

**Информатика, 9 – 11 классы, муниципальный этап
Практический тур**

Памятка участнику

- *На туре разрешается пользоваться только письменными принадлежностями и предоставленным компьютером.*
- *Апелляция по результатам практического тура не предусмотрена, но участник имеет право присутствовать при тестировании сданного им решения.*
- ***Время решения задач практического тура – 4 часа.***
- *Решение задач практического тура предусматривает ввод исходных данных из файла и вывод результатов в файл. Все файлы являются текстовыми.*
- *Время работы программы на любом тесте не может превышать 1 секунду.*
- *Файл исходных данных находится в текущем каталоге. Выходной файл также должен находиться в текущем каталоге.*
- *Формат входного и выходного файлов определен в условии задач.*
- *Решение принимается в виде файла с исходным текстом на языке Turbo Pascal 7.0. Для компиляции будет использоваться пакетный компилятор bpc с опциями по умолчанию.*
- *В первой строке файла с решением необходимо поместить комментарий с указанием фамилии, имени, класса и школы участника, а также задать идентификатор и название решаемой задачи.*
- ***Не забывайте периодически сохранять файлы!***

Задача А. Двоичные числа (10 баллов)

Найдите количество натуральных чисел Z , удовлетворяющих неравенству $A \leq Z \leq B$, таких, что в записи Z в двоичной системе счисления используется ровно 2 единицы. Например, если $A=10; B=20$; то таких чисел 5 (это числа $10=1010_2$; $12=1100_2$; $17=10001_2$; $18=10010_2$; $20=10100_2$).

Формат входного файла input.txt:

На вход программы поступают два числа, записанных через пробел — A, B ($0 \leq A \leq B \leq 10^9$)

Формат выходного файла output.txt:

Выведите одно число – количество чисел Z .

Input.txt	Output.txt
10 20	5

Задача В. Шахматный код (10 баллов)

Кроваво-красное солнце медленно поднималось над горизонтом, его теплые лучи согревали бескрайние просторы Зазеркалья. «Всем доброго утра», - сказала Алиса, заходя в кабачок Болванщика. Невнятное бормотание было ей ответом. «Что случилось, друзья?», - спросила она у завсегдатаев, задумчиво сидевших за столами, глубоко погруженных в свои мысли. «Видишь ли, Алиса», - пробормотал Белый Кролик, - «сегодня мы всю ночь играли с Болванщиком в шахматы, устали – сил нет, а прервать партию не можем – вдруг, пока мы спим, кто-то переставит фигуры!» При этих словах Кролик грозно посмотрел на кого-то, а Вы догадываетесь, кто это был, и сердито фыркнул. «Все элементарно просто», - сказала Алиса, - «давайте запишем положение фигур на бумаге, а потом восстановим положение фигур на доске».

- Как Вы знаете, шахматная доска состоит из 8 строк, обозначенных цифрами от 1 до 8, и 8 столбцов, обозначенных латинскими буквами от a до h. Нижняя строка имеет номер 1, а верхняя 8, самый левый столбец обозначается буквой a, последний – h.
- На доске присутствуют белые и черные фигуры: король (k), ферзь (q), ладья (r), слон (b), конь (n) и пешка (p).
- Будем обозначать черные фигуры маленькими латинскими буквами (k, q, r, b, n, p), а белые – заглавными (K, Q, R, B, N, P).
- Пустые клетки строки будем обозначать числом от 1 до 8, обозначающим длину непрерывной горизонтальной области пустых клеток, расположенных подряд в порядке слева направо.
- Символом «/» будем обозначать конец строки поля.

Задание: по имеющейся информации о шахматной доске составить строку, кодирующую положение шахматных фигур и, наоборот, по закодированной строке восстановить состояние шахматной доски.

Формат входного файла input.txt:

Первая строка содержит число 0 или 1. Число 0 означает первую подзадачу, то есть по шахматной доске получить закодированную строку. Число 1 означает обратную задачу – по закодированной строке восстановить положение фигур на шахматной доске.

