

Strim_6_2

Задача № 1 (4761)

Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду Сместиться на (a, b) (где a, b — целые числа), перемещающую Чертёжника из точки с координатами (x, y) в точку с координатами $(x+a, y+b)$. Если числа a, b положительные, то значение соответствующей координаты увеличивается, если отрицательные — уменьшается. Например, если Чертёжник находится в точке с координатами $(4, 2)$, то команда Сместиться на $(2, -3)$ переместит Чертёжника в точку $(6, -1)$. Запись

Повтори k раз

Команды

конец

означает, что последовательность Команд повторится k раз.

Чертёжнику был дан для исполнения следующий алгоритм:

Повтори 10 раз

Сместиться на $(3, 6)$

Сместиться на $(7, -2)$

Сместиться на $(-10, -4)$

Конец

Перед началом алгоритма Чертёжник находился в точке с координатами $(0, 0)$. Определите, сколько точек с целочисленными координатами будут находиться внутри области, ограниченной линией, заданной данным алгоритмом. Точки на линии учитывать не следует.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=0h2m5s

Решение

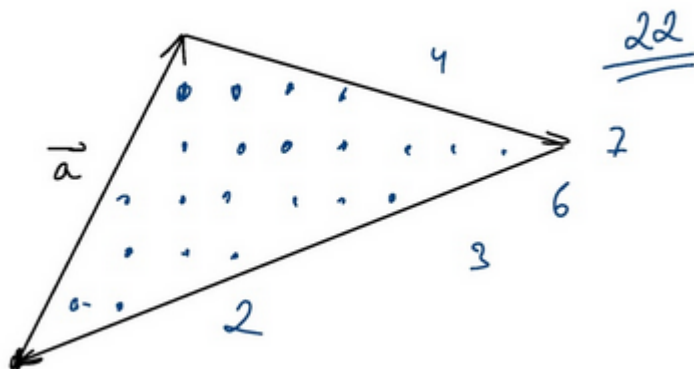
Исполнитель Чертежник движется по координатной плоскости, оставляя за собой линию, похожую на след от черепашки. Однако его движение происходит несколько иначе.

У него всего одна команда – перемещение на вектор (a, b) , где a и b являются целыми числами. Эта команда перемещает "Чертежника" из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$.

Таким образом, находясь в определенной позиции, он сдвигается на указанный вектор (a, b) . Также существует цикл, аналогичный циклу у черепашьего движения. Для выполнения ему был предоставлен следующий алгоритм: повторение последовательности из трех смещений десять раз. В начале алгоритма "Чертежник" находился в точке с координатами $(0, 0)$. Задача состоит в определении количества точек с целочисленными координатами, которые находятся внутри области, ограниченной линией, описанной этим алгоритмом. При этом точки, лежащие непосредственно на границе фигуры, учитываться не должны.

Сначала рассмотрим решение задачи аналитическим методом. Пусть у нас есть исходная точка с координатами $(0, 0)$. Из нее мы сдвигаемся на вектор $(3, 6)$, что означает перемещение на 3 единицы вправо и на 6 единиц вверх. Назовем этот вектор $A: (3, 6)$.

Из новой точки мы перемещаемся на вектор $(7, -2)$, что соответствует движению на 7 единиц вправо и на 2 единицы вниз. Затем выполняется третье смещение на вектор $(-10, 4)$, которое приводит к перемещению на 10 единиц влево и на 4 единицы вверх. В результате этих действий мы возвращаемся в исходную точку $(0, 0)$, образуя замкнутый контур в форме прямоугольника. Теперь, когда мы знаем форму траектории, можно приступить к подсчету точек внутри этого треугольника. Подсчитав их вручную, получаем следующее распределение: 4; 7; затем 6; далее, 3 и, наконец, 2. Важно помнить, что точки, находящиеся прямо на границе контура, не учитываются. Таким образом, общее количество внутренних точек составляет 22 точки.



Ответ: 22

Задача №2 (12915)

Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду Сместиться на (a, b) (где a, b — целые числа), перемещающую Чертёжника из точки с координатами (x, y) в точку с координатами $(x+a, y+b)$. Если числа a, b положительные, то значение соответствующей координаты увеличивается, если отрицательные — уменьшается.

Например, если Чертёжник находится в точке с координатами $(4, 2)$, то команда Сместиться на $(2, -3)$ переместит Чертёжника в точку $(6, -1)$.

Запись Повтори k [Команда1 Команда2 ... КомандаS] означает, что последовательность из S команд повторится k раз.

Чертёжнику был дан для исполнения следующий алгоритм:

Повтори 10 [

- сместиться на $(0, 2)$
- сместиться на $(2, 0)$
- сместиться на $(0, 10)$
- сместиться на $(-2, 0)$
- сместиться на $(0, 2)$
- сместиться на $(6, 0)$
- сместиться на $(0, -2)$
- сместиться на $(-2, 0)$
- сместиться на $(0, -10)$
- сместиться на $(2, 0)$
- сместиться на $(0, -2)$
- сместиться на $(-6, 0)$

]

Перед началом алгоритма Чертёжник находился в точке с координатами (0, 0). Определите, сколько точек с целочисленными координатами будут находиться внутри области, ограниченной линией, заданной данным алгоритмом. Точки на линии следует учитывать.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=0h11m5s

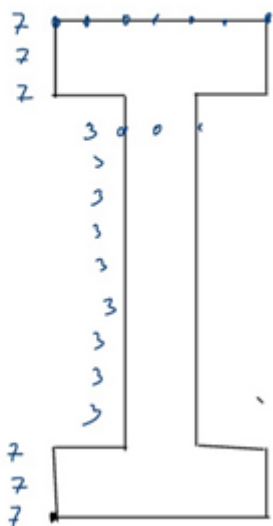
Решение

Есть чертежник, который выполняет определенный алгоритм. Нужно определить количество точек с числовыми координатами, которые находятся внутри области, ограниченной данной линией.

Давайте решим эту задачу аналитическим методом, а затем попробуем написать программу. На самом деле программирование здесь довольно простое; никаких проблем возникнуть не должно. Начнем с определенной точки. Из нее мы двигаемся сначала на две клетки вверх, затем на две клетки вправо и далее на 10 десять клеток вверх.

Итак, продолжаем двигаться вверх на десять клеток. Выглядит как детская головоломка "нарисуй по стрелкам". Далее двигаемся на -2, затем на 2, после чего на 6 и на -2. Далее снова идем вниз на -2. Линия получилась замкнутой, и она повторяется десять раз, образуя определенную фигуру.

$$6 \cdot 7 + 9 \cdot 3 = 42 + 27 = \underline{\underline{69}}$$



Теперь нужно подсчитать количество точек с целыми координатами внутри этой фигуры, включая точки на границе. Посчитаем их.

Всего выходит 69 точек.

Решение с помощью программы.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=0h14m45s

Реализуем решение программно. Очевидно, что поведение этого исполнителя отличается от того, что мы привыкли видеть у черепашки. Там у нас были команды для перемещения вверх-вниз, влево-вправо, вперед-назад. Теперь же

команда будет смещать объект на определенное расстояние. Давайте посмотрим, как создать такую команду.

