слияния, которая по двум отсортированным частям получает исходный отсортированный массив. В рамках процедуры слияния сравниваются элементы сливаемых массивов (начиная с начала) и меньший (в случае сортировки по возрастанию) из них записываем в финальный. И затем, в массиве у которого оказался меньший (в случае сортировки по возрастанию) элемент, переходим к следующему элементу и сравниваем теперь его. В конце, если один из массивов закончился, мы просто дописываем в финальный другой массив. После мы наш финальный массив записываем вместо двух исходных и получаем отсортированный участок. С точки зрения исходного массива данная операция атомарна, то есть не имеет промежуточных этапов.

*Пример работы сортировки слиянием:*

1. [4 3 1 6 5] — исходный массив
2. [[4 3 1][6 5]] — разделение массивов
3. [[[4 3][1]][6 5]] — самый первый неотсортированный подмассив разделяется
4. [[[[4][3]][1]][6 5]] — самый первый неотсортированный подмассив разделяется
5. [[3 4][1][6 5]] — производится слияние первых двух одноэлементных массивов. Порядок элементов изменился — шаг 1
6. [[1 3 4][6 5]] — производится слияние первых двух отсортированных массивов (результата слияния и одноэлементного). Порядок изменился — шаг 2.
7. [[1 3 4][6][5]] — неотсортированный подмассив разделяется
8. [[1 3 4][5 6]] — производится слияние одноэлементных подмассивов. Порядок изменился — шаг 3
9. [1 3 4 5 6] — производится слияние подмассивов. Порядок элементов не изменился — таким образом, было 3 шага.

Сегодня на спецкурсе по олимпиадному программированию Илья узнал про два алгоритма сортировки: сортировку пузырьком и сортировку слиянием. Илья стал изучать, как эти сортировки работают на частично отсортированных массивах. Особенно его интересует, какая из сортировок занимает меньше шагов. Шагом Илья называет изменение порядка элементов массива в ходе работы алгоритма. Например, в сортировке пузырьком на массиве [4 2 1 3] по мнению Ильи 4 шага:

1. [4 2 1 3] -> [2 4 1 3]
2. [2 4 1 3] -> [2 1 4 3]

3. [2 1 4 3] -> [2 1 3 4]

4. [2 1 3 4] -> [1 2 3 4]

А в сортировке слиянием на массиве [4 2 3 1] 3 шага:

1. [4 2 3 1] -> [2 4 3 1]

2. [2 4 3 1] -> [2 4 1 3]

3. [2 4 1 3] -> [1 2 3 4]

Помогите Илье определить, какая сортировка сделает меньше шагов на массиве [1 2 3 ... 127 128 256 255 ... 130 129], состоящем из двух частей – в первой половине натуральные числа от 1 до 128 отсортированы по возрастанию, во второй – числа от 129 до 256 отсортированы по убыванию (таким образом, всего в массиве 256 элементов). В ответе укажите два числа – количество шагов при сортировке массива пузырьком и число шагов при сортировке слиянием.

**Ответ: 8128 127**

#### 4. Поиск и фильтрация данных (2 балла)

##### [Реакция и хладнокровие]

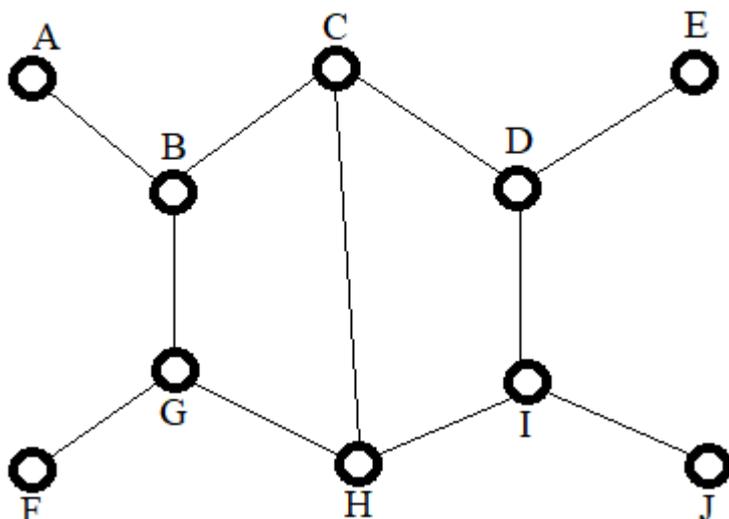
В некоторой игре персонаж имеет 5 характеристик - Сила, Интеллект, Реакция, Хладнокровие и Техника. Игрок выбирает для персонажа одну либо две характеристики, которые станут основными, одну либо две дополнительных характеристики. Остальные характеристики (от одной до трёх) персонаж не использует. В рамках исследования была проанализированы 1000 различных персонажей. Известно, что персонажей с двумя основными и двумя дополнительными характеристиками оказалось 130, а персонажей с двумя основными и одной дополнительной в 2 раза больше, чем персонажей с одной основной и двумя дополнительными характеристиками. Персонажи с основной характеристикой Реакция и дополнительной Хладнокровие составили 10% от всех персонажей, имеющих по одной основной и дополнительной характеристике. Определите максимальное количество персонажей с основной характеристикой Реакция и дополнительной Хладнокровие при котором такой результат исследования был бы возможен..

