

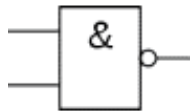
## Информатика. 11 класс Вариант 1

*Работа рассчитана на 240 минут.*

***В задачах 1-2 необходимо привести полное обоснованное решение на бланках олимпиады.***

**Задача 1.** У Бельчонка есть два друга Совенок и Зайчонок. Каждое утро они решают, пойдут ли они гулять или останутся дома. Решение принимается большинством голосов. Так как Бельчонок увлекается электроникой, то он решил сделать схему из логических элементов, на вход которой подавалось бы решение каждого из друзей о прогулке (0 или 1), а на выходе было решение, принятое большинством (0 или 1). Проблема в том, что в запасах у Бельчонка не нашлось стандартных логических элементов И, ИЛИ, НЕ, зато было много элементов И-НЕ, имеющих следующую таблицу истинности и обозначение

X	Y	И-НЕ
0	0	1
0	1	1
1	0	1
1	1	0



Нарисуйте логическую схему, имеющую три входа Б, С, З (Бельчонок, Совенок, Зайчонок) и один выход, на котором появляется 1, если большинство из друзей хотят идти гулять. В схеме можно использовать только элементы И-НЕ. Необходимо построить схему, содержащую минимальное количество логических элементов.

Ответ и решение к задаче № 1 указывается на бланке Олимпиады.

**Задача 2.** Бельчонок построил Робота, который передвигается по тропинке, разделенной на ячейки, в каждой из которых может лежать либо шишка (Ш), либо гриб (Г), либо ничего (Н). Робот передвигается по тропинке, переходя в соседнюю ячейку вправо или влево. При этом он может оставлять в ячейке шишку, гриб, либо забирать из ячейки то, что там лежит. У Робота есть неограниченный запас шишек и грибов.

Программа для Робота представляет собой пронумерованную последовательность команд, которые могут быть следующими:

Н N	Забрать из текущей ячейки то, что там лежит и перейти к выполнению команды, записанной в строке с номером N
Г N	Положить в текущую ячейку гриб и перейти к выполнению команды, записанной в строке с номером N
Ш N	Положить в текущую ячейку шишку и перейти к выполнению команды, записанной в строке с номером N

> N	Робот перемещается в ячейку справа и переходит к выполнению команды, записанной в строке с номером N
< N	Робот перемещается в ячейку слева и переходит к выполнению команды, записанной в строке с номером N
? NH, NG, NSH	Робот проверяет, что находится в текущей ячейке и переходит к выполнению команды, записанной в строке с номером NH, если в текущей ячейке ничего нет, к строке с номером NG, если в текущей ячейке лежит гриб и к строке с номером NSH, если в текущей ячейке лежит шишка. При этом содержимое текущей ячейки не меняется.

В начальный момент Робот расположен левее «кучи» грибов и шишек. Пустых ячеек в «куче» нет. Он должен, следуя программе, разделить данную кучу на две, оставив в левой только шишки, а в правой – грибы. Кучи должны быть разделены одной пустой ячейкой. Расположение новых куч относительно исходной кучи может быть любым.

Пример начальных данных (Р – позиция Робота)

	Р			Ш	Г	Г	Ш	Ш											
--	---	--	--	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--

Результат работы (положение Робота после завершения программы не принципиально)

				Ш	Ш		Г	Г	Г		Р								
--	--	--	--	---	---	--	---	---	---	--	---	--	--	--	--	--	--	--	--

Изложите сначала алгоритм на естественном языке, а затем напишите программу для Робота, следуя которой он выполнит задание.

Ответ и решение к задаче № 2 указывается на бланке Олимпиады.

**Задача 3.** У Бельчонка есть электрический робот, который перемещается по прямоугольному полю. Робот начинает движение из верхней левой клетки и заканчивает в правой нижней. При этом он может перемещаться только вправо и вниз. Робот работает на электрической батарее, в начальный момент её заряд равен значению, записанному в верхней левой клетке. В каждой клетке поля записано число, означающее, на сколько процентов в данной клетке Робот может подзарядить свою батарею. Больше 100% батарея зарядиться не может. Если заряд батареи станет 0%, Робот прекращает дальнейшее движение. Заряд батареи расходуется следующим образом: если Робот совершает то же движение, что и на предыдущем шаге (например, два раза подряд двигается вниз), то на переход из клетки в клетку он тратит 10% заряда батареи, а если он совершает движение, отличное от того, которое было на предыдущем шаге (например, он двигался вправо, а потом вниз), то ему приходится сначала

повернуться, на что расходуется 5% заряда батареи, а только потом перейти в другую клетку. Таким образом, в этом случае он израсходует 15% заряда.