Используем команду `go to`, которая позволяет переместиться в точку с заданными координатами. Проблема заключается в том, что эта команда рассчитывает координаты относительно начала системы координат $(0, 0)$, а не от текущего положения черепашки. Нам необходимо определить текущее положение черепашки и добавить к нему нужные значения.

Для этого мы можем воспользоваться функциями `xCor` и `yCor`, которые возвращают текущие координаты черепашки. Таким образом, чтобы реализовать команду "смещаться на", мы можем использовать следующую конструкцию:

```
goto(xCor() + a, yCor() + b)
```

Это позволит переместить черепашку из текущей позиции в новую точку с координатами $(x + a, y + b)$.

Теперь рассмотрим, как это будет выглядеть в коде. Начнем с импорта библиотеки `Turtle` и написания функции.

1. Импорт библиотеки `Turtle`:

```
from turtle import *
```

2. Настройка параметров черепашки:

Выключим трассировку движения черепашки (`tracer(0)`), хотя это необязательно. Также установим начальный угол поворота черепашки (`lt 90`), который в нашем случае не играет ключевой роли.

```
tracer(0)
lt(90)
```

3. Установка масштаба и размера экрана:

Укажем размер окна и масштабируем изображение, чтобы оно соответствовало нашим требованиям.

```
screensize()
```

4. Создание функции для смещения:

Напишем функцию, которая будет выполнять команду `go to` с учетом текущих координат черепашки и требуемых смещений. Обратим внимание, что название функции может быть записано кириллицей.

```
def сместиться(a, b):
    goto(xcor() + a * r, ycor() + b * r)
```

5. Использование функции в цикле:

Для демонстрации работы нашей функции создадим цикл, который будет многократно вызывать функцию `сместиться` с различными значениями смещений.

Проведем тестирование и оценим результат. На выходе получаем фигуру, созданную черепашкой. Добавление сетки облегчит определение координат точек.

Так будет выглядеть программа целиком:

```
from turtle import *
tracer(0)
r = 20
screensize(10000,10000)
def сместиться(a,b):
    goto(xcor()+a*r, ycor()+b*r)
for i in range(10):
```

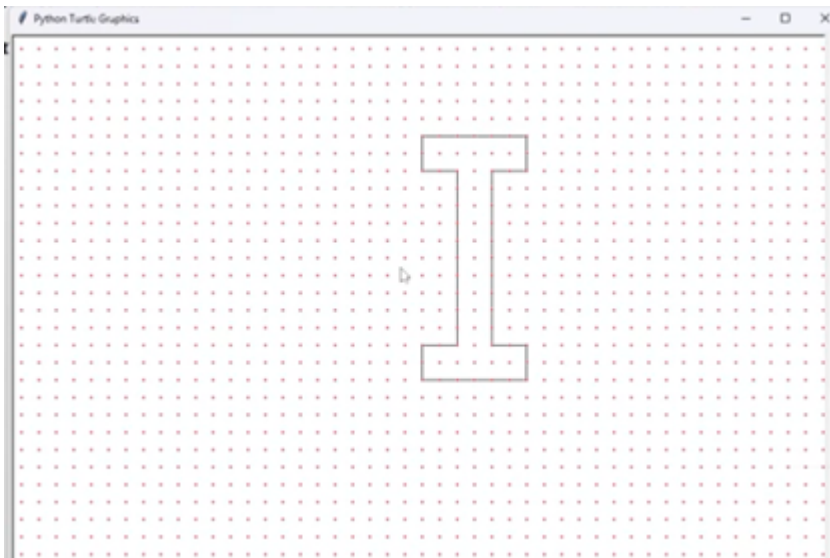
```

    сместиться (0, 2)
    сместиться (2, 0)
    сместиться (0, 10)
    сместиться (-2, 0)
    сместиться (0, 2)
    сместиться (6, 0)
    сместиться (0, -2)
    сместиться (-2, 0)
    сместиться (0, -10)
    сместиться (2, 0)
    сместиться (0, -2)
    сместиться (-6, 0)

up()
for x in range(-50, 50):
    for y in range(-50, 50):
        goto(x*r, y*r)
        dot(3, 'red')
update()

```

Результат работы программы:



Таким образом, мы получили готовую картину с фигурой и целочисленными точками. Подсчёт точек будет осуществляться аналогично подсчёту в аналитическом решении.

Ответ: 69

Задача № 3 (17860)

Исполнитель Черепаха действует на плоскости с декартовой системой координат. В начальный момент Черепаха находится в начале координат, её голова направлена вдоль положительного направления оси ординат, хвост опущен. При опущенном хвосте Черепаха оставляет на поле след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существует 5 команд: Поднять хвост, означающая переход к перемещению без рисования; Опустить хвост, означающая переход в режим рисования; Вперёд n (где n – целое число), вызывающая передвижение Черепахи на n единиц в том направлении, куда указывает её голова; Назад n (где n – целое число), вызывающая передвижение в противоположном голове направлении; Направо m (где m – целое число), вызывающая изменение направления движения

на m градусов по часовой стрелке, Налево m (где m — целое число), вызывающая изменение направления движения на m градусов против часовой стрелки.

Запись Повтори k [Команда1 Команда2 ... Команда S] означает, что последовательность из S команд повторится k раз.

Черепаше был дан для исполнения следующий алгоритм:

Повтори 9 [Вперёд 22 Направо 90 Вперед 6 Направо 90]

Поднять хвост

Вперед 1 Направо 90 Вперёд 5 Налево 90

Опустить хвост

Повтори 9 [Вперёд 53 Направо 90 Вперёд 75 Направо 90]

Определите периметр области пересечения фигур, ограниченных заданными алгоритмом линиями

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=0h23m20s

Решение

Решим эту задачу аналитическим методом. Черепаше необходимо выполнить алгоритм, состоящий из следующих этапов:

1. Повторить девять раз: сделать 22 шага вперед, повернуть направо на 90 градусов, пройти шесть шагов вперед и снова повернуть направо на 90 градусов.
2. Поднять хвост, пройти один шаг вперед, повернуть направо на 90 градусов, пройти пять шагов вперед, повернуть налево на 90 градусов и опустить хвост.

Предположим, что черепаха изначально находится в стартовой позиции, обращенной вверх (в соответствии с условиями задачи). На первом этапе она делает 22 шага вперед, затем поворачивается направо на 90 градусов. Далее она идет еще шесть шагов вперед и вновь поворачивается направо на 90 градусов.

Для визуализации процесса можно представить, как черепаха чертит линию длиной 22 шага, затем поворачивается направо и проходит еще шесть шагов. После второго поворота на 90 градусов она возвращается в исходное положение, но уже обращена вниз.

Данный процесс повторяется девять раз. С каждым циклом черепаха движется по одному и тому же маршруту, возвращаясь в исходную точку, но изменяя

направление взгляда. По завершении последнего цикла она остановится и будет смотреть в сторону.

После этого черепаха поднимает хвост, делает один шаг вперед, поворачивается направо на 90 градусов и проходит пять шагов вперед. Таким образом, она перемещается в новую позицию.