Подзадача 0.

- Далее следует 8 строк по 8 символов каждая. Строчными латинскими символами обозначены черные фигуры (k, q, r, b, n, p), а белые – заглавными (K, Q, R, B, N, P), причем король (k), ферзь (q), ладья (r), слон (b), конь (n) и пешка (p). Символом «.» (точка) обозначены пустые клетки.

Подзадача 1.

- Далее следует одна строка, содержащая буквы (описанные выше), цифры и символ «/», разделяющий строки доски.

Формат выходного файла output.txt:

- Выведите в выходной файл закодированную строку состояния шахматной доски (подзадача 0) или состояние шахматной доски (подзадача 1).

Input.txt	Output.txt
<pre>0 rnbqkbnr pppppppp PPPPPPPP RNBQKBNR</pre>	<pre>rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR</pre> <p>{ данный пример описывает первоначальное положение шахматных фигур на игровой доске: сверху черные фигуры, снизу – белые. Кодирование идет слева - направо, сверху- вниз }</p>
<pre>1 r6r/p2ppppp/1pp5/8/8/8/P3PPPP/RNBQKBNR</pre> <p>{ у черных только 2 ладьи и 8 пешек, у белых – все фигуры и 5 пешек }</p>	<pre>r.....r p..ppppp .pp..... P...PPPP RNBQKBNR</pre>

Особенности тестов:

- 50% тестов подзадача 0.
- 50% тестов подзадача 1.

Задача С. Луноход (15 баллов)

Жители Зазеркалья любили проводить свободное время за различными играми, смысл которых нам не всегда понятен. Однако часто безобидные игры заканчивались спорами, порой перераставшими в крупные ссоры. Вот и сейчас, зайдя в кабачок Болванщика, Алиса кожей ощутила наэлектризованную атмосферу конфликта. Белый Кролик и Болванщик свирепо смотрели друг на друга, молнии сверкали в их глазах. «Если ты не признаешь мою победу», - кричал Болванщик, - «то я введу против тебя санкции, и Ряженку ты будешь пить в другом месте!» «А я в ответ введу продуктовое эмбарго», - шипел Кролик. «Тише, друзья», - сказала Алиса, - «давайте разберемся в причинах вашей ссоры». «Мы играли в настольную игру Луноход, я выиграл, а ОН не хочет это признать!», - завопил Кролик. «НЕТ! Выиграл я, у меня все ходы записаны!», - возразил Болванщик. «Контора пишет», - услышал он в ответ. «Спокойствие, только спокойствие!», - процитировала Алиса любимый мультфильм, - «сейчас разберемся».

В игре «Луноход» участвуют двое и ходят по очереди. У каждого участника есть свой луноход, которым можно управлять, перемещая его в соседние клетки игрового поля при помощи команд: L, R, D, U (влево, вправо, вниз, вверх). Игровое поле разбито на клетки, каждая клетка может быть пустой, содержать кусок стены или монетку (номиналом 1 - 9). Задача игрока собрать монетки суммарной стоимостью большей, чем наберет противник (в этом случае игрок победит). Первый игрок (как ни странно) ходит первым, его луноход располагается в верхнем левом углу карты, луноход второго – в нижнем правом. Игроки по очереди делают ход (L,R,D,U), заставляя луноход перемещаться в заданном направлении. Если в этой клетке располагается монетка, то игрок прибавляет ее стоимость к набранной сумме, при этом монетка удаляется с карты. Луноходы стартуют из гарантированно пустой клетки.

Игра заканчивается, если:

- в данной клетке располагается стена;
- в результате хода луноход вышел за пределы карты;
- все монетки на карте собраны;
- луноходы обоих игроков оказались в одной клетке.

(хотя протокол игры может продолжиться – играют жулики!). Выигрывает тот игрок, который наберет большую сумму очков, при равенстве очков объявляется ничья.

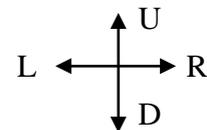
Задание: по имеющемуся протоколу игры напишите программу, которая определит победителя и количество очков, набранных каждым игроком.