Задача Бельчонок – провести Робота из верхней левой клетки в правую нижнюю так, чтобы заряд батареи на финише оказался наибольшим. На основе приведенных в [файле данных](#) определите максимальное значение заряда батареи Робота в финишной ячейке. Для решения задачи нужно использовать только средства электронных таблиц (формулы и/или функции).

В качестве ответа нужно записать найденное число на бланке Олимпиады и прикрепить таблицу с расчетами в формате

ФИО участника класс площадка номер задания:

<https://www.dropbox.com/request/Oi6THwOs3U93AdYiVpuH>

**Задача 4.** В лесу есть несколько полянок, которые все соединены тропинками. Тропинки расположены так, что между любыми двумя полянками есть один и только один маршрут, проходящий, возможно, через другие полянки. На каждой из полянок растут ягоды. Бельчонок и Мышонок играют в такую игру. В начале они находятся на какой-то из полянок. Бельчонок съедает все ягоды на полянке, выбирает следующую полянку, которая соединена с текущей, и они вместе с Мышонком убегают на выбранную Бельчонок полянку. На ней уже Мышонок съедает все ягоды, выбирает следующую полянку, они вдвоем перемещаются на нее. И так далее по очереди. На полянки с уже съеденными ягодами перемещаться по условиям игры нельзя. Понятно, что рано или поздно друзьям некуда будет перемещаться после того, как кто-то из них съел ягоды на текущей полянке. Тот, кто последним съел ягоды, считается победившим в игре.

Напишите программу, которая по данным о том, как соединены полянки, и номеру полянки, с которой друзья начинают игру, определит, кто выиграет, если каждый игрок будет придерживаться оптимальной стратегии.

Входные данные

Первая строка входного текстового файла содержит два целых:  $N$  - количество полянок ( $N \leq 1000$ ) и  $K$  - номер начальной полянки. Следующие  $N - 1$  строк содержат пары целых чисел - номера полянок, соединенных тропинкой. По всем тропинкам можно перемещаться в обе стороны. Полянки пронумерованы с 1 до  $N$ .

Выходные данные

Программа должна выводить  $M$ , если выигрывает Мышонок (то есть, Бельчонок, начинающий игру, в любом случае проиграет) или  $Vx$  если выигрывает Бельчонок. Здесь  $x$  – номер полянки, на которую Бельчонок

должен перейти после своего первого хода. Если  $x$  определяется неоднозначно, то следует указать наименьшее из возможных значений.

Пример:

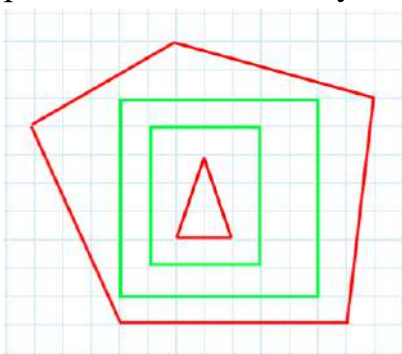
Входные данные	Выходные данные
2 1 1 2	M
4 1 1 2 1 3 3 4	B3

Данные для задачи доступны по ссылке: <https://crm.dovuz.sfu-kras.ru/~O0I97>

В качестве ответа нужно записать найденные числа на бланке Олимпиады и прикрепить исходный файл программы в формате ФИО участника\_класс\_площадка\_номер задания.txt:

<https://www.dropbox.com/request/Oi6THwOs3U93AdYiVpuH>

**Задача 5.** Бельчонок и Зайчонок скачали редактор `polygon`, позволяющий рисовать произвольные многоугольники. Бельчонок все многоугольники рисует зеленым цветом, а Зайчонок – красным. Чтобы рисунок получился красивым (по мнению друзей), каждый многоугольник расположен так, что  $K$  многоугольников находятся внутри него, а сам он находится внутри  $N-K-1$  многоугольников ( $N$  – общее количество многоугольников,  $0 \leq K \leq N-1$ ). При этом никакие два многоугольника не пересекаются ни в одной точке. Пример расположения многоугольников приведен на рисунке.



Программа `polygon` сохраняет в текстовом файле информацию о цвете многоугольника и координатах всех его вершин. Порядок вершин может быть любым (зависит от того, с какой вершины начали рисовать многоугольник). Многоугольники сохраняются в файле в том же порядке, в котором их рисовали. Начало координат расположено в верхнем левом углу, ось  $x$  направлена вправо, ось  $y$  – вниз.

Используя данные в файле, который сформировала программа `polygon`, определите, сколько красных многоугольников расположено так, что и внутри

них и снаружи них находится хотя бы один многоугольник зеленого цвета. Для приведенного выше рисунка это количество равно нулю.

#### Входные данные

Первая строка входного текстового файла содержит целое число  $N$  – общее количество многоугольников. Далее следуют  $N$  строк ( $N \leq 10000$ ), в каждой из которых сначала записан символ  $R$  или  $G$ , обозначающий цвет многоугольника, а затем через пробел несколько (в зависимости от количества углов многоугольника) пар  $(x, y)$  целых неотрицательных чисел – координаты всех вершин многоугольника.