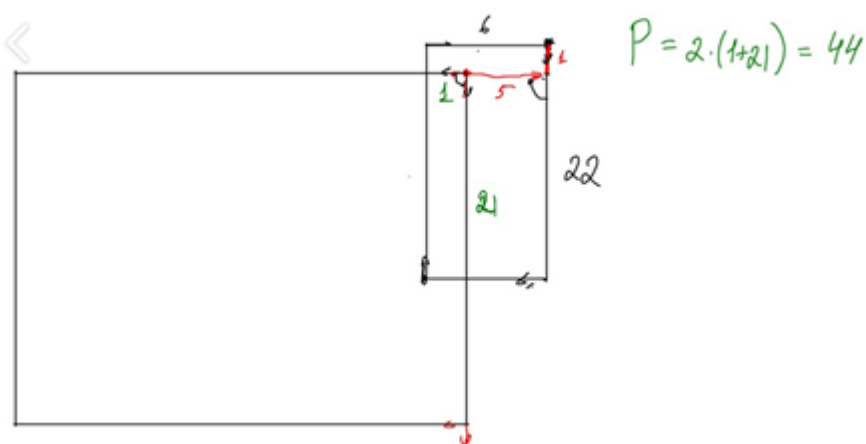
Следуя указанным инструкциям, можно заметить, что черепаха сначала проходит один шаг вперед, затем поворачивается направо и проходит еще пять шагов. В результате она оказывается в новой точке и обращена вниз.

По окончании всех описанных действий черепаха занимает новую позицию: пройдя один шаг вперед и повернувшись направо, она проходит еще пять шагов. В итоге она оказывается в новом месте и обращена вниз.

Когда черепаха опускает хвост, начинается следующий этап ее перемещения. Она проходит 53 шага вперед, поворачивается направо на 90 градусов, затем проходит 75 шагов вперед и снова поворачивается направо на 90 градусов. Таким образом, она рисует больший прямоугольник вокруг меньшего.

В результате образуются две геометрические фигуры: пересекающиеся большой и малый прямоугольник. Задача заключается в определении периметра области их пересечения. Размеры этой области составляют 5 на 21 единиц. Число 21 объясняется тем, что сторона большего прямоугольника равна 22 единицам.

Правильный ответ — 44 единицы.



Ответ: 44

Задача № 4 ()

Исполнитель Цапля действует на плоскости с декартовой системой координат. В начальный момент Цапля находится в начале координат, её клюв направлен вдоль положительного направления оси ординат, клюв опущен. При опущенном клюве Цапля оставляет на поле след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существует три команды: Вперёд n (где n - целое число), вызывающая передвижение Цапли на n единиц в том направлении, куда указывает её клюв; Направо m (где m число), вызывающая изменение направления движения на m градусов по часовой стрелке; Дуга r, a, b (где r, a, b - целые число), вызывающая передвижение Цапли из текущей точки с координатами (x, y) по дуге полуокружности с центром в точке с координатами $(x + a, y + b)$ и радиусом r , движение по полуокружности идёт по часовой стрелке.

Запись Повтори k [Команда1 Команда2 ... КомандаS] означает, что последовательность из S команд повторится k раз.

Цапле был дан для исполнения следующий алгоритм:

Повтори 7 [Дуга 4, 0, 4 Дуга 4, 4, 0 Дуга 4, 0, -4 Дуга 4, -4, 0].

Определите, сколько точек с целочисленными координатами будут находиться внутри области, ограниченной линией, заданной данным алгоритмом. Точки на линии учитывать не следует.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=0h35m30s

Решение

Суть задания заключается в том, что у нас есть исполнитель под названием Цапля, который во многом напоминает другого исполнителя – Черепаху. Однако у Цапли имеется уникальная команда, называемая Дуга.

Как и у Черепахи, у Цапли есть команды для движения вперёд и поворота вправо, а также возможность указания направления. Команда "Дуга r " перемещает Цаплю из её текущего положения (координаты x, y) вдоль полуокружности. Полуокружность строится с центром в точке $(x+b, y+r)$ и радиусом r . Перемещение происходит по часовой стрелке.

Рассмотрим, как именно следует выполнять команду Дуга". Полуокружность начинается в точке с координатами X, Y и имеет центр в точке $X+B, Y+R$, радиус r . Движение осуществляется по часовой стрелке.

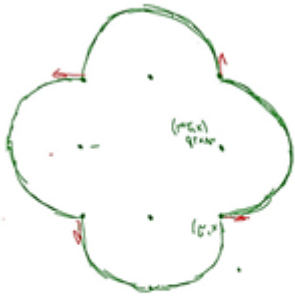
Таким образом команда Дуга 4, 0, 4 указывает на создание дуги радиусом 4, где центр окружности смещён на 0 клеток вправо и на 4 клетки вверх относительно исходной позиции. Смещение составляет 0,4 от текущей точки.

Мы должны нарисовать полуокружность, начиная с этого места, с центром здесь и двигаться по часовой стрелке. Далее идёт вторая дуга. Вторая дуга задаётся параметрами "4, 4, 0", что означает дугу радиуса 4 с центром, смещённым на 4 клетки вправо. Центр новой окружности располагается здесь, и мы снова рисуем полуокружность по часовой стрелке.

Затем третья дуга с параметрами "4, 0, -4", что даёт радиус 4 и смещение центра на 0 клеток вправо и 4 клетки вниз. Центр третьей окружности находится здесь, и опять же, дуга рисуется по часовой стрелке.

Последняя дуга "4,-4,-0" требует смещения центра на 4 клетки влево. В результате мы возвращаемся к начальной точке, и таким образом создаём цветок. Полученное изображение, действительно, выглядит как цветок. Можно ли подсчитать в нем количество точек?

При таком уровне точности сложно получить точный результат, фактически мы можем только представить, как должна выглядеть эта фигура. Поэтому, для получения ответа мы воспользуемся программой.



Полукружность в черепахе
`circle(-радиус, 180)`

Решение задачи программой.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=0h41m20s

Модуль `turtle` предоставляет возможность создания полукруга с помощью функции `circle`. Эта функция позволяет рисовать полную окружность, однако при необходимости можно ограничить её до определённого угла, например, 180 градусов, что соответствует половине круга.

Следует учитывать особенности ориентации рисунка, так как по умолчанию движение происходит против часовой стрелки. Чтобы изменить направление на противоположное, используется отрицательное значение радиуса. Важно также корректно устанавливать начальное положение курсора, так как его ориентация влияет на дальнейшее выполнение команд.

Подключив модуль `turtle`, приступим к выполнению задачи.

Сначала зададим значение масштаба, `r`, равным 30. После этого установим размер экрана — 10000 на 10000 пикселей. Развернем черепаху влево таким образом, чтобы её направление было направлено вверх.

Чтобы первая дуга отобразилась корректно, нужно, чтобы черепаха была направлена влево. Для этого выполним поворот на 90 градусов и нарисует окружность с радиусом 4 и углом 180 градусов.

Затем снова повернем черепаху еще на 90 градусов, чтобы вернуть ее в вертикальное положение.