**Ответ: 84**

#### 5. Телекоммуникационные технологии (2 балла)

##### [Файерволы]

В некоторой локальной сети устройства соединены каналами передачи сообщений. Устройства обозначены заглавными буквами английского алфавита.



Соединения между устройствами защищены файерволами, пропускающими сообщение в любом направлении только в том случае, если оно соответствует условию файервола. Ниже приведена таблица условий. Символом  $m$  обозначено передаваемое сообщение,  $\&$  - побитовая конъюнкция,  $|$  - побитовая дизъюнкция.

Соединение	Условие файервола
AB	$m \& 1000 = 0$
BC	$m \& 11 = 10$
CD	$m \& 111 = 111$
DE	$m \& 1011 = 1000$
BG	$m \& 1 = 1$
CH	$m   0 = m$
DI	$m \& 1000 = 0$
FG	$m \& 101 = 101$
GH	$m   10101 = 10101$
HI	$m \& 100 = 100$
IJ	$m \& 1010 = 0$

Сообщения, передаваемые между устройствами, являются натуральными числами, записанными в двоичной системе счисления, возможно, с использованием ведущих нулей. Сообщение передаётся по кратчайшему (проходящему через минимально возможное число соединений) возможному пути. Определите наименьшее возможное двоичное число  $m$ , которое можно отправить в сообщении, которое сможет попасть из узла  $A$  в узел  $J$  и последовательность узлов, через которые сообщение пройдёт (включая начальный и конечный узел). В ответе запишите через пробел сначала двоичное число без ведущих нулей – искомое сообщение, а затем последовательность заглавных английских букв без пробелов – узлы на пути сообщения в порядке, в котором они были посещены. Если сообщение не может быть доставлено укажите NULL.

Пример записи ответа: 10101 ABCDE

Ответ: 101 ABGHIJ

## 6. Операционные системы (3 балла)

### [Чья маска?]

Сэм изучает регулярные выражения. Чтобы применить полученные знания, он написал параметризованный скрипт для поиска файла по маске. Маска задаётся в виде регулярного выражения. В качестве значений параметров могут быть использованы числа или заглавные буквы английского алфавита.

Скрипт имеет следующие параметры:

1. Искать в только текущей папке или в ней и её подпапках (передаётся значение  $S$  для поиска в текущей папке и  $S$  для поиска в ней и подпапках)
2. Если поиск осуществляется в подпапках, то какая максимальная глубина поиска (натуральное число - если глубина ограничена, буква  $A$  – если ограничения нет; число 1 соответствует поиску только в текущей папке)
3. После сколько совпадений остановить поиск (натуральное число - если необходимо ограничить число найденных совпадений, буква  $A$  – если ограничения нет). В случае, если не удастся найти заданное число совпадений, будут показаны те, что удалось найти.
4. Регулярное выражение, по которому осуществляется поиск. Формат регулярного выражения описан ниже.

Для задания регулярных выражений приняты следующие обозначения:

$c$  Любой неспециальный символ  $c$  соответствует самому себе. Специальными символами будем считать только символы  $[, ], \{, \}, *, +, -, ?$  – эти символы не могут по условию данной задачи встретиться в тексте.