#### Выходные данные

Программа должна выводить единственное число – количество красных многоугольников, которые расположены так, что и внутри них и снаружи них находится хотя бы один многоугольник зеленого цвета.

Пример (соответствует рисунку выше):

Входные данные	Выходные данные
4 R 6 8 7 5 8 8 G 11 3 11 10 4 10 4 3 R 13 3 6 1 1 4 4 11 12 11 G 9 9 5 9 5 4 9 4	0

Данные для задачи доступны по ссылке: <https://crm.dovuz.sfu-kras.ru/~O0I97>

В качестве ответа нужно записать найденные числа на бланке Олимпиады и прикрепить исходный файл программы в формате ФИО участника\_класс\_площадка\_номер задания.txt:

<https://www.dropbox.com/request/Oi6THwOs3U93AdYiVpuH>

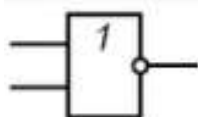
## Информатика. 11 класс Вариант 2

*Работа рассчитана на 240 минут.*

***В задачах 1-2 необходимо привести полное обоснованное решение на бланках олимпиады.***

**Задача 1.** У Бельчонка есть два друга Совенок и Зайчонок. Каждое утро они решают, пойдут ли они гулять или останутся дома. Решение принимается большинством голосов. Так как Бельчонок увлекается электроникой, то он решил сделать схему из логических элементов, на вход которой подавалось бы решение каждого из друзей о прогулке (0 или 1), а на выходе было решение, принятое большинством (0 или 1). Проблема в том, что в запасах у Бельчонка не нашлось стандартных логических элементов И, ИЛИ, НЕ, зато было много элементов ИЛИ-НЕ, имеющих следующую таблицу истинности и обозначение

X	Y	И-НЕ
0	0	1
0	1	0
1	0	0
1	1	0



Нарисуйте логическую схему, имеющую три входа Б, С, З (Бельчонок, Совенок, Зайчонок) и один выход, на котором появляется 1, если большинство из друзей хотят идти гулять. В схеме можно использовать только элементы ИЛИ-НЕ. Необходимо построить схему, содержащую минимальное количество логических элементов.

Ответ и решение к задаче № 1 указывается на бланке Олимпиады.

**Задача 2.** Бельчонок построил Робота, который передвигается по тропинке, разделенной на ячейки, в каждой из которых может лежать либо шишка (Ш), либо гриб (Г), либо ничего (Н). Робот передвигается по тропинке, переходя в соседнюю ячейку вправо или влево. При этом он может оставлять в ячейке шишку, гриб, либо забирать из ячейки то, что там лежит. У Робота есть неограниченный запас шишек и грибов.

Программа для Робота представляет собой пронумерованную последовательность команд, которые могут быть следующими:

Н N	Забрать из текущей ячейки то, что там лежит и перейти к выполнению команды, записанной в строке с номером N
Г N	Положить в текущую ячейку гриб и перейти к выполнению команды, записанной в строке с номером N
Ш N	Положить в текущую ячейку шишку и перейти к выполнению команды, записанной в строке с номером N



на предыдущем шаге (например, он двигался влево, а потом вверх), то ему придется сначала повернуться, на что расходуется 5% заряда батареи, а только потом перейти в другую клетку. Таким образом, в этом случае он израсходует 15% заряда.

Задача Бельчонок – провести Робота из нижней правой клетки в левую верхнюю так, чтобы заряд батареи на финише оказался наибольшим. На основе приведенных в [файле данных](#) определите максимальное значение заряда батареи Робота в финишной ячейке. Для решения задачи нужно использовать только средства электронных таблиц (формулы и/или функции).

В качестве ответа нужно записать найденное число на бланке Олимпиады и прикрепить таблицу с расчетами в формате

ФИО участника класс площадка номер задания:

<https://www.dropbox.com/request/Oi6THwOs3U93AdYiVpuH>

**Задача 4.** В лесу есть несколько полянок, которые все соединены тропинками. Тропинки расположены так, что между любыми двумя полянками есть один и только один маршрут, проходящий, возможно, через другие полянки. На каждой из полянок живет волк на привязи, который перейти на другую полянку не может. Отважные Бельчонок и Зайчонок играют в такую игру. В начале они находятся на какой-то из полянок. Бельчонок дразнит волка, находящегося на этой полянке, выбирает следующую полянку, которая соединена с текущей, и они вместе с Зайчонок убегают на выбранную Бельчонок полянку. На ней уже Зайчонок дразнит волка, выбирает следующую полянку, они вдвоем перемещаются на нее. И так далее по очереди. Возвращаться на уже посещенные полянки со злым волком нельзя. Понятно, что рано или поздно кто-то из друзей, после того, как раздражил Волка, не сможет сделать ход, то есть убежать будет некуда. Тот, кто не может сделать ход, проигрывает.