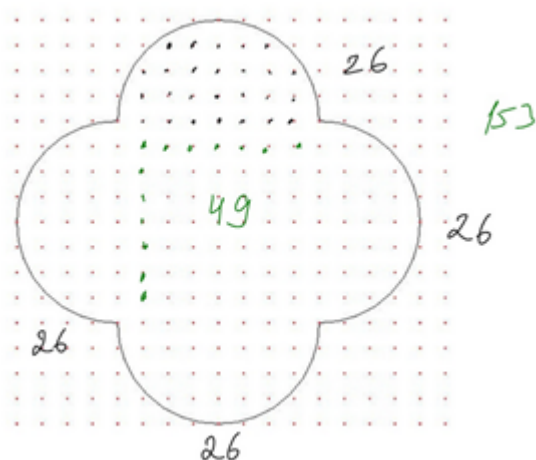
Теперь продолжим рисование полукружий. Этот процесс будет повторяться дважды, после чего следует выполнить поворот вправо. Сделав поворот на 90 градусов влево, а затем еще на 90 градусов, получим третью полукружность. Затем выполнив команду `up()`, поднять хвост, рисуем координатную сетку. Завершаем программу командой `update()`

```
from turtle import *
tracer(0)
lt(90)
screensize(10000,10000)
r = 30
lt(90)
circle(-4*r, 180)
lt(90)
circle(-4*r, 180)
lt(90)
circle(-4*r, 180)
lt(90)
circle(-4*r, 180)
up()
for x in range(-50,50):
```

```
for y in range(-50, 50):
    goto(x*r, y*r)
    dot(3, 'red')
```

```
update()
```

Результат работы программы:



Таким образом, у нас получилась исходная форма фигуры, соответствующая заданным параметрам. Подсчёт количества точек показывает, что каждая из четырёх частей содержит 26 точек, а центральная часть — 49 точек. Общее количество точек равно 153.

Ответ: 153

Задание №5 ()

Исполнитель Цапля действует на плоскости с декартовой системой координат. В начальный момент Цапля находится в начале координат, её клюв направлен вдоль положительного направления оси ординат, клюв опущен. При опущенном клюве Цапля оставляет на поле след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существует три команды: Вперёд n (где n - целое число), вызывающая передвижение Цапли на n единиц в том направлении, куда указывает её клюв; Направо m (где m число), вызывающая изменение направления движения на m градусов по часовой стрелке; Дуга r, a, b (где r, a, b - целые число), вызывающая передвижение Цапли из текущей точки с координатами (x, y) по дуге полуокружности с центром в точке с координатами $(x + a, y + b)$ и радиусом r , движение по полуокружности идёт по часовой стрелке.

Запись Повтори k [Команда1 Команда2 ... КомандаS] означает, что последовательность из S команд повторится k раз.

Цапле был дан для исполнения следующий алгоритм:

Направо 180 Вперёд 3 Направо 90 Вперёд 48 Направо 90 Вперёд 3

Повтори 6 [Дуга 4, 4, 0].

Определите, сколько точек с целочисленными координатами будут находиться внутри области, ограниченной линией, заданной данным алгоритмом. Точки на линии учитывать не следует

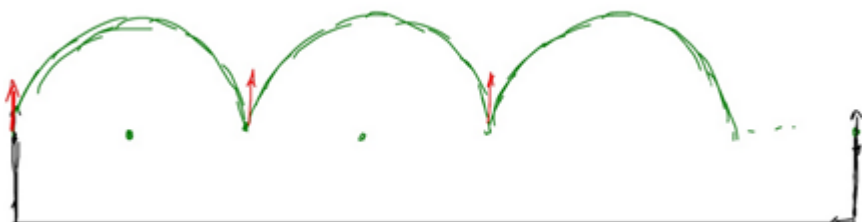
Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=0h50m0s

Решение

Используем для решения задачи Черепаху. Сначала она поворачивается направо на 180 градусов, то есть оказывается внизу. После этого она движется вперед на три клетки. Затем она делает поворот направо на 90 градусов и проходит вперед еще 48 клеток. Это достаточно большое расстояние, поэтому используем примерный масштаб.

Дальше она опять поворачивает направо на 90 градусов и идет вперед еще на 3 клетки. В итоге наш маршрут выглядит так: вниз, вперед на 3, направо на 90, вперед на 48, направо на 90 и вперед на 3. Теперь необходимо повторить операцию Дуга 4, 4, 0 шесть раз. Дуга 4 4 0 – это дуга с радиусом 4, начинающаяся от точки, расположенной на расстоянии 4 клетки вправо от центра. Далее та же самая дуга повторяется еще пять раз.

Итак, всего таких дуг должно быть шесть. Если присмотреться, видно, что диаметр составляет 8, а ширина фигуры равна 6 на 8, что дает 48. Следовательно, наша фигура точно замкнется.

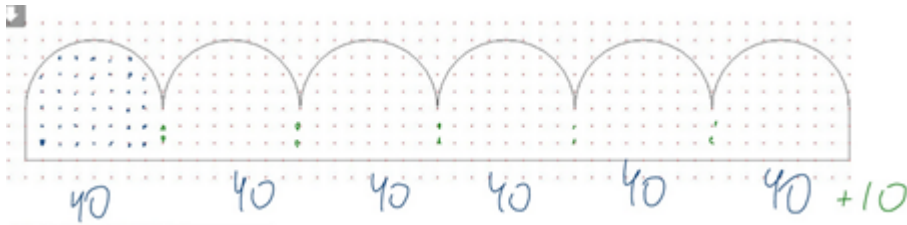


Построим это изображение в Python:

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=0h52m50s

```
from turtle import *
tracer(0)
lt(90)
screensize(10000,10000)
r = 30
rt(180)
fd(3*r)
rt(90)
fd(48*r)
rt(90)
fd(3*r)
for i in range(6):
    circle(-4*r, 180)
    rt(180)
up()
for x in range(-50,50):
    for y in range(-50,50):
        goto(x*r,y*r)
        dot(3, 'red')
update()
```

Результат работы программы:



В одном ряду 40 точек. Всего рядов 6, значит, получаем 240 точек. Кроме того, есть ещё точки посередине, их 10. Добавляем эти 10 точек к общему количеству, и получаем итоговое число – 250 точек.

Ответ: 250

Задание № 6(13837)

Исполнитель Кузнечик существует на числовой прямой. Кузнечик имеет три команды:

Вперед 3 – перемещает кузнечика на 3 единицы вперед,

Назад 4 – перемещает кузнечика на 4 единицы назад,

Перекрасить – закрашивает не закрашенную клетку или отменяет закрашку закрашенной клетки, которая находится в точке, где в данный момент находится исполнитель.

Запись Повтори k раз Команды конец означает, что последовательность Команд повторится k раз.