$[...]$  Любой символ из  $...$ ; допустимы диапазоны типа  $a-z$  (последовательно идущие символы в алфавите); возможно объединение диапазонов, например  $[a-z0-9]$  и сочетание диапазонов и отдельных символов  $[a-z0-9~\#]$ .

$r^*$  Ноль или более вхождений символа  $r$ , может применяться и для диапазонов, например  $[a-z\#]^*$  означает ноль или более вхождений любых символов из диапазона от  $a$  до  $z$  или символа  $\#$  в любом порядке.

$r^+$  Одно или более вхождений символа  $r$ , может применяться и для диапазонов, например  $[a-z]^+$  означает одно или более вхождений любых символов из диапазона от  $a$  до  $z$  или символа  $>$  в любом порядке.

$r^?$  Ноль или одно вхождение символа  $r$ , может применяться и для диапазонов, например  $[a-z@]^?$  означает ноль или одно вхождение любого символа из диапазона от  $a$  до  $z$  или символа  $@$ .

$r1r2$  За символом или диапазоном  $r1$  следует символ или диапазон  $r2$ .

$\{ \}$  Число вхождений предыдущего выражения. Например, выражение  $[0-9]\{5\}$  соответствует подстроке из пяти десятичных цифр.

$^$  Символ начала строки. Регулярное выражение должно начинаться с этого символа.

$\$$  Символ конца строки. Регулярное выражение должно заканчиваться этим символом.

Пример: регулярное выражение  $^a+[a-z]\{5\}.[0-9]^*\$$  позволяет найти все последовательности символов, которые начинаются с одного или нескольких символов  $a$ , после которых идут ровно 5 маленьких латинских букв, затем точка и затем может следовать любое количество (в том числе ноль) арабских цифр.

Фрагмент файловой системы, на котором он проверял скрипт выглядит следующим образом:

folder1	folder2	folder3	folder4	folder5
folder2	folder3	folder5	vhs.exe	balloon.mp4
baobab.png	folder4	bavaria.mov	qsort.cpp	bunny_baks.jpg
abba_song.mp3	blablacar.zip	baldurs.mp4	mergesort.java	abcabcabc.doc
albania.zip	lalaland.mov	black.png	algorythm.java	
the_abs.doc	absolute.cpp	band.png	sort_algo.jar	
car_ab.jpg			alter_ego_a.mp4	
vabadab.doc			tart.zip	
the_asbolute.mov			alter_ego_b.mp4	

В первой строке каждого столбца указано название папки, далее – её содержимое. Все элементы с названием  $folder^*$ , где  $*$  является числом, являются папками. В момент запуска скрипта обработка начинается с содержимого  $folder1$  в алфавитном порядке. Сначала скрипт проверяет файлы, затем содержимое подпапок. В случае нескольких подпапок они также проверяются в алфавитном порядке.

Скрипт был запущен с параметрами  $S, 2, 5$  и некоторым регулярным выражением и вывел следующий результат:

albania.zip  
abba\_song.mp3  
baobab.png  
car\_ab.jpg  
the\_abs.doc

Определите, какие из представленных регулярных выражений могли быть использованы:

1.  $^{\wedge}[a-z]^*a[a-z]^*a[a-z0-9.]^*\$$
2.  $^{\wedge}a[a-z]^*\$$
3.  $^{\wedge}[a-z]^*a[a-z0-9.]^*\$$
4.  $^{\wedge}[a-z0-9.]{6}[a-z0-9.]^*\$$
5.  $^{\wedge}[a-z]^*a[a-z.]^*\$$
6.  $^{\wedge}[a-z0-9.]+\$$
7.  $^{\wedge}[a-z]^*a[a-z0-9]^*\$$

В ответ запишите через пробел в порядке возрастания номера всех регулярных выражений, которые могли обеспечить подобный результат.

*Примечание: Глубиной поиска в файловой системе называют число папок в пути от корневой до текущей, включая корень. Например, глубина поиска в корневой папке – 1, т.к. путь до неё содержит всего одну папку – её саму. Если текущая папка лежит в корневой папке – её глубина равна 2, и так далее. Количеством совпадений (количеством результатов поиска) называют число найденных файлов, которые удовлетворяют условиям поиска.*

**Ответ: 3 4 6**

## 7. Технологии программирования (2 балла)

### [Рассадка по аудитории]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	3 секунды
Ограничение по памяти	256 мегабайт

В начале учебного года на урок программирования пришли  $k$  школьников. Места для посадки в аудитории разбиты на  $n$  рядов, пронумерованных от 1 до  $n$ . В каждом ряду  $m$  мест, пронумерованных от 1 до  $m$ . Поскольку это начало учебного года, необходимо в первую очередь определиться с рассадкой.

По различным причинам (по состоянию здоровья или просто потому что захотелось), у некоторых школьников есть ограничения по тому, где они могут сидеть. Ограничения бывают двух видов:

школьник номер  $s_i$  может сидеть только на рядах с  $r_{i,1}$  по  $r_{i,2}$ ;

школьник номер  $s_i$  может сидеть только на местах с номерами с  $c_{i,1}$  по  $c_{i,2}$ .

Всего таких ограничений ровно  $p$ . Для одного и того же школьника может быть не более одного ограничения каждого из двух видов. Назовем школьника *недовольным*, если ему не было выдано место в аудитории, либо если хотя бы одно из его ограничений не выполнено.

Изначально ни одному школьнику не было назначено место. Затем преподаватель  $q$  раз отдавал поручения вида «школьнику  $s_i$  сесть на место  $c_i$  в ряду  $r_i$ ». Если при этом

или это место уже было кем-то занято,

или школьник  $s_i$  уже сидел на каком-то другом месте, а при такой пересадке количество невыполненных ограничений **строго увеличивалось**,

поручение игнорировалось, и школьник  $s_i$  оставался на месте. Обратите внимание, что если у школьника не было назначенного ему места, а назначенное место свободно, поручение будет выполнено вне зависимости от ограничений школьника.

Обработайте все поручения преподавателя и выведите в конце количество недовольных школьников.

### Формат входных данных

Решите задачу для  $t$  различных классов. В первой строке ввода дано целое число  $t$  — количество классов, для которых требуется решить задачу ( $1 \leq t \leq 100$ ). Затем следуют описания классов и последовательностей перемещений школьников.

В первой строке описания класса через пробел даны четыре целых числа  $n, m, k$  и  $p$  — размеры аудитории, количество школьников и количество ограничений ( $1 \leq n, m \leq 100, 1 \leq k \leq 1000; 0 \leq p \leq 2k$ ).

Каждая из следующих  $p$  строк содержит ограничение в формате «R  $s_i r_{i,1} r_{i,2}$ » или «C  $s_i c_{i,1} c_{i,2}$ » — либо ограничение на номер ряда, либо ограничение на номер места для школьника  $s_i$  ( $1 \leq s_i \leq k; 1 \leq r_{i,1} \leq r_{i,2} \leq n; 1 \leq c_{i,1} \leq c_{i,2} \leq m$ ). Гарантируется, что у каждого школьника есть не более одного ограничения каждого вида.

В следующей строке дано целое число  $q$  — количество поручений преподавателя ( $1 \leq q \leq 5000$ ). Каждая из следующих  $q$  строк описывает поручение преподавателя в формате «MOVE  $s_i r_i c_i$ » — школьнику  $s_i$  пересесть в ряд  $r_i$  на место  $c_i$  ( $1 \leq s_i \leq k; 1 \leq r_i \leq n; 1 \leq c_i \leq m$ ).

### Формат выходных данных

Для каждого из  $t$  классов выведите в отдельной строке единственное целое число — количество недовольных школьников в классе после обработки всех поручений преподавателя.

### Пример

Стандартный ввод	Стандартный вывод
3	3
2 2 4 0	1
3	97
MOVE 1 1 1	
MOVE 2 1 1	
MOVE 3 1 1	
2 2 4 2	
R 1 2 2	
C 3 1 1	
4	
MOVE 1 1 1	
MOVE 2 1 2	
MOVE 3 2 1	
MOVE 4 2 2	
3 3 100 2	
R 1 1 2	
R 2 2 3	
5	
MOVE 1 2 3	
MOVE 2 1 2	
MOVE 100 1 1	
MOVE 99 1 1	
MOVE 98 2 2	

## 8. Технологии программирования (4 балла)

### [Космические шашки]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

Как вам известно, находясь на орбите, космонавтам абсолютно нечего делать. Наши герои, представители небезызвестной династии покорителей космоса Шелби: Валентиныч и Валерьяч, решили взять с собой набор для игры в космические шашки, дабы не заскучать во время экспедиции Voyager-11.