Напишите программу, которая по данным о том, как соединены полянки, и номеру полянки, с которой друзья начинают игру, определит, кто выиграет, если каждый игрок будет придерживаться оптимальной стратегии.

Входные данные

Первая строка входного текстового файла содержит два целых:  $N$  - количество полянок ( $N \leq 1000$ ) и  $K$  - номер начальной полянки. Следующие  $N - 1$  строк содержат пары целых чисел - номера полянок, соединенных тропинкой. По всем тропинкам можно перемещаться в обе стороны. Полянки пронумерованы с 1 до  $N$ .

Выходные данные



Программа должна выводить  $Z$ , если выигрывает Зайчонок (то есть, Бельчонок, начинающий игру, в любом случае проиграет) или  $Vx$  если выигрывает Бельчонок. Здесь  $x$  – номер полянки, на которую Бельчонок должен перейти после своего первого хода. Если  $x$  определяется неоднозначно, то следует указать наименьшее из возможных значений.

Пример:

Входные данные	Выходные данные
3 1 1 2 2 3	Z
4 1 1 2 1 3 3 4	V2

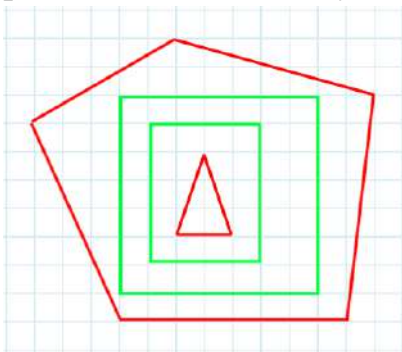
Данные для задачи доступны по ссылке: <https://crm.dovuz.sfu-kras.ru/~00197>

В качестве ответа нужно записать найденные числа на бланке Олимпиады и прикрепить исходный файл программы в формате ФИО

участника\_класс\_площадка\_номер задания.txt:

<https://www.dropbox.com/request/Oi6THwOs3U93AdYiVpuH>

**Задача 5.** Бельчонок и Зайчонок скачали редактор `polygon`, позволяющий рисовать произвольные многоугольники. Бельчонок все многоугольники рисует зеленым цветом, а Зайчонок – красным. Чтобы рисунок получился красивым (по мнению друзей), каждый многоугольник расположен так, что  $K$  многоугольников находятся внутри него, а сам он находится внутри  $N-K-1$  многоугольников ( $N$  – общее количество многоугольников,  $0 \leq K \leq N-1$ ). При этом никакие два многоугольника не пересекаются ни в одной точке. Пример расположения многоугольников приведен на рисунке.



Программа `polygon` сохраняет в текстовом файле информацию о цвете многоугольника и координатах всех его вершин. Порядок вершин может быть любым (зависит от того, с какой вершины начали рисовать многоугольник). Многоугольники сохраняются в файле в том же порядке, в котором их

рисовали. Начало координат расположено в верхнем левом углу, ось  $x$  направлена вправо, ось  $y$  – вниз.

Используя данные в файле, который сформировала программа `polygon`, найдите количество углов и цвет самого маленького (который содержится внутри всех остальных многоугольников) и самого большого (который содержит в себе все остальные многоугольники) многоугольника. Для приведенного выше рисунка это 3R и 5R (в ответе используются обозначения R – красный, G – зеленый)

#### Входные данные

Первая строка входного текстового файла содержит целое число  $N$  – общее количество многоугольников. Далее следуют  $N$  строк ( $N \leq 10000$ ), в каждой из которых сначала записан символ R или G, обозначающий цвет многоугольника, а затем через пробел несколько (в зависимости от количества углов многоугольника) пар  $(x, y)$  целых неотрицательных чисел – координаты всех вершин многоугольника.

#### Выходные данные

Строка, содержащая количество углов и цвет самого маленького (который содержится внутри всех остальных многоугольников) и самого большого (который содержит в себе все остальные многоугольники) многоугольника.

Пример (соответствует рисунку выше):

Входные данные	Выходные данные
4 R 6 8 7 5 8 8 G 11 3 11 10 4 10 4 3 R 13 3 6 1 1 4 4 11 12 11 G 9 9 5 9 5 4 9 4	3R 5R

Данные для задачи доступны по ссылке: <https://crm.dovuz.sfu-kras.ru/~O0197>

В качестве ответа нужно записать найденные числа на бланке Олимпиады и прикрепить исходный файл программы в формате ФИО

участника\_класс\_площадка\_номер задания.txt:

<https://www.dropbox.com/request/Oi6THwOs3U93AdYiVpuH>

## Информатика. 11 класс

3 вариант

*Работа рассчитана на 240 минут.*

***Все решения должны быть полными и обоснованными.***

### ОБРАТИТЕ ВНИМАНИЕ!

В задачах 1-2 необходимо привести полное обоснованное решение на бланках олимпиады. Для решения данных задач нельзя использовать языки программирования. Решение необходимо оформить на бланке Олимпиады

Для задачи 3 в бланк заносится только ответ. По ссылке необходимо прикрепить файл, при помощи которого вы получили ответ.