Определите, сколько клеток будет закрашено после выполнения следующего алгоритма

Повторить 7 раз

 Повторить 2 раз

 Вперед 3

 Конец

 Перекрасить

 Назад 4

 Перекрасить

Конец

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=0h59m10s

Решение

Кузнечик перемещается вдоль числовой прямой. Он может выполнять три различные команды: перемещение вперёд на три единицы, возврат назад на четыре единицы и перекраска текущей клетки. Команда "перекрасить" изменяет состояние клетки — если клетка была пустой, она становится закрашенной, а если уже была закрашена, то краска удаляется (т.е., происходит смена состояния от нуля к единице и наоборот).

Рассмотрим теперь конкретный алгоритм. Наша цель — выяснить, сколько клеток останется закрашенными после того, как мы выполним семь повторений следующей последовательности команд.

Представим себе числовую прямую, которая состоит из отдельных клеток. Допустим, что

"Кузнечик" начинает свое движение с какой-то конкретной точки.

Каковы его действия? Во-первых, он движется вперёд на три клетки. Затем он снова продвигается вперёд на три клетки. Далее он выполняет команду "перекрасить", меняя цвет текущей клетки, и затем возвращается назад на четыре клетки, при этом также перекрашивая эту клетку.

Таким образом, во время первого цикла "Кузнечик" проходит три клетки вперёд, потом еще три клетки вперёд, а затем возвращается на четыре клетки назад. В результате его конечная позиция

оказывается сдвинутой на две клетки вправо относительно стартовой точки.

На втором цикле "Кузнечик" повторяет ту же последовательность шагов, но уже начиная с нового положения. Поэтому после каждого полного цикла его позиция сдвигается на две клетки вправо. Это означает, что он каждый раз перекрашивает новые клетки.

Проанализировав все семь циклов, можно увидеть, что некоторые клетки перекрашивались более одного раза. Например, одна из клеток могла быть сначала окрашена, а затем снова очищена. В конечном счёте останутся лишь те клетки, которые были закрашены последним.

В итоге остается всего 4 закрашенные клетки. Все остальные клетки либо остались незакрашенными, либо были перекрашены дважды (окрашены, а затем очищены).



Решение задачи программой

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=1h7m50s

1. Создадим массив из 100 элементов, каждый из которых инициализируем значением 0.
2. Переменная k будет хранить текущую позицию кузнечика.
3. В цикле выполняем следующие шаги:
 - Двигаемся вперёд на 3 единицы дважды.
 - Меняем значение текущей клетки (если элемент равен 0, ставим 1, иначе – 0).
 - Возвращаемся назад на 4 единицы.
 - Повторяем эти операции 7 раз.
4. После завершения цикла выводим количество клеток, которые были перекрашены в 1.

```
a = [0]*100
k = 0
for i in range(7):
    k += 3
    k += 3
    a[k] = 0 if a[k]==1 else 1
    k -= 4
    a[k] = 0 if a[k]==1 else 1
print(a.count(1))
```

Результат работы программы:

4

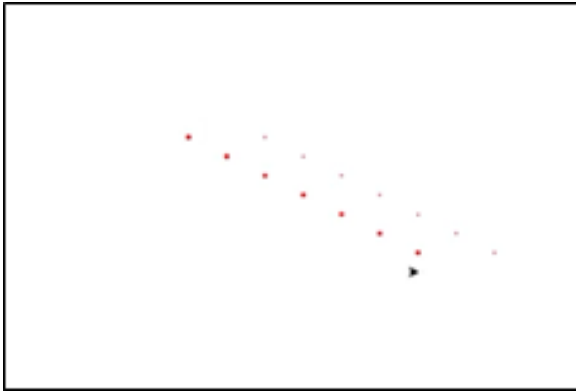
Решение задачи программой II:

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=1h14m50s

Идея использовать действия исполнителя не на одной, а на нескольких (двух) прямых.

```
from turtle import *
r = 20
up()
for i in range(7):
    fd(3*r)
    fd(3*r)
    dot(3, 'red')
    bk(4*r)
    dot(5, 'red')
    #опустим Черепаху на 1 вниз
    rt(90)
    fd(1*r)
    lt(90)
```

Результат работы программы:



Ответ: 4

Задание №7 (13838)

Исполнитель Вертун существует на поле, разделенном на одинаковые квадратные клетки.

Вертун имеет четыре команды:

- Вперед x – перемещает Вертуна на x клеток вперед,
- Вправо – поворачивает Вертуна на 90 градусов вправо,
- Влево – поворачивает Вертуна на 90 градусов влево,
- Закрасить – закрашивает клетку, в которой находится Вертун.

Запись Повтори k раз Команды конец означает, что последовательность Команд повторится k раз.

Определите, сколько клеток будет закрашено после выполнения следующего алгоритма

Повторить 10 раз

Вперед 2

Закрасить

Вперед 3

Вправо

Вперед 3

Влево

Вперед 1

Закрасить
Влево
Вперед 3
Влево
Конец

Решение задачи программой

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=1h24m40s

Безусловно для решения этой задачи мы также можем использовать исполнителя Черепаха. И, хотя, в «Черепахе» это не квадраты клеток, но можно представить, что целочисленные точки являются этими самыми клетками. То есть, точки с координатами (1,1), (1,2), (1,3) и т.д., которые находятся на пересечении целых чисел, могут быть представлены как клетки. Таким образом, мы можем мысленно нарисовать сетку из таких точек-клеток.

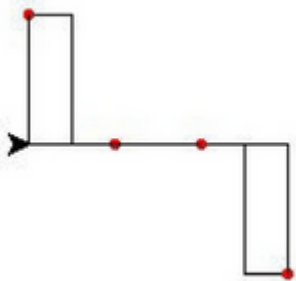
Теперь давайте начнем движение по этим клеткам и закрашивать некоторые из них. В цикле `for i in range(10)` черепаха будет выполнять несколько команд.

- Идти вперед на 3 шага;
- Повернуть направо на 90 градусов;
- Снова идти вперед на 3 шага;
- Повернуть налево на 90 градусов;
- Сделать еще один шаг вперед;
- Закрасить эту клетку;
- Повернуться налево и сделать еще 3 шага вперед.

После выполнения всех десяти циклов получается 4 закрашенных клетки

```
from turtle import *
speed('fastest')
r = 20
for i in range(10):
    fd(2*r)
    dot(5, 'red')
    fd(3*r)
    rt(90)
    fd(3*r)
    lt(90)
    fd(1*r)
    dot(5, 'red')
    lt(90)
    fd(3*r)
    lt(90)
```

Результат работы программы:



Решение

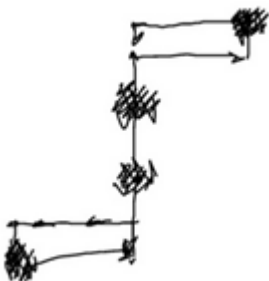
Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=1h28m50s

Начнем с исходной позиции. Вперед на два шага вперед и закрасить. Продолжить движение вперед еще на два шага и повторить окрашивание. Затем переместиться на три шага вперед. Повернуться направо. Сделать три шага вперед.

Повернуть налево. При повороте налево фактически происходит подъем. Переместиться на один шаг вперед, добавить краску. Продолжить движение вперед. Повернуть налево. Повторить поворот налево. Продолжать двигаться в этом направлении. Сделать три шага вперед. Опять повернуть налево.