Формат входного файла input.txt:

Первая строка содержит три числа N, M, K – размеры игрового поля ($2 \leq N$, $M \leq 100$) и количество ходов в партии ($1 \leq K \leq 10000$).

Далее следует N строк, каждая содержит M символов (без пробелов):

- ‘.’ (точка) – клетка поля пуста;
- ‘#’ – в клетке располагается стена (препятствие);
- ‘1’..‘9’ – в клетке располагается монета с указанным номиналом.



В следующей строке располагается K символов (L,R,D,U), задающих направление перемещения лунохода соответствующего игрока (нечетные – первого, четные – второго).

Формат выходного файла output.txt:

Выведите в первой строке выходного файла номер победителя (1 или 2) или число 0, если игроки набрали одинаковое количество очков. Во второй строке выведите через пробел два числа – количество очков, которые набрали первый и второй игрок соответственно.

Input.txt	Output.txt
7 5 131... ...9. .###.# .12.# #.3. RLDURLRLRULLD	1 10 6

Задача D. Шифр Цезаря (20 баллов)

Страшный шпионский скандал разразился в Зазеркалье – оказывается, Черная Королева не только шпионила за перепиской всех жителей Зазеркалья, но и взламывала их аккаунты Вконтакте и Одноклассниках, а также лайкала под чужим ником никчемные фотографии и селфи горе-фотографов. Алиса была в бешенстве, когда обнаружила, что под ее ником в Одноклассниках кто-то удалил пост «Крем наш!» рекламного ролика одной очень известной парфюмерной фирмы. «Это уже перебор!» - подумала Алиса и решила отомстить. Для этого она замыслила вскрыть шифр ЧК. Известно, что для засекречивания своей переписки, ЧК пользуется шифром Цезаря, а именно:

1. вся переписка ведется с использованием латинского алфавита (на транслите);
2. выбирается целое число K ($1 \leq K < 10\,000\,000$);
3. каждый символ латинского алфавита циклически сдвигается на K позиций вправо. Например, при $K=1$ из символа ‘a’ получается символ ‘b’, из ‘c’ – ‘d’, ‘z’ – ‘a’ и т.д. Строчные и заглавные буквы считаются различными и сдвигаются внутри своей группы.

Проблема Алисы состояла в том, что число K ей было неизвестно, однако Алиса узнала, что в сообщении обязательно встречается слово S , а все сообщение гарантировано дешифруется однозначно. Словом считается непрерывная последовательность латинских букв, ограниченная либо пробелом, либо другими символами (не буквами), либо началом или концом строки.

Помогите Алисе вскрыть шифр Черной Королевы, напишите программу, которая по данной строке и слову S , определит число K и расшифрует сообщение ЧК.

Формат входного файла input.txt:

В первой строке файла находится слово S (длина не более 200 символов).
 Во второй строке число N – длина зашифрованного сообщения ($1 < N \leq 10000$).
 В третьей строке располагается N символов – зашифрованное сообщение.

Формат выходного файла output.txt:

Выходной файл должен содержать N символов – расшифрованное сообщение.

Input.txt	Output.txt
Alisa {слово S } 17 {длина строки} Vcd, Bmjtb, lAba.	Abc, Alisa, lZaz.
dzigu 25 hocoov grev njsqe cfudbx	xeseel wful dzigu svktrn

Особенности тестов:

- $N \leq 255$ – не менее 6 тестов

Ограничения и особенности тестов:

- 5 секунд на тест.
- $2 \leq N \leq 10$ порядка 10 тестов
- $2 \leq N \leq 20$, количество коридоров не более 10^8 – 15 тестов
- $2 \leq N \leq 20$ – 5 тестов

Задача F. Управление роботом (30 баллов)

Однажды Алисе приснился странный сон, будто стоит она на огромной площади, заполненной толпой народа, причем все жители прыгают, выкрикивая – «Кто не скачет, тот стоит!» «Что за жуткий бред?», - подумала Алиса и немедленно проснулась, - «слава богу, у нас подобное невозможно», - усмехнулась она. «Зачем мы столько лет изучали историю, если ее уроки все равно никто не помнит? Кто выведет из транса бедных жителей Зазеркалья? Кто, кроме НЕЕ?» - такие мысли фейерверком взрывались в голове Алисы. «Пора заняться своим любимым делом – освобождением Белой Королевы», - подумала Алиса, - «ведь только ее внутренняя доброта и чистота способны вернуть разум человечеству!»