Для задач 4-5 в бланк заносится только полученный ответ. По ссылке необходимо прикрепить файлы с программой, с помощью которой вы получили данные ответы

Исходные файлы для заданий находятся в общем хранилище для каждого класса, содержащем папки, названия которых соответствуют номеру задания (например, папка task3 содержит данные для задания 3). В каждой папке даны исходные файлы, именованные по принципу test\_номер\_класса\_номер\_варианта\_номер\_задания\_номер\_исходного\_файла.txt

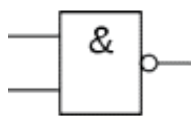
Например, файлы test\_11klass\_v1\_4\_1.txt, test\_11klass\_v1\_4\_2.txt, test\_11klass\_v1\_4\_3.txt содержат исходные данные для задачи 4 первого варианта 11-го класса. Участник олимпиады должен «пропустить» через программу все 3 исходных файла.

Исходные файлы для выполнения заданий: <https://crm.dovuz.sfu-kras.ru/~0SV2t>

### Задача 1.

У Бельчонка есть два друга Совенок и Зайчонок. Каждое утро они решают, пойдут ли они гулять или останутся дома. Решение принимается большинством голосов, но так как Бельчонок – старший из них, то его решение решающее. Если он не хочет гулять, то все друзья остаются сидеть по домам. Так как Бельчонок увлекается электроникой, то он решил сделать схему из логических элементов, на вход которой подавалось бы решение каждого из друзей о прогулке (0 или 1), а на выходе было решение, принятое большинством (0 или 1). Проблема в том, что в запасах у Бельчонка не нашлось стандартных логических элементов И, ИЛИ, НЕ, зато было много элементов И-НЕ, имеющих следующую таблицу истинности и обозначение

X	Y	И-НЕ
0	0	1
0	1	1
1	0	1
1	1	0



Нарисуйте логическую схему, имеющую три входа Б, С, З (Бельчонок, Совенок, Зайчонок) и один выход, на котором появляется 1, если большинство



Результат работы (положение Робота после завершения программы не принципиально)

	Р			Г	Г		Ш	Ш	Ш									
--	---	--	--	---	---	--	---	---	---	--	--	--	--	--	--	--	--	--

Изложите сначала алгоритм на естественном языке, а затем напишите программу для Робота, следуя которой он выполнит задание.

### Задача 3.

У Бельчонка есть электрический робот, который перемещается по прямоугольному полю. Робот начинает движение из нижней левой клетки и заканчивает в верхней правой. При этом он может перемещаться только влево и вверх. Робот работает на электрической батарее, в начальный момент её заряд равен значению, записанному в левой нижней клетке. В каждой клетке поля записано число, означающее, на сколько процентов в данной клетке Робот может подзарядить свою батарею. Больше 100% батарея зарядиться не может. Если заряд батареи станет 0%, Робот прекращает дальнейшее движение. Заряд батареи расходуется следующим образом: если Робот совершает то же движение, что и на предыдущем шаге (например, два раза подряд двигается вверх), то на переход из клетки в клетку он тратит 10% заряда батареи, а если он совершает движение, отличное от того, которое было на предыдущем шаге (например, он двигался вверх, а потом вправо), то ему приходится сначала повернуться, на что расходуется 5% заряда батареи, а только потом перейти в другую клетку. Таким образом, в этом случае он израсходует 15% заряда.

Задача Бельчонка – провести Робота из нижней левой клетки в правую верхнюю так, чтобы заряд батареи на финише оказался наибольшим. В качестве ответа нужно указать максимальное значение заряда батареи Робота в финишной ячейке. Для решения данной задачи можно использовать только средства электронных таблиц. Ответ требуется вписать в бланк и приложить файл с формулами и расчетами.

### Задача 4.

В лесу есть несколько полянок, которые все соединены тропинками. Тропинки расположены так, что между любыми двумя полянками есть один и только один маршрут, проходящий, возможно, через другие полянки. На каждой из полянок растут ягоды. Бельчонок и Мышонок играют в такую игру. В начале они находятся на какой-то из полянок. Бельчонок съедает все ягоды на полянке, выбирает следующую полянку, которая соединена с текущей, и они вместе с Мышонком убегают на выбранную Бельчонок полянку. На ней уже

Мышонок съедает все ягоды, выбирает следующую полянку, затем они вдвоем перемещаются на нее. И так далее по очереди. На полянки с уже съеденными ягодами по условиям игры перемещаться нельзя. Понятно, что рано или поздно друзьям некуда будет перемещаться после того, как кто-то из них съел ягоды на текущей полянке. Тот, кто последним съел ягоды, считается победившим в игре.