С данной точки двигаемся вперед на два шага. Окрашиваем их. Затем продолжаем движение вперед на три шага. Один, два, три. Поворот налево. Сделать шаг вперед. Сделать три шага. Повернуть направо. И снова три шага вперед.

Один, два, три. Вот так. Повернуть налево. Передвинуться на один шаг вперед и залить его краской. Получилась такая фигура.



Мы также получили 4 закрашенных точки.

Ответ: 4

Задание №8 (13833)

Исполнитель Водолей переливает воду между тремя колбами А, В и С.

Водолей имеет три команды: Наполни А/В/С - наполняет одну из колб, указанную в команде (например, команда Наполни А полностью наполняет колбу А); Вылей А/В/С - опустошает одну из колб, указанную в команде (например, команда Вылей С опустошит колбу С); Перелей из А/В/С в А/В/С - переливает воду из колбы _1, название которой указано после слова "из", в колбу _2, имя которой указано после слова "в". Если колба _2 может поместить весь объем воды из колбы _1, то

переливается вся вода, если нет, только то количество, которое колба_2 может вместить, не поместившийся в колбу_2 объем, остается в колбе_1.

Запись Повтори k раз Команды конец означает, что последовательность Команд повторится k раз.

Колба А имеет объем 4 литра, В – 6 литров, С – 3 литра.

Исполнителю был задан следующий алгоритм:

Повтори 5 раз

Наполни А

Перелей из А в С

Перелей из А в В

Вылей С

Конец

Перелей из В в С

Сколько литров воды останется в колбе В после выполнения этого алгоритма?

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=1h32m0s

Решение

Давайте рассмотрим, что представляет собой Водолей. Это исполнитель, задачей которого является перемещение воды между тремя сосудами различного объема.

У Водолея имеются три основные команды:

- Наполнить (А, В или С) – полностью заполняет указанный сосуд;
- Вылить – опорожняет выбранный сосуд, сливая всю находящуюся в нем воду;
- Перелить – перемещает воду из одного сосуда в другой. Если второй сосуд способен принять весь объем воды, он будет заполнен полностью; иначе вода переливается лишь частично, а остаток остается в первом сосуде. Например, попробуем перелить 3 литра из первого во второй. Поскольку второй сосуд может вместить только 2 литра, в нем окажется 2 литра, а 1 литр останется в первом. Перейдем к решению задачи. Имеются три сосуда: А (объем 4 литра), В (6 литров) и С (3 литра). Нам необходимо узнать, сколько воды останется в сосуде В после выполнения определенного набора команд.

Изначально все сосуды пусты. Первая команда – заполнить А. Теперь в А содержится 4 литра воды. Затем мы переливаем воду из А в С. Так как С может вместить всего 3 литра, 1 литр остается в А. Далее мы переливаем оставшиеся литры из А в В, оставляя В с одним литром воды, а С с тремя. Следующая команда – вылить содержимое С. Таким образом, после первого цикла мы имеем: 0 литров в А, 1 литр в В и 0 литров в С.

На втором цикле мы снова начинаем с заполнения А, затем переливаем 3 литра в С и оставшийся литр в В. После этого мы опять опорожняем С, получая 0 литров в А, 2 литра в В и 0 литров в С. Этот процесс продолжается пять раз, пока в В не накопится 5 литров воды. Последняя команда – перелить воду из В в С. Поскольку С может вместить только 3 литра, после этой операции в В останется 2 литра воды.

Ответ: 2

	A	B	C
1	0	0	0
2	1	0	0
3	1	0	3
4	0	1	3
5	0	1	0

2	0	2	0
3	0	3	0
4	0	4	0
5	0	5	0
	0	2	3

Решение задачи программой

Потребуется выполнение следующих действий:

- Наполнение сосуда;
- Опустошение сосуда;
- Переливание воды из одного сосуда в другой.

Наиболее сложным является операция переливания, поскольку она требует учета текущего объема каждого сосуда и их максимальной вместимости.

Для этого:

Создадим список для хранения объемов жидкостей в каждом сосуде. Индексы списка будут соответствовать следующим сосудам:

- Индекс 0 соответствует сосуду А,
- Индекс 1 соответствует сосуду В,
- Индекс 2 соответствует сосуду С.

Определим максимальную емкость каждого сосуда:

- Для А: 4 литра,
- Для В: 6 литров,
- Для С: 3 литра.

Напишем функции для выполнения основных операций:

- `наполни(i)` — наполняет сосуд с индексом *i* до максимума;
- `вылей(i)` — опустошает сосуд с индексом *i*;
- `перелей(i, j)` — переливает жидкость из сосуда с индексом *i* в сосуд с индексом *j*.

Функция `перелей(i, j)` учитывает текущий объем обоих сосудов и выполняет операцию переливания соответствующим образом.

После написания функций реализуем основной цикл, который выполнит необходимые операции пять раз и выведет итоговый результат.

```
mx = [4, 6, 3]
a = [0, 0, 0]
def наполни(i):
    a[i] = mx[i]
def вылей(i):
    a[i] = 0
def перелей(i, j):
    остатокj = mx[j] - a[j]
    if a[i] <= остатокj:
        a[j] = a[j] + a[i]
        a[i] = 0
    else:
        a[j] = mx[j]
        a[i] = a[i] - остатокj
```

```
for i in range(5):  
    наполни(0)  
    перелей(0,2)  
    перелей(0,1)  
    вылей(2)  
перелей(1,2)  
print(a)
```

Результат работы программы:

[0, 2, 3]

Программа выводит список [0, 2, 3], что означает, что в итоге в сосуде В остается 2 литра воды, как и требовалось по условию задачи.

Ответ: 2

Задание №9 (11862)

Исполнитель Водолей работает с колбами трёх различных объёмов, выраженных в условных единицах. В начальный момент времени колбы пусты. В систему команд исполнителя входит 3 команды: команда наполни X наполняет водой колбу X, команда вылей X опустошает колбу X, команда перелей из X в Y переливает воду из колбы X в колбу Y.

Объём колбы А – 8, объём колбы В – 10, объём колбы С – 7. Перед исполнителем поставлена задача – отмерить 5 условных единиц воды. Известно, что выполнив данный алгоритм, исполнитель справился с задачей:

Наполнить А
Перелить из А в С
Перелить из С в В
Вылить А
Наполнить А
Перелить из А в В

Какое суммарное количество воды было в колбах по завершении алгоритма?

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=1h37m45s

Решение

Давайте рассмотрим задачу более подробно.

У нас есть три колбы: А, В и С. Их объёмы составляют соответственно 8, 10 и 7 единиц (например, литров). Исполнителю необходимо отмерить ровно 5 единиц воды. Для этого ему нужно следовать определенному алгоритму действий с этими колбами.

Сначала давайте уточним начальные условия:

- Колба А изначально пуста, её объем составляет 8 единиц.
- Колба В также пуста, ее объем – 10 единиц.
- Колба С тоже пуста, она может вместить до 7 единиц.

