

Муниципальный этап олимпиады по информатике (7-8 класс)

24 ноября 2023

1. Игра в кубики

Алиса и Боб играют, бросая две игральные кости (два кубика). Выигрывает тот, кто сможет выбросить «пару» — одинаковое количество очков на обоих кубиках. Если оба игрока выбрасывают «пару» или у обоих нет «пары», то побеждает игрок, выбросивший большее количество очков в сумме, а при равенстве количества очков считается, что игра закончилась вничью.

Определите, кто выиграл в данной игре по количеству очков на каждом кубике.

Ввод содержит четыре целых числа A_1, A_2, B_1, B_2 ($1 \leq A_1, A_2, B_1, B_2 \leq 6$), по одному числу в строке. Первые два числа A_1, A_2 — количество очков на кубиках, выброшенных Алисой, следующие два числа B_1, B_2 — количество очков на кубиках, выброшенных Бобом.

Вывести `Alice`, если победила Алиса, вывести `Bob`, если победил Боб, а в случае ничьей вывести `Tie`.

Пример ввода 1	Пример вывода 1
2 2 4 5	Alice
Пример ввода 2	Пример вывода 2
2 3 4 1	Tie

Система оценки и описание подзадач

Подзадача 1 (30 баллов)

$A_1 \neq A_2, B_1 \neq B_2$ (у обоих игроков нет «пары»)

В этой подзадаче 3 теста, каждый тест оценивается в 10 баллов. Баллы за каждый тест начисляются независимо.

Подзадача 2 (30 баллов)

$A_1 = A_2, B_1 = B_2$ (у обоих игроков выпала «пара»)

В этой подзадаче 3 теста, каждый тест оценивается в 10 баллов. Баллы за каждый тест начисляются независимо.

Подзадача 3 (40 баллов)

$1 \leq A_1, A_2, B_1, B_2 \leq 6$ (только у одного из игроков выпала «пара»)

Необходимые подзадачи: 1, 2.

В этой подзадаче 8 тестов, каждый тест оценивается в 5 баллов. Баллы за каждый тест начисляются независимо.

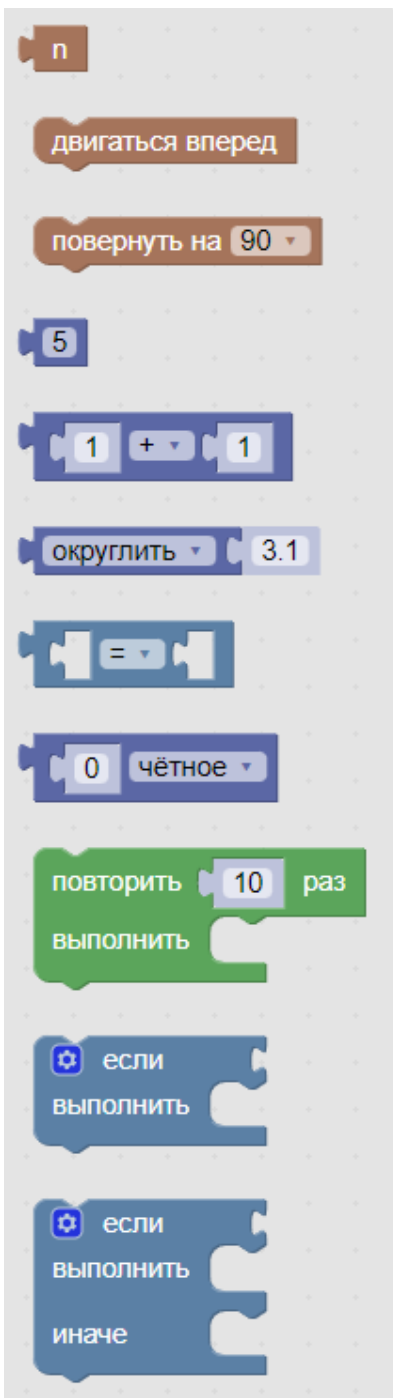
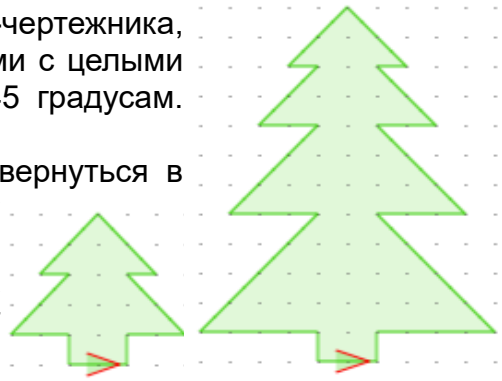
По запросу сообщается результат окончательной проверки на каждом тесте.