Напишите программу, которая по данным о том, как соединены полянки, и номеру полянки, с которой друзья начинают игру, определит, кто выиграет, если каждый игрок будет придерживаться оптимальной стратегии.

Входные данные

Первая строка входного текстового файла содержит два целых:  $N$  - количество полянок ( $N \leq 1000$ ) и  $K$  - номер начальной полянки. Следующие  $N - 1$  строк содержат пары целых чисел - номера полянок, соединенных тропинкой. По всем тропинкам можно перемещаться в обе стороны. Полянки пронумерованы с 1 до  $N$ .

Выходные данные

Программа должна выводить  $M$ , если выигрывает Мышонок (то есть, Бельчонок, начинающий игру, в любом случае проиграет) или  $Vx$  если выигрывает Бельчонок. Здесь  $x$  – номер полянки, на которую Бельчонок должен перейти после своего первого хода. Если  $x$  определяется неоднозначно, то следует указать наименьшее из возможных значений.

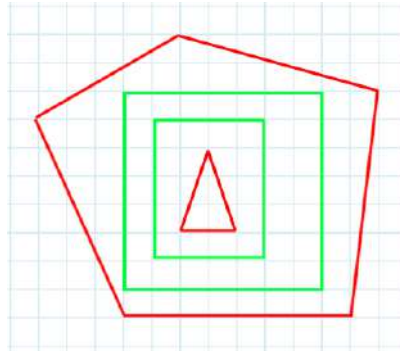
Пример:

Входные данные	Выходные данные
2 1 1 2	M
4 1 1 2 1 3 3 4	V3

### Задача 5

Бельчонок и Зайчонок скачали редактор `polygon`, позволяющий рисовать произвольные многоугольники. Бельчонок все многоугольники рисует зеленым цветом, а Зайчонок – красным. Чтобы рисунок получился красивым (по мнению друзей), каждый многоугольник расположен так, что  $K$  многоугольников находятся внутри него, а сам он находится внутри  $N - K - 1$  многоугольников ( $N$  – общее количество многоугольников,  $0 \leq K \leq N - 1$ ). При

этом никакие два многоугольника не пересекаются ни в одной точке. Пример расположения многоугольников приведен на рисунке.



Программа `polygon` сохраняет в текстовом файле информацию о цвете многоугольника и координатах всех его вершин. Порядок вершин может быть любым (зависит от того, с какой вершины начали рисовать многоугольник). Многоугольники сохраняются в файле в том же порядке, в котором их рисовали. Начало координат расположено в верхнем левом углу, ось  $x$  направлена вправо, ось  $y$  – вниз.

Используя данные в файле, который сформировала программа `polygon`, определите, сколько зеленых многоугольников расположено так, что и внутри них и снаружи них находится хотя бы один многоугольник красного цвета. Для приведенного выше рисунка это количество равно двум.

#### Входные данные

Первая строка входного текстового файла содержит целое число  $N$  – общее количество многоугольников. Далее следуют  $N$  строк ( $N \leq 10000$ ), в каждой из которых сначала записан символ `R` или `G`, обозначающий цвет многоугольника, а затем через пробел несколько (в зависимости от количества углов многоугольника) пар  $(x, y)$  целых неотрицательных чисел – координаты всех вершин многоугольника.

#### Выходные данные

Программа должна выводить единственное число - сколько зеленых многоугольников расположено так, что и внутри них и снаружи них находится хотя бы один многоугольник красного цвета.

Пример (соответствует рисунку выше):

Входные данные	Выходные данные
<pre>4 R 6 8 7 5 8 8 G 11 3 11 10 4 10 4 3 R 13 3 6 1 1 4 4 11 12 11 G 9 9 5 9 5 4 9 4</pre>	<pre>2</pre>

## Информатика. 11 класс

4 вариант

*Работа рассчитана на 240 минут.*

***Все решения должны быть полными и обоснованными.***

### ОБРАТИТЕ ВНИМАНИЕ!

В задачах 1-2 необходимо привести полное обоснованное решение на бланках олимпиады. Для решения данных задач нельзя использовать языки программирования. Решение необходимо оформить на бланке Олимпиады

Для задачи 3 в бланк заносится только ответ. По ссылке необходимо прикрепить файл, при помощи которого вы получили ответ.