Космические шашки представляют собой прямоугольное поле из  $n$  строк и  $m$  столбцов, каждая клетка которого может быть пустой или содержать либо черную, либо белую шашку. Строки пронумерованы от 1 до  $n$  сверху вниз, столбцы — от 1 до  $m$  слева направо.

Ходы делаются по очереди, ход заключается в передвижении своей шашки на незанятое поле одним из двух способов:

- Если есть свободная вперед по диагонали клетка, шашка может быть передвинута на одну клетку по диагонали. Белые могут совершить такой ход только по направлению **уменьшения** номера строки, а черные — по направлению **увеличения** номера строки.
- Если на соседней с шашкой диагональной клетке находится шашка соперника, и за ней имеется свободное поле, шашка может быть передвинута на следующую по диагонали клетку за шашкой соперника (при этом шашка соперника снимается с поля). Если после этого хода имеется продолжение для взятия других шашек соперника, ход может быть продолжен. Такой ход может производиться в любом из четырех направлений по диагонали.

Иными словами, можно либо сделать ход на одну клетку по диагонали, после чего остановиться, либо сразу начать брать шашки противника в произвольном количестве. Но нельзя сначала сделать ход первым способом, а затем продолжить его вторым.

При совершении очередного хода шашка игрока может стать *дамкой*. Белая шашка становится дамкой, если доходит до 1-й строки игрового поля, а черная — если доходит до  $n$ -й строки игрового поля.

В очередной день, играя долгую партию, Валентиныч задался вопросом: а может ли сейчас какая-то шашка за один ход стать дамкой? Зная текущее расположение шашек на поле и чей сейчас ход, помогите Валентинычу найти ответ на его вопрос.

#### Формат входных данных

Требуется найти ответ на задачу для  $t$  различных партий. В первой строке ввода дано целое число  $t$  — количество партий, для которых надо решить задачу ( $1 \leq t \leq 100$ ). Далее следуют  $t$  описаний текущего состояния партии.

В первой строке каждого описания партии через пробел даны два целых числа  $n$  и  $m$  — размеры поля, а также символ 'w', если сейчас ход белых, и 'b', если ход черных ( $1 \leq n, m \leq 1000$ ).

Следующие  $n$  строк описывают поле;  $i$ -я строка ввода описывает  $i$ -ю строку поля. Каждая строка содержит ровно  $m$  символов, каждый из которых может быть либо '.', если соответствующая клетка свободна, либо 'W', если на клетке расположена белая шашка, либо 'B', если на клетке расположена черная шашка.

Гарантируется, что сумма размеров полей ( $n \cdot m$ ) по всем  $t$  партиям не превосходит  $10^6$ .

#### Формат выходных данных

Для каждой партии выведите в отдельной строке через пробел два целых числа — номер строки и номер столбца, в которых расположена шашка, которая за ближайший ход может выйти в дамки.

Если такой шапки нет, выведите «-1 - 1» (без кавычек). Если таких шашек несколько, выберите шапку, расположенную на строке с минимальным номером, а из всех таких — в столбце с минимальным номером.

**Пример**

Стандартный ввод	Стандартный вывод
8	5 2
5 5 w	7 3
.....	9 6
..B..	3 3
.....	2 4
..B..	-1 -1
.W.W.	1 2
8 8 w	-1 -1
.....	
BBBBBBBB	
.B.B.B.B	
BBB.B.B.	
.W.W.W.W	
.B.....	
..W.....	
.....	
10 9 b	
.....B.B	
....B....	
.BB.B.BW.	
..W.....	
WBW.W.BB.	
..BBWBBWB	
B....BB.	
.B..WBB..	
....B...	
.....	
8 9 w	
.....	
BB.BBB..B	
.WW....B	
.....W..	
.....	
.....	
.B.....	
.....	
2 7 w	
B.BB...	
...WW..	
7 2 b	
..	
..	
..	
.W	
.W	
W.	
..	
2 8 b	
.B.BB...	
.....	
5 5 w	
.....	
.....	
.....	
.....	
.....	