Хотя пример ввода 1 не соответствует ограничениям подзадач 1 и 2, ваше решение должно давать правильный ответ и на этом примере для выполнения дальнейшего тестирования.

2. Ёлочка

Напишите в Blockly программу для робота-чертежника, который может перемещаться только между точками с целыми координатами и поворачивать на углы, кратные 45 градусам. При движении робот проводит линию.

После выполнения программы робот должен вернуться в начальную точку в центре ствола, нарисовав ёлочку высотой n веток. Длина нижней ветки ёлки (расстояние от ствола) должна быть равна n , длина остальных веток ёлки должна уменьшаться на 1 от нижней к верхней. На рисунке показаны ёлочки высотой 2 и 4 ветки.

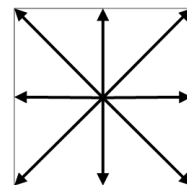


Для написания программы управления роботом вы можете использовать следующие блоки, заданные в категории "Робот":

Первый блок позволяет получить количество веток n (вы можете задавать это значение в соответствующем поле).

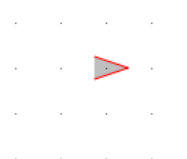
Второй блок указывает роботу двигаться вперед в текущем направлении.

При этом робот перемещается в соседнюю точку с целыми координатами, как показано на рисунке.

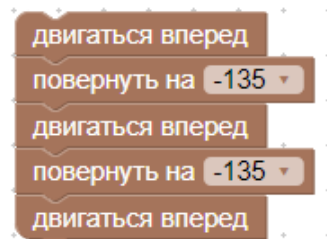


Третий блок указывает роботу изменить направление на указанный угол по часовой стрелке (для положительных значений) или против часовой стрелки (для отрицательных значений).

Первоначально робот направлен направо:



Например, чтобы нарисовать треугольник, нужно выполнить следующие команды:



Следующие пять блоков позволяют выполнять арифметические вычисления, проверять четность и сравнивать значения между собой.

Девятый блок дает возможность повторить выполнение некоторой группы блоков. Например, для рисования восьмиугольника можно выполнить программу:



Последние два блока позволяют выбрать выполняемую группу блоков в зависимости от условия. В настройках блока (символ шестеренки) можно добавить или убрать дополнительные ветки выбора. После завершения настройки нужно повторно нажать на символ шестерёнки.

Для создания переменной в категории "Переменные" нужно щелкнуть по кнопке "Создать переменную" и ввести её название. После этого в категории "Переменные" появятся блоки для получения и изменения значения переменной.

Для запуска программы используйте кнопки:



Первая кнопка позволяет отправить ваше решение для проверки в проверяющую систему соревнований, вторая кнопка выполняет запуск программы локально, третья — пошаговое выполнение или временная остановка программы, четвертая — завершение выполнения программы, после которой программа будет выполняться сначала.

Две кнопки над полем для рисования позволяют менять масштаб изображения как во время выполнения программы, так и после.