Чтобы добраться до пещеры, в которой томилась Белая Королева, необходимо было сесть в транспортного робота. Робот мог перемещаться только в четырех направлениях: вверх, вниз, влево, вправо. Все Зазеркалье было разбито на квадратные зоны (N×M), робот мог перемещаться в соседнюю зону, повинаясь, заложенной в него программе, высаживать или забирать в ней пассажиров.

Программа состояла из команд, каждая начиналась с новой строки:

- L – переместиться в зону, расположенную слева;
- R – переместиться в зону, расположенную справа;
- D – переместиться в зону, расположенную снизу;
- U – переместиться в зону, расположенную сверху;
- WHILE условие ДЕЙСТВИЕ (только одна из команд L, R, D, U) – повторяет заданное действие, пока условие истинно. Слова WHILE, условие и действие разделены строго одним пробелом;
- В качестве условия использовалась одна из четырех функций:
 - UEMPTY – проверяет зону, расположенную выше, возвращает истину, если зона свободна, то есть не содержит стену.
 - DEMPTY – проверяет зону, расположенную ниже, возвращает истину, если зона свободна, то есть не содержит стену.
 - LEMPTY – проверяет зону, расположенную левее, возвращает истину, если зона свободна, то есть не содержит стену.
 - REMPTY – проверяет зону, расположенную правее, возвращает истину, если зона свободна, то есть не содержит стену.

Каждая зона Зазеркалья могла:

- Быть пустой (обозначим ее точкой ‘.’);
- Содержать непроходимую стену или препятствие (обозначим ее ‘#’);
- Содержать пещеру (обозначим ее ‘@’).

Транспортный робот стартует из некоторой зоны и движется по Зазеркалью, подчиняясь заложенной в него программе. Робот разрушается, если в результате выполнения команды он вынужден переместиться за пределы карты или в зону с препятствием.

Ваша задача по имеющимся программе движения робота и карте Зазеркалья, определить количество всех зон, стартуя из которых, робот сможет попасть в зону с пещерой Белой Королевы и посадить там Алису.

Формат входного файла input.txt:

Первая строка содержит два числа N и M, разделенные пробелом. ($2 \leq N$, $M \leq 50$) - размеры Зазеркалья.

В следующих N строках по M символов в каждой задается карта Зазеркалья (**нет пробелов!**):

- символ '.' (точка) означает, что это пустая клетка;
- символ '@' обозначает пещеру Белой Королевы;
- '#' – обозначает непроходимое препятствие;

В следующей строке дается число K ($1 \leq K \leq 1000$) - количество команд, которые заданы транспортному роботу.

В последующих K строках задается последовательность из K команд – программа, управляющая передвижением транспортного робота.

Формат выходного файла output.txt:

Вывести единственное число – количество зон, из которых робот сможет добраться до пещеры (зону с пещерой считать не надо).

Input.txt	Output.txt
<pre> 5 7# .####.#.. ..####. ..#...@ 5 U WHILE EMPTY R WHILE EMPTY D R WHILE EMPTY D </pre>	<pre> 2 Символом 0 обозначены зоны, двигаясь из которых, робот доберется до пещеры:# 0####0.#.. ..####. ..#...@ </pre>
<pre> 10 10##.#... .#...#... ...####... ...#..... .#....#...#@.#.... ..##..... .##...#... 7 L L U L L U L </pre>	<pre> 3 ##.#... .#...#... ...####... ...#..... .#....#...#@0#.... ..##..0... .##...#..0. </pre>