Теперь рассмотрим пошагово действия исполнителя:

1. Шаг 1: Колбу А заполняем полностью водой. Теперь в ней находится 8 единиц воды.
о $A = 8, B = 0, C = 0$
2. Шаг 2: Переливаем воду из колбы А в колбу С. Так как С может вместить только 7 единиц, туда попадает 7 единиц воды, а одна единица остаётся в А.
о $A = 1, B = 0, C = 7$
3. Шаг 3: Затем всю воду из С переливаем в В. Это возможно, поскольку В может вместить больше, чем содержится в С.
о $A = 1, B = 7, C = 0$
4. Шаг 4: Опустошаем колбу А, затем снова наполняем её полностью.
о $A = 8, B = 7, C = 0$
5. Шаг 5: Переливаем воду из А в В. Поскольку в В уже имеется 7 единиц воды, и его максимальный объем – 10 единиц, в неё можно добавить ещё 3 единицы. Таким образом, в А останется 5 единиц воды.
о $A = 5, B = 10, C = 0$

Таким образом, после выполнения всех шагов, итоговое распределение воды следующее:

- В колбе А осталось 5 единиц воды,
- В колбе В – 10 единиц воды,
- Колба С пуста.

Суммарное количество воды во всех трёх колбах равно 15 единицам.

A	B	C
0	0	0
8	0	0
1	0	7
1	7	0
0	7	0
8	7	0
5	10	0

Ответ : 15

Задание №10 (6443)

Исполнитель Черепаха действует на плоскости с декартовой системой координат. В начальный момент Черепаха находится в начале координат, её голова направлена вдоль положительного направления оси ординат, хвост опущен. При опущенном хвосте Черепаха оставляет на поле след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существует 6 команд: Поднять хвост, означающая переход к перемещению без рисования; Опустить хвост, означающая переход в режим рисования; Вперёд n (где n – целое число), вызывающая передвижение Черепахи на n единиц в том направлении, куда указывает её голова; Назад n (где n – целое число), вызывающая передвижение в противоположном голове направлении; Направо m (где m – целое число), вызывающая изменение направления движения на m градусов по часовой стрелке, Налево m (где m – целое число), вызывающая изменение направления движения на m градусов против часовой стрелки.

Запись

Повтори k [Команда1 Команда2 ... КомандаS]

означает, что последовательность из S команд повторится k раз. Черепахе был дан для исполнения следующий алгоритм:

Вперёд 2

Повтори 5 [

Вперёд x Направо 90 Вперёд 3 Направо 90

Вперёд x Налево 90 Вперёд 1 Налево 90

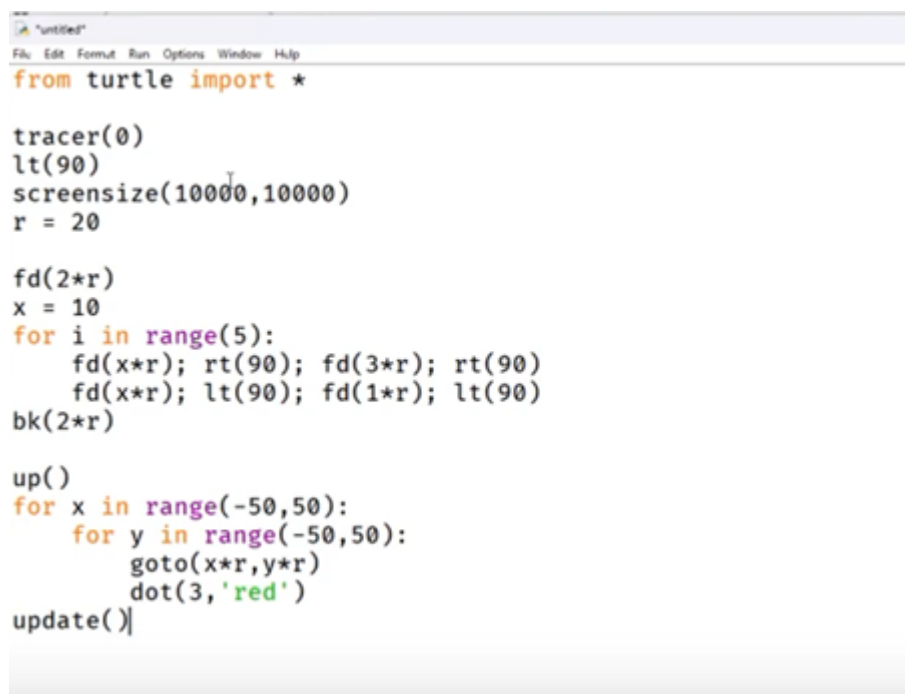
]

Назад 2

Определите, при каком наибольшем натуральном x количество точек с целочисленными координатами внутри области, ограниченной линией, нарисованной Черепахой, и осью абсцисс, меньше 25000. Точки, расположенные на линии, не учитывать.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=1h49m20s

Решение

A screenshot of a Python IDE window titled "untitled". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code in the editor is as follows:

```
from turtle import *

tracer(0)
lt(90)
screensize(10000,10000)
r = 20

fd(2*r)
x = 10
for i in range(5):
    fd(x*r); rt(90); fd(3*r); rt(90)
    fd(x*r); lt(90); fd(1*r); lt(90)
bk(2*r)

up()
for x in range(-50,50):
    for y in range(-50,50):
        goto(x*r,y*r)
        dot(3,'red')
update()
```

Нужно создать формулу, которая позволит вычислить количество точек, исходя из значения x. Нам необходимо вывести формулу, позволяющую подсчитать число точек, когда известно значение x.

(№ 6443) Исполнитель Черепаха действует на плоскости с декартовой системой координат. В начальный момент Черепаха находится в начале координат, её голова направлена вдоль положительного направления оси ординат, хвост опущен. При опущенном хвосте Черепаха оставляет на поле след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существует 6 команд: **Поднять хвост**, означающая переход к перемещению без рисования; **Опустить хвост**, означающая переход в режим рисования; **Вперёд n** (где n – целое число), вызывающая передвижение Черепахи на n единиц в том направлении, куда указывает её голова; **Назад n** (где n – целое число), вызывающая передвижение в противоположном голове направлении; **Направо m** (где m – целое число), вызывающая изменение направления движения на m градусов по часовой стрелке, **Налево m** (где m – целое число), вызывающая изменение направления движения на m градусов против часовой стрелки.