Система оценки и описание подзадач

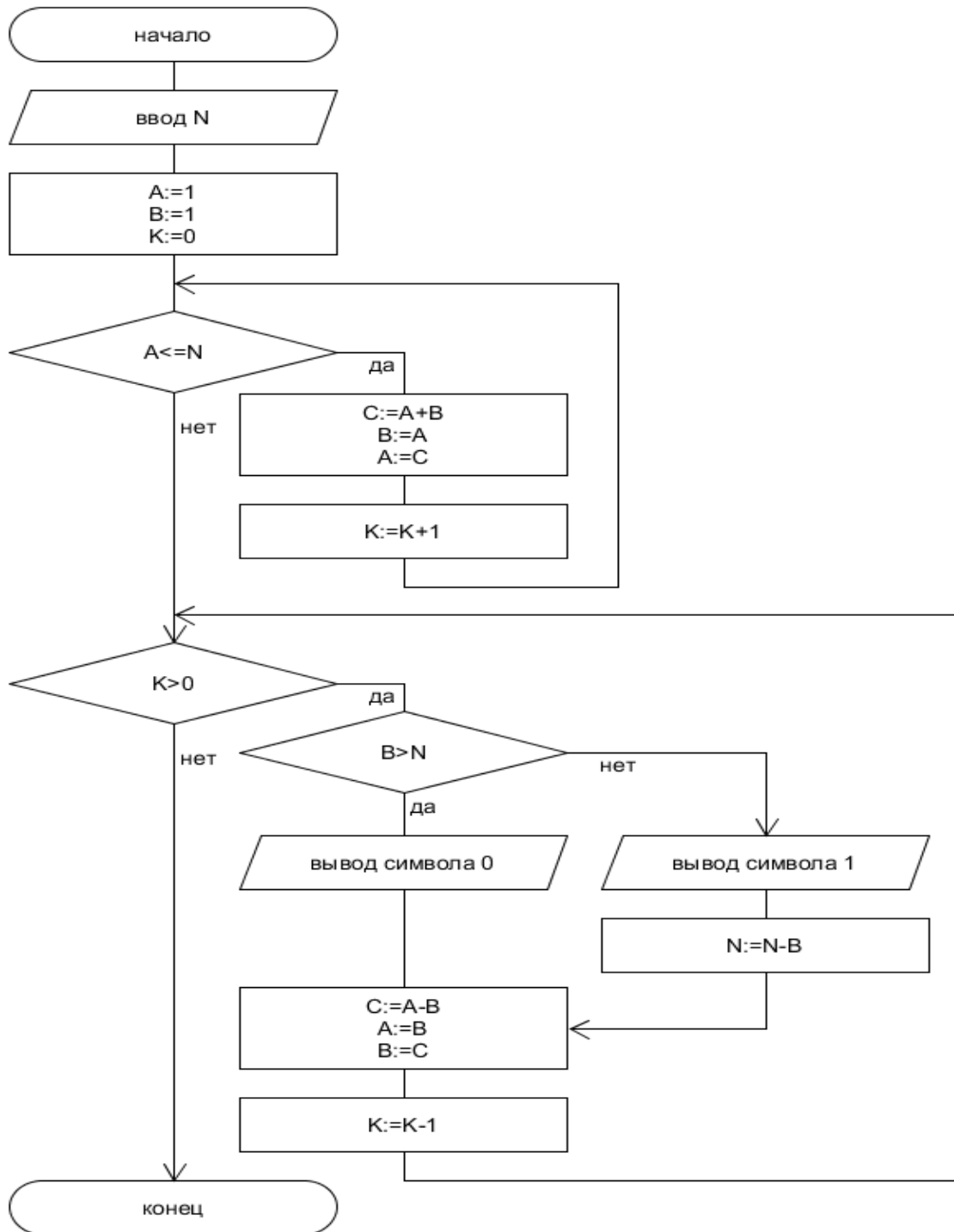
В этой задаче 10 тестов для n от 1 до 10, каждый тест оценивается в 10 баллов. Баллы за каждый тест начисляются независимо.

По запросу сообщается результат окончательной проверки на каждом тесте.

Для написания программы для робота используется специальная версия среды Blockly. Вы можете использовать полную среду Blockly при решении других задач этого соревнования, выбрав пункт Blockly в информационном меню задачи.

3. Алгоритм

Реализуйте на языке программирования алгоритм, представленный на схеме.



Первая строка ввода содержит одно целое число N ($1 \leq N \leq 1000000000$).
Вывести последовательность символов 0 и 1 на одной строке.

Пример ввода 1	Пример вывода 1
21	1000000
Пример ввода 2	Пример вывода 2
123	1010000000

Система оценки

В этой задаче 5 тестов, каждый тест оценивается в 20 баллов. Баллы за каждый тест начисляются независимо.

По запросу сообщается результат окончательной проверки на каждом тесте.

4. Сумма цифр

Вычислите сумму цифр в числах от A до B включительно. Например, сумма цифр в числах от 1 до 14 равна $1+2+3+4+5+6+7+8+9+(1+0)+(1+1)+(1+2)+(1+3)+(1+4)=60$

Ввод содержит два целых числа A и B ($1 \leq A \leq B \leq 10^9$), по одному числу в строке.

Вывод должен содержать одно целое число — сумму цифр в числах от A до B включительно.

Пример ввода	Пример вывода
1 14	60

Система оценки и описание подзадач

Подзадача 1 (20 баллов)

$A=1, 1 \leq B \leq 1000$

В этой подзадаче 4 теста, каждый тест оценивается в 5 баллов. Баллы за каждый тест начисляются независимо.

Подзадача 2 (40 баллов)

$1 \leq A \leq B \leq 10^9, B-A \leq 100000$.

Необходимые подзадачи: 1.

В этой подзадаче 8 тестов, каждый тест оценивается в 5 баллов. Баллы за подзадачу начисляются только в случае, если все тесты для этой подзадачи успешно пройдены.

Подзадача 3 (40 баллов)

$1 \leq A \leq B \leq 10^9$.

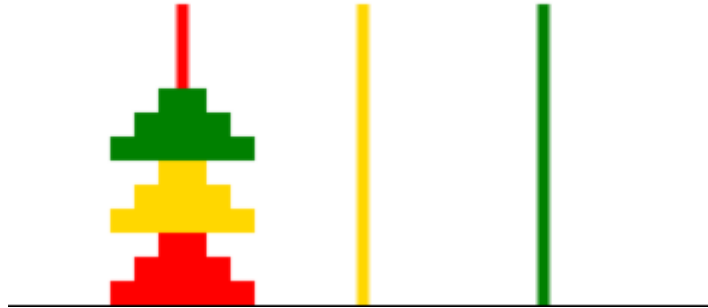
Необходимые подзадачи: 1, 2.

В этой подзадаче 8 тестов. Баллы за подзадачу начисляются только в случае, если все тесты для этой подзадачи успешно пройдены.

По запросу сообщается результат окончательной проверки на каждом тесте в подзадачах 1, и о первой ошибке в подзадачах 2 и 3.

5. Пирамидки

Напишите в Blockly программу для робота, который перемещает разноцветные диски трех различных размеров между тремя колышками. Первый колышек красного цвета, второй — жёлтого, третий — зелёного. Первоначально все девять дисков находятся на первом колышке.



После выполнения программы диски на колышках должны быть распределены по цвету (во всех подзадачах) — на каждом колышке только диски, совпадающие по цвету с цветом колышка — и упорядочены по размеру (только в подзадачах 2 и 4) — на каждом колышке размеры дисков возрастают сверху вниз.

The screenshot shows the following blocks in the Blockly workspace:

- Three 'Move disk from 1 to 2' blocks.
- 'Number of disks on 1' block.
- 'Color of disk on 1' block.
- 'Size of disk on 1' block.
- Two 'if-then' blocks.
- One 'if-then-else' block.
- A '1' block.
- An '=' block with '0' as a value.
- An 'and' block.
- A 'not' block.

Для управления роботом вы можете использовать следующие блоки из категории "Робот".

Первый блок позволяет переместить диск с одного колышка на другой. Если диска на указанном колышке нет, то команда игнорируется.

Второй блок позволяет получить количество дисков на указанном колышке.

Третий и четвёртый блок позволяет узнать цвет и размер верхнего диска на указанном колышке. Цвет возвращается как число от 1 до 3 (1 — красный, 2 — жёлтый, 3 — зелёный), размер как число от 1 до 3. Если на указанном колышке нет ни одного диска, то оба блока возвращают число 0.

Например, для перемещения трёх верхних дисков можно выполнить следующую программу:

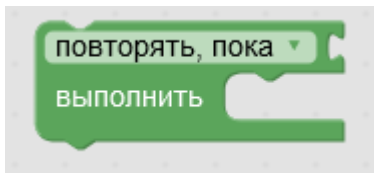
The screenshot shows a sequence of three 'Move disk from 1 to 3' blocks stacked vertically.

Следующие блоки находятся в категории «Управление». Первые два блока позволяют выбрать выполняемую группу блоков в зависимости от условия. В настройках блока (символ шестеренки) можно добавить или убрать дополнительные ветки выбора. После завершения настройки нужно повторно нажать на символ шестерёнки.

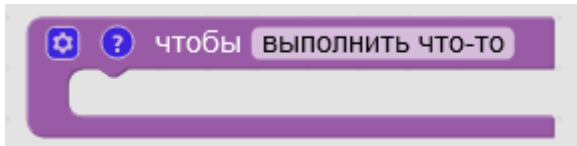
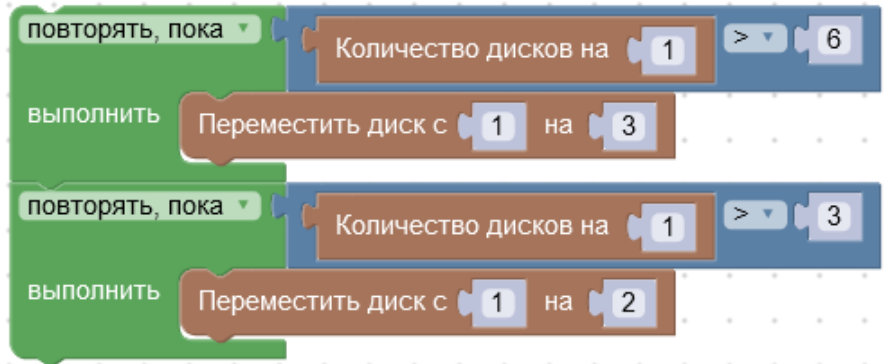
Третий блок позволяет задать число как аргумент для какого-либо блока.

Четвертый блок позволяет сравнить два значения.

Пятый и шестой блоки позволяют задавать сложные условия, используя логические операции И, ИЛИ, НЕ.

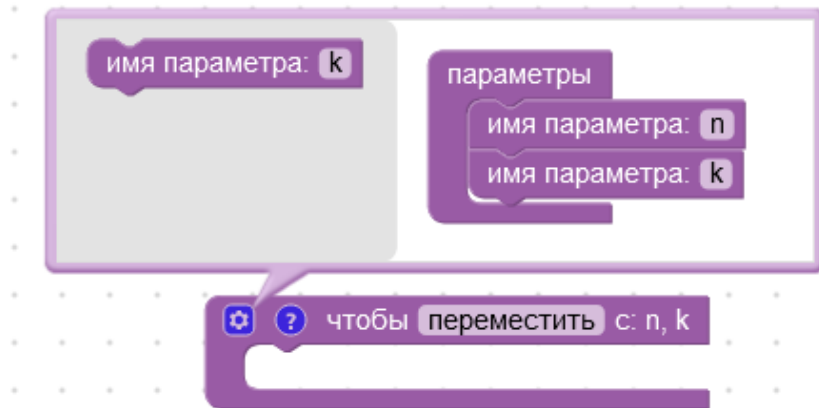


Седьмой блок позволяет повторять некоторую группу блоков, пока условие истинно. Например, для перемещения шести верхних дисков можно написать программу:

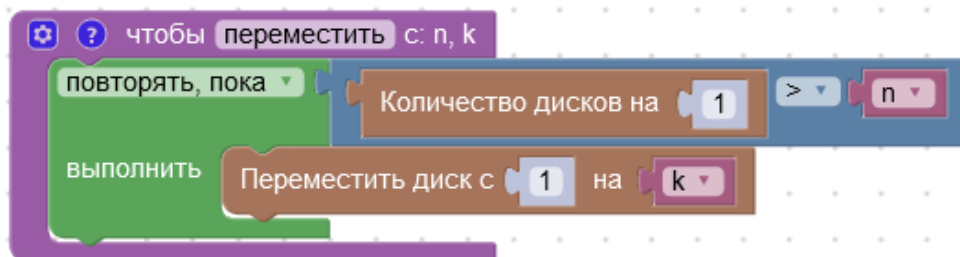


В этой программе две похожих группы блоков, за исключением двух констант. Упростить программу поможет блок из категории «Функции». После вставки блока и изменения имени нужно щелкнуть символ шестеренки и добавить параметры n и k.

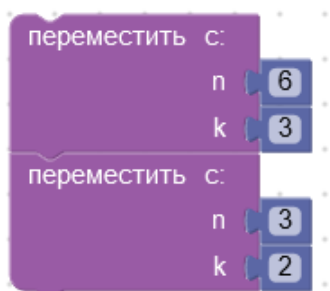
шестеренки и добавить параметры n и k.



После завершения настройки нужно повторно нажать на символ шестерёнки. Добавленные параметры появятся в категории "Переменные" и их можно будет использовать в блоках вместо констант:



Созданная функция появляется в категории «Функция» и её можно добавить в программу, указав значения для параметров с помощью блока-числа из категории «Управление»:

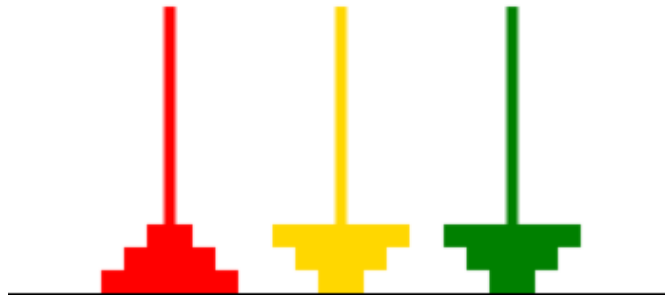


Для запуска программы используйте кнопки:



Первая кнопка позволяет отправить ваше решение для проверки в проверяющую систему соревнований, вторая кнопка выполняет запуск программы локально, третья — пошаговое выполнение или временная остановка программы, четвертая — завершение выполнения программы, после которой программа будет выполняться сначала.

Например, после запуска на выполнение программы, показанной выше, состояние поля изменится на



В поле "Начальное состояние" можно задать начальный порядок дисков. При нажатии левой кнопкой мыши на диск, этот диск переставляется на вершину первого колышка.

Система оценки и описание подзадач

Подзадача 1 (10 баллов)

В этой подзадаче 1 тест, с начальным расположением дисков, показанным на рисунке.



Необходимо разложить диски по цветам — на колышке 1 должны быть диски только цвета 1 (красного), на колышке 2 — только цвета 2 (жёлтого), на колышке 3 — только цвета 3 (зелёного). В каком порядке расположены диски по размерам, не имеет значения.

Подзадача 2 (10 баллов)

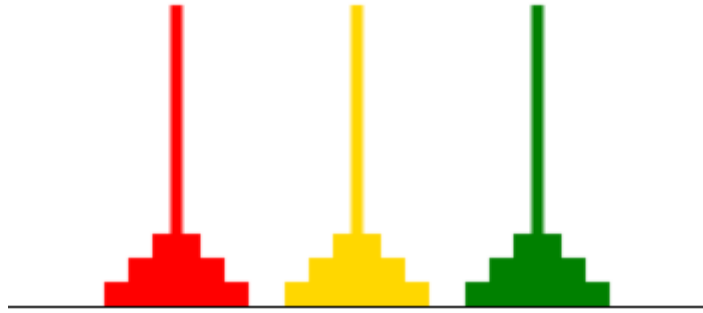
Необходимые подзадачи: 1.

В этой подзадаче 1 тест, показанный на рисунке.



Необходимо разложить диски по цветам и размерам так, чтобы конечное

состояние совпало с расположением на рисунке:



Подзадача 3 (50 баллов)

Необходимые подзадачи: 1.

Девять дисков расположены на первом колышке. Необходимо разложить диски по цветам — на колышке 1 должны быть диски только цвета 1 (красного), на колышке 2 — только цвета 2 (жёлтого), на колышке 3 — только цвета 3 (зелёного). В каком порядке расположены диски по размерам, не имеет значения.

В этой подзадаче 5 тестов, каждый тест оценивается в 10 баллов. Баллы за каждый тест начисляются независимо.

Подзадача 4 (30 баллов)

Необходимые подзадачи: 1,2,3

Девять дисков расположены на первом колышке. Необходимо разложить диски по цветам и размерам (см. рисунок выше).

В этой подзадаче 6 тестов, каждый тест оценивается в 5 баллов. Баллы за каждый тест начисляются независимо.

По запросу сообщается результат окончательной проверки на каждом тесте.