Для задач 4-5 в бланк заносится только полученный ответ. По ссылке необходимо прикрепить файлы с программой, с помощью которой вы получили данные ответы

Исходные файлы для заданий находятся в общем хранилище для каждого класса, содержащем папки, названия которых соответствуют номеру задания (например, папка task3 содержит данные для задания 3). В каждой папке даны исходные файлы, именованные по принципу `test_номер класса_номер варианта_номер задания_номер исходного файла.txt`

Например, файлы `test_11klass_v1_4_1.txt`, `test_11klass_v1_4_2.txt`, `test_11klass_v1_4_3.txt` содержат исходные данные для задачи 4 первого варианта 11-го класса. Участник олимпиады должен «пропустить» через программу все 3 исходных файла.

Исходные файлы для выполнения заданий:

<https://www.dropbox.com/sh/6mzgwvmltbpktys/AAB7RQ3V6k1Qqdg5gH3MTjZpa?dl=0>

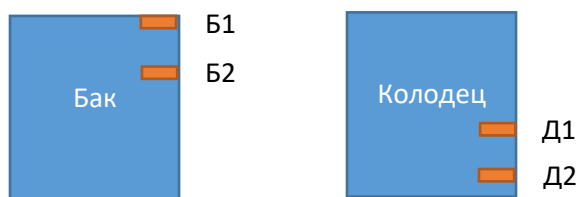
### **Задача 1.**

У Медведя есть колодец с электрическим насосом. Насос накачивает воду в бак, из которого Медведь потом поливает огород. Проблема в том, что в этом году засуха и колодец не очень быстро наполняется водой. Поэтому, чтобы наполнить бак, Медведю приходится по несколько раз включать и выключать насос, так как если насос выкачает всю воду из колодца и будет работать «вхолостую», то он сломается.

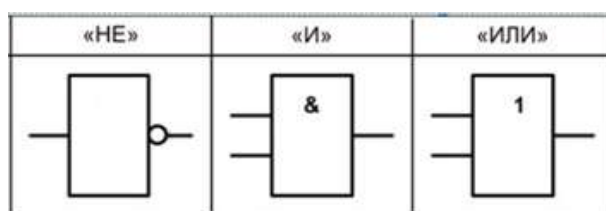
Бельчонок, увлекающийся электроникой, решил придумать схему, состоящую из логических элементов, двух датчиков уровня воды, размещенных в колодце и двух датчиков, размещенных в баке, которая позволила бы автоматически накачивать воду в бак по следующему алгоритму.

Как только уровень воды в баке понижается ниже датчика Б2, запускается процесс накачивания воды, который останавливается при достижении полного бака (датчик Б1). Процесс накачивания состоит в том, что, чтобы насос не работал вхолостую при снижении уровня воды в колодце. При уровне воды ниже датчика Д2 насос отключается и включается только тогда, когда вода в колодце наберется, то есть достигнет датчика Д1.





Разработанная схема должна иметь четыре входа для подключения датчиков и один выход для подключения насоса. Датчики воды выдают сигнал логической «1», если находятся в воде, и логического «0» в противном случае. Насос включается, если на выходе для насоса появляется логическая «1». В логической схеме можно использовать стандартные элементы И, ИЛИ, НЕ. Их условные обозначения, которые нужно использовать на схеме, приведены ниже.



## Задача 2.

Бельчонок построил Робота, который передвигается по тропинке, разделенной на ячейки, в каждой из которых может лежать либо шишка (Ш), либо ничего (Н). Робот передвигается по тропинке, переходя в соседнюю ячейку вправо или влево. При этом он может оставлять в ячейке шишку либо забирать её. В каждой ячейке может лежать только одна шишка. У Робота есть неограниченный запас шишек.

Программа для Робота представляет собой пронумерованную последовательность команд, которые могут быть следующими:

Н N	Забрать из текущей ячейки шишку, если она там лежит, и перейти к выполнению команды, записанной в строке с номером N. Если до выполнения данной команды в ячейке пусто, то она остается пустой.
Ш N	Положить в текущую ячейку шишку и перейти к выполнению команды, записанной в строке с номером N. Если ячейке в уже лежит шишка, то она остается, вторая шишка не добавляется.
> N	Робот перемещается в ячейку справа и переходит к выполнению команды, записанной в строке с номером N
< N	Робот перемещается в ячейку слева и переходит к выполнению команды, записанной в строке с номером N

? NH, NШ	Робот проверяет, что находится в текущей ячейке, и переходит к выполнению команды, записанной в строке с номером NH, если в текущей ячейке ничего нет, или к строке с номером NШ, если в текущей ячейке лежит шишка. При этом содержимое текущей ячейки не меняется.
----------	--

В начальный момент Робот расположен левее «кучи» шишек. Пустых ячеек в «куче» нет. Он должен, следуя программе, разделить данную кучу на две равных по количеству. Если в куче было нечетное количество шишек, то в левой куче должно остаться на 1 больше. Кучи должны быть разделены одной пустой ячейкой. Расположение новых куч относительно исходной кучи может быть любым.