Запись

Повтори k [Команда1 Команда2 ... КомандаS]

означает, что последовательность из S команд повторится k раз. Черепахе был дан для исполнения следующий алгоритм:

Вперёд 2

Повтори 5 [

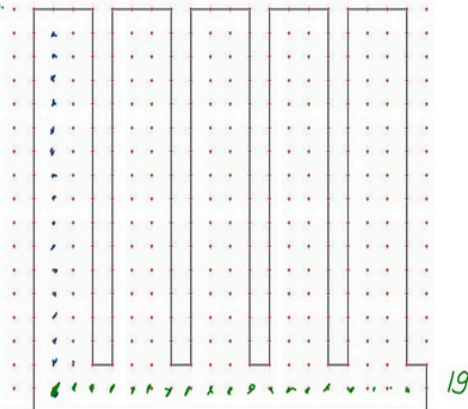
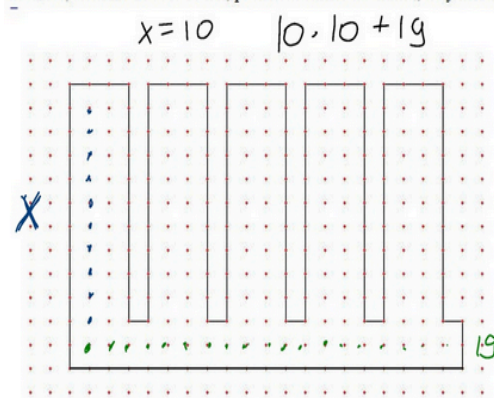
Вперёд x Направо 90 Вперёд 3 Направо 90

Вперёд x Налево 90 Вперёд 1 Налево 90

]

Назад 2

Определите, при каком наибольшем натуральном x количество точек с абсциссой, меньше 25000. Точки, расположенные на линии, не учитывать.



В общем виде $10x + 19$

$x=15$ $10 \cdot 15 + 19$

$$10x + 19 < 25000$$

$$x < \frac{25000 - 19}{10}$$

$$x < 2498,1$$

$$x_{\max} = 2498$$

й линией, нарисованной Черепахой, и осью

У нас есть фигура, похожая на букву "Щ". Точки на линиях учитывать не требуется. Нужно найти связь между x и количеством точек. Допустим, $x = 10$. Это определяет высоту фигуры.

Посчитаем несколько примеров. Здесь у нас 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 точек. Если увеличить длину, станет 11 точка. Таким образом, количество точек здесь не 10, как могло бы показаться, а x.

Если мы увеличим фигуру до 15 точек, разница станет видна. Создадим вторую картинку для наглядности. На ней видно, что при $x = 15$, количество точек будет 15.

Таким образом, линейная зависимость прослеживается четко. В одном ряду 19 точек. Независимо от изменения x, в этом ряду всегда будет 19 точек. Только столбцы изменяются в зависимости от x.

Итак, сколько всего точек? Посмотрим: $10x + 19$. В общем случае формула выглядит так: $10x + 19$.

Теперь, зная эту формулу, мы можем решить задачу: при каком максимальном значении x общее количество точек будет меньше 25000?

Решив простое неравенство, получаем, что x должно быть меньше $(25000 - 19) / 10$. Ответ: максимальное значение x – 2498.

Ответ : 2498

Задание №11 *

Исполнитель Черепаха передвигается по плоскости и оставляет след в виде линии. Черепаха может выполнять две команды: Вперёд n (n – число) и Направо m (m – число). По команде Вперёд n Черепаха перемещается вперёд на n условных единиц. По команде Направо m Черепаха поворачивается на месте на m градусов по часовой стрелке, при этом соответственно меняется направление дальнейшего движения.

Запись Повтори k [Команда1 Команда2 ... КомандаS] означает, что заданная последовательность из S команд повторится k раз.

В начальный момент на поле находятся две Черепахи. Первая Черепаха находится в начале координат и направлена вверх (вдоль положительного направления оси ординат). Вторая Черепаха находится в неизвестной точке поля и направлена вправо (вдоль положительного направления оси абсцисс).

Каждая Черепаха выполнила следующую программу:

Повтори 2 [Вперёд 15 Направо 90 Вперёд 8 Направо 90]

Определите максимально возможное количество точек с целочисленными координатами, которые могут оказаться внутри пересечения фигур, нарисованных двумя Черепахами. Точки, находящиеся на линиях, не учитывать.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241244?t=2h2m10s

Решение

Для начала рассмотрим первую черепаху, которая находится в начале координат и рисует прямоугольник размером 15×8 , направленный вертикально. Вторая черепаха также рисует аналогичный прямоугольник, но он направлен горизонтально.

Мы должны найти такое расположение этих двух прямоугольников относительно друг друга, чтобы пересечение содержало наибольшее количество целых точек.

Анализ положения:

Первая черепаха рисует прямоугольник с вершинами в точках $(0, 0)$, $(15, 0)$, $(15, 8)$ и $(0, 8)$.

Положение этой черепахи фиксировано.

Вторая черепаха может располагаться в любой точке поля, поэтому мы можем перемещать её и поворачивать прямоугольник на 90 градусов относительно первого.

Пересечение прямоугольников:

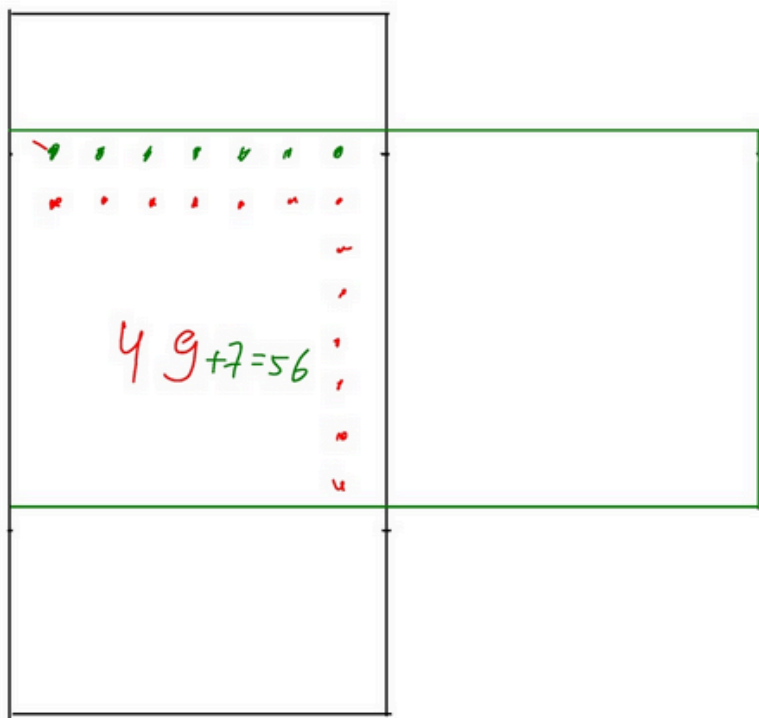
При пересечении двух прямоугольников образуется область, которая содержит некоторое количество целых точек. Если оба прямоугольника расположены таким образом, что их стороны совпадают с линиями сетки, то максимальное количество точек в пересечении равно 49 (7×7).

Однако, если вторую черепаху сместить на половину клетки вверх, то количество доступных точек увеличится. При таком смещении верхняя граница прямоугольника второй черепахи пройдёт через середину клеток, делая доступными дополнительные точки.

Итоговое решение:

Таким образом, после смещения второй черепахи на половину клетки вверх, общее количество точек с целочисленными координатами в области пересечения увеличивается до 56 .

Это максимальный результат, который можно достичь при условии, что положение первой черепахи фиксировано. Если бы обе черепахи могли свободно перемещаться, то можно было бы добиться ещё большего количества точек, сдвигая первый прямоугольник аналогичным образом.



Ответ : 56

Telegram: @fast_ege