Пример начальных данных (Р – позиция Робота)

	Р			Ш	Ш	Ш	Ш	Ш											
--	---	--	--	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--

Результат работы (положение Робота после завершения программы не принципиально)

				Ш	Ш	Ш		Ш	Ш		Р								
--	--	--	--	---	---	---	--	---	---	--	---	--	--	--	--	--	--	--	--

Изложите сначала алгоритм на естественном языке, а затем напишите программу для Робота, следуя которой он выполнит задание.

### Задача 3.

У Бельчонка есть робот, который перемещается по прямоугольному полю. Робот начинает движение из верхней левой ячейки и заканчивает в правой нижней. Он может перемещаться вправо, вниз и по диагонали вправо-вниз. При этом два раза подряд по диагонали Робот передвигаться не может. У Робота есть счёт, который при посещении ячейки изменяется на записанную в ней величину. В начале счёт равен значению верхней левой ячейки.

Задача Бельчонка – провести Робота из верхней левой ячейки в правую нижнюю так, чтобы счёт Робота в финишной ячейке был наибольшим. В качестве ответа нужно указать максимальное значение счёта Робота в финишной ячейке. Для решения данной задачи можно использовать только средства электронных таблиц. Ответ требуется вписать в бланк и приложить файл с формулами и расчетами.

#### Задача 4.

Бельчонок и Зайчонок играют «в слова». Бельчонок всегда начинает первым. Он пишет слово, затем Зайчонок пишет следующее слово, отличающееся от слова Бельчонка только на 1 букву. При этом разрешается заменить одну букву, добавить одну букву в любое место слова или удалить одну букву. При этом все слова берутся из словаря, придумывать свои слова нельзя. Также нельзя повторять слова, которые уже были написаны в ходе текущей игры. Так как словарь ограничен, то рано или поздно у очередного игрока не будет возможности написать слово. В этом случае игра заканчивается и тот игрок, кто не смог написать очередное слово, признается проигравшим.

Напишите программу, которая, используя словарь возможных слов, по заданному начальному слову (тому, которое написал Бельчонок) определяет, кто из игроков имеет выигрышную стратегию и, если выигрывает Зайчонок, выводит, какое первое слово ему нужно написать, чтобы выиграть. Если таких слов несколько, то следует указать первое по алфавиту (сортировка в лексикографическом порядке)

Например, дан словарь

НОРД

КОРД

НОРА

КОРД

НОД

ДОД

НОР

и начальное слово «НОРА». В этом случае при любом ходе Зайчонка (он может написать «КОРА» или «НОРД») Бельчонок выигрывает, написав слово «КОРД», после которого Зайчонок не сможет продолжить игру. В этом случае программа должна вывести В (побеждает Бельчонок).

Если игра начинается со слова «НОД», то Зайчонку следует написать слово «ДОД». Тогда Бельчонок не сможет продолжить игру и Зайчонок победит. В этом случае программа должна вывести Z ДОД (побеждает Зайчонок, ему следует написать ДОД).

Словарь находится в файле dict.txt и содержит допустимые слова, которые можно использовать при игре.

Напишите программу, использующую словарь и определяющую, кто выиграет при безошибочной игре обоих игроков, если известно, что Бельчонок написал первое слово

- а) НСР
- б) QYAT
- в) NTCC

### Задача 5.

Бельчонок и Зайчонок скачали редактор `lines`, позволяющий рисовать различные отрезки. Бельчонок все отрезки рисует зеленым цветом, а Зайчонок – красным. Известно, что любая пара нарисованных друзьями отрезков имеет или одну точку пересечения или вообще не пересекается. То есть отрезки не накладываются друг на друга.

Программа `lines` сохраняет в текстовом файле информацию о цвете отрезка и координатах его концов. Отрезки сохраняются в файле в том же порядке, в котором их рисовали. Начало координат расположено в верхнем левом углу, ось  $x$  направлена вправо, ось  $y$  – вниз.

Используя данные в файле, который сформировала программа `lines`, определите, какой отрезок имеет больше всего пересечений с отрезками другого цвета. В качестве ответа выведите его координаты. Если таких отрезков несколько, выведите координаты того, который расположен в файле раньше.

#### Входные данные

Первая строка входного текстового файла содержит целое число  $N$  – общее количество отрезков. Далее следуют  $N$  строк ( $N \leq 1000$ ), в каждой из которых сначала записан символ `R` или `G`, обозначающий цвет отрезка, а затем через пробел две пары  $(x, y)$  целых неотрицательных чисел – координаты начала и конца отрезка.

#### Выходные данные

Программа должна выводить координаты концов отрезка, удовлетворяющего условиям задачи.

Пример:

Входные данные	Выходные данные
----------------	-----------------

5 R 1 9 5 1 G 4 7 8 2 R 6 2 6 8 G 1 5 3 8 G 2 2 9 9	6 2 6 8
--	---------

Рисунок, соответствующий примеру:

