

Обзор возможных Исполнителей в заданиях ГИА

21 марта 2011 г.

Васинькина Н.Н.,
учитель информатики и ИКТ
МОУ «СОШ №51» г. Саратова
natalivas2006@yandex.ru

На Государственной итоговой аттестации Исполнители встречаются в различных видах задач, например:

- задана система команд формального исполнителя, приведён конкретный алгоритм, результат работы которого требуется определить
(Задания с выбором ответа)
- задана система команд формального исполнителя, приведён некоторый алгоритм, требуется составить алгоритм с меньшим числом шагов, приводящий к такому же результату
(Задания с кратким ответом)
- задана система команд формального исполнителя, требуется составить алгоритм, приводящий к конкретному результату
(Задания с развёрнутым ответом)

СОДЕРЖАНИЕ

№8. Задания с выбором одного ответа

- Черепашка №№ 11, 21, 2, 31, 2, 3, 41, 2, 3, 4, 51, 2, 3, 4, 5, 61, 2, 3, 4, 5, 6, 71, 2, 3, 4, 5, 6, 7, 8

№16. Задания с кратким ответом

- Вычислитель 1, Делитель 1, Утроитель, Удвоитель №№ 11, 2, Вычислитель 2.12.1, 2.2, Конструктор 1.3 1.3, 1.P

№ 23. Задания с развёрнутым ответом

- Робот №№ 11, 21, 2, 31, 2, 3, 41, 2, 3, 4, 51, 2, 3, 4, 5, 61, 2, 3, 4, 5, 6, 7

ПОКА «Экзотика»

- Задания с выбором одного ответа

Черепашка № 9, Кузнечик №№ 11, 21, 2, 31, 2, 3, 4,

Черепашка (?) 1.11.1, 1.21.1, 1.2, 2, Робот (?) 1, Карандаш №№ 11, 2,

- Задания с кратким ответом

Черепашка №№ 10 (10 (I сп.)10 (I сп.), 10 (10 (I сп.), 10 (II10 (I сп.), 10 (II сп.)10 (I сп.), 10 (II сп.), 1110 (I сп.), 10 (II сп.), 11, 12

Попрыгунчик №№ 11, 2, Чертёжник

- Задания с развёрнутым ответом

№8. Задания с выбором одного ответа

Черепашка

Черепашка

- Исполнитель Черепашка перемещается на экране компьютера, оставляя след в виде линии.
- В каждый конкретный момент известно положение исполнителя и направление его движения.
- СКИ исполнителя (две команды):
 - Вперёд n (где n — целое число), вызывающая передвижение Черепашки на n шагов в направлении движения.
 - **Направо** m (где m — целое число), вызывающая изменение направления движения на m градусов по часовой стрелке.
- Запись **Повтори k [Команда 1 Команда 2 Команда 3]** означает, что последовательность команд в скобках повторится k раз.

Черепашке дан для исполнения алгоритм:

Повтори **6** [Направо 60 Вперёд 15]

Какая фигура появится на экране?

1. правильный пятиугольник
2. правильный треугольник
3. правильный шестиугольник
4. правильный двенадцатиугольник

Решение.

Выпуклый многоугольник называется правильным, если у него все стороны равны и все углы равны.

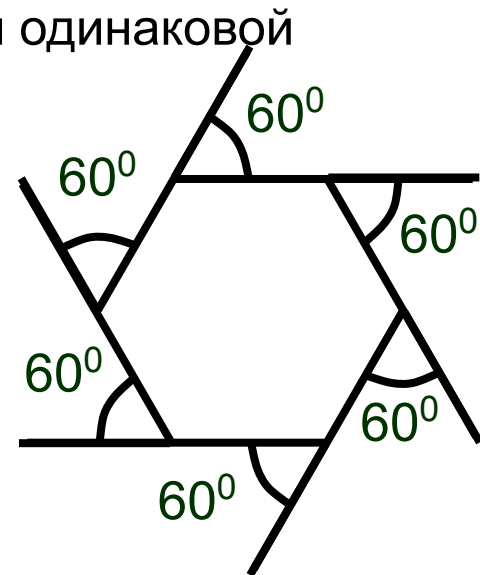
Черепашка шесть раз повторяет группу команд, указанную в скобках. При этом она каждый раз поворачивает в одном направлении — направо, на один и тот же угол и прочерчивает отрезки одинаковой длины.

Выполнив весь алгоритм, Черепашка совершит поворот на $60^\circ \cdot 6 = 360^\circ$.

Сумма внешних углов многоугольника = 360° .

Следовательно, получается замкнутая ломаная, образующая правильный многоугольник, состоящий из шести равных сторон, т.е.

правильный шестиугольник.





Черепашке был дан для исполнения следующий алгоритм:

Повтори 8 [Направо 45 Вперёд 18 Направо 45]

Какая фигура появится на экране?

- 1) восьмиконечная звезда
- 2) правильный восьмиугольник
- 3) незамкнутая ломаная линия
- 4) квадрат

Решение.

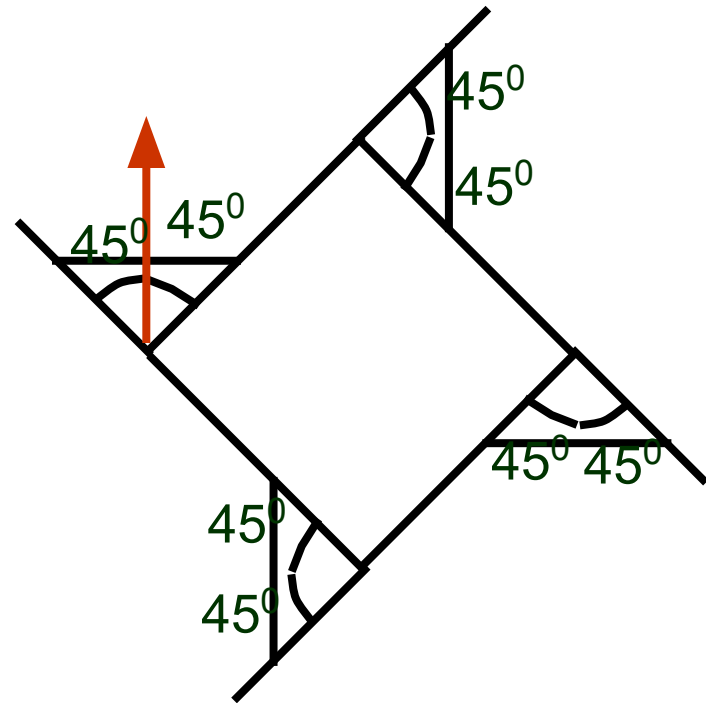
Если это правильный многоугольник, то сумма его внешних углов = 360° .

• Величина одного внешнего угла = $45^{\circ} + 45^{\circ} = 90^{\circ}$

$360^{\circ} / 90^{\circ} = 4$ (делится нацело)

⇒ КВАДРАТ,

а поскольку **Повтори 8...**, квадрат нарисуетя дважды по одному контуру



3) Черепашке был дан для исполнения следующий алгоритм:

Повтори 3 [Направо 45 Вперёд 18 Направо 45]

Какая фигура появится на экране?

- 1) восьмиконечная звезда
- 2) правильный восьмиугольник
- 3) незамкнутая ломаная линия
- 4) квадрат

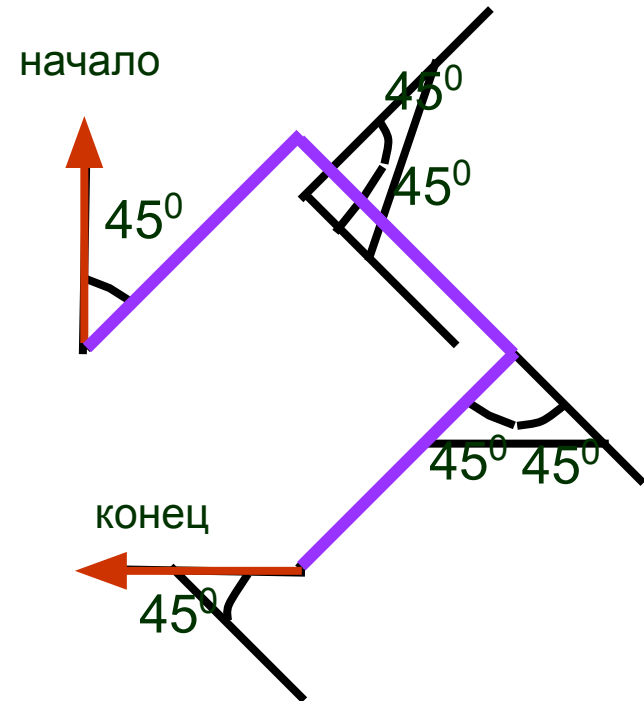
Решение.

Если это правильный многоугольник, то сумма его внешних углов = 360° .

Величина одного внешнего угла = $45^{\circ} + 45^{\circ} = 90^{\circ}$

$360^{\circ} / 90^{\circ} = 4$ (делится нацело) \Rightarrow КВАДРАТ,

НО!!! **Повтори 3**, а нужно 4 как минимум! \Rightarrow Ломаная не замыкается





Черепашке был дан для исполнения следующий алгоритм:

Повтори 4 [Направо 45 Вперёд 25 Направо 75]

Какая фигура появится на экране?

- 1) ромб
- 2) правильный треугольник
- 3) незамкнутая ломаная линия
- 4) правильный восьмиугольник

5
Черепашке был дан для исполнения следующий алгоритм:

Повтори 7 [Направо 33 Вперёд 10 Направо 39]

Какая фигура появится на экране?

- 1) правильный семиугольник
- 2) правильный треугольник
- 3) правильный пятиугольник
- 4) правильный четырнадцатиугольник

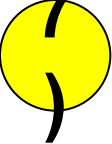


**Повтори 5 [Вперёд 36 Направо 36
Вперёд 36 Направо 36]**

Какая фигура появится на экране?

- 1) правильный треугольник
- 2) равнобедренная трапеция
- 3) правильный пятиугольник
- 4) правильный десятиугольник





Исполнитель *Черепашка* перемещается на экране компьютера, оставляя след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существуют две команды:

Вперёд n (где n — целое число), вызывающая передвижение *Черепашки* на n шагов в направлении движения;

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

Алгоритм: Повтори 5 [Команда1, Команда2] означает, что последовательность команд в скобках повторяется 5 раз.

Черепашке был дан для исполнения следующий алгоритм:

Повтори 8 [Вперёд 10 **Налево 40]**

Какая фигура появится на экране?

1. незамкнутая ломаная линия
2. квадрат
3. правильный девятиугольник
4. правильный восьмиугольник



Черепашке был дан для исполнения следующий алгоритм:

Повтори 9 [Вперёд 10 Направо 60 Вперёд 10]

Какая фигура появится на экране?

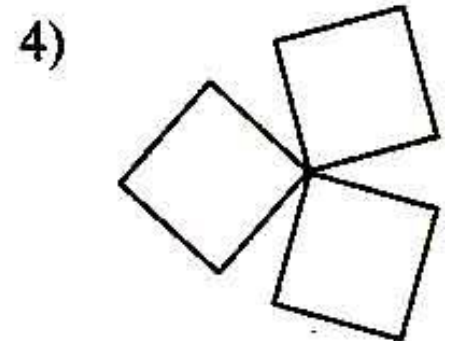
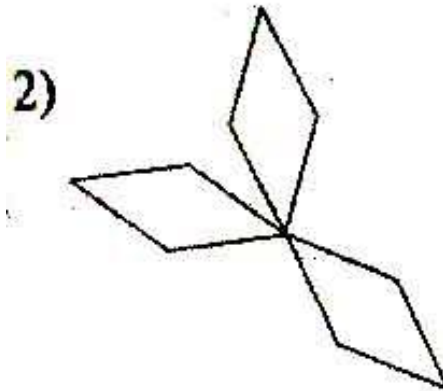
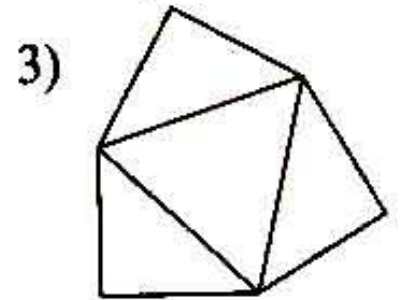
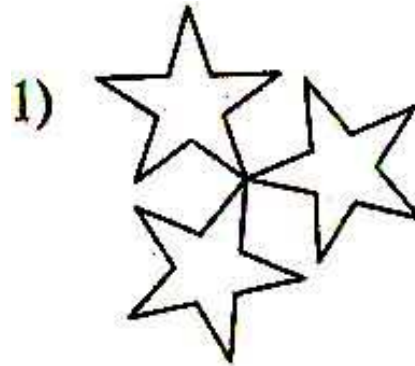
- 1) правильный шестиугольник
- 2) незамкнутая ломаная линия
- 3) квадрат
- 4) правильный девятиугольник
- 5) правильный треугольник





• Черепашке был дан для исполнения следующий алгоритм:

- Повтори 3 [Повтори 4 [Вперёд 30 Направо 50 Вперёд 30 Направо 130] Направо 130]
- Какая фигура появится на экране?



0.

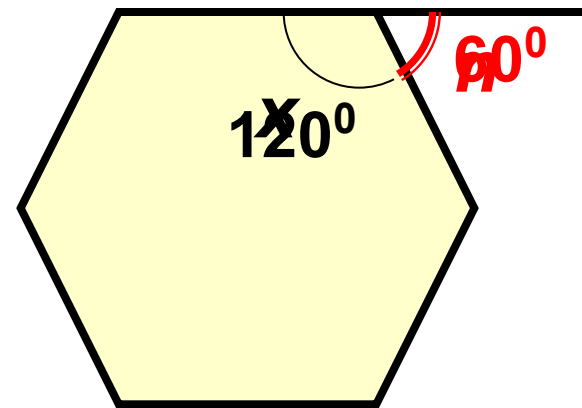
Каким должно быть значение n в цикле:

Повтори 10 [Вперед 10 Направо n], чтобы на экране появился правильный шестиугольник?

Решение (I способ).

Угол поворота Черепашки в вершине шестиугольника будет равен углу, смежному с внутренним углом, т.е.

$$n = 180^{\circ} - x.$$



Сумма внутренних углов правильного шестиугольника вычисляется по формуле $(n-2) \times 180^{\circ}$, где $n = 6$. \Rightarrow

\Rightarrow величина одного внутреннего угла x :

$$x = \frac{(n - 2) \cdot 180^{\circ}}{n} = \frac{(6 - 2) \cdot 180^{\circ}}{6} = 120^{\circ}.$$

$$n = 180^{\circ} - x = 180^{\circ} - 120^{\circ} = 60^{\circ}.$$

0.

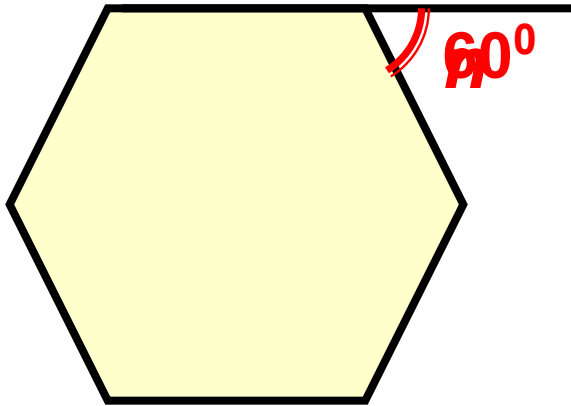
2. Каким должно быть значение n в цикле:

Повтори 10 [Вперед 10 Направо n], чтобы на экране появился правильный шестиугольник?

Решение (II способ).

Сумма **ВНЕШНИХ** углов правильного многоугольника 360^0 .

У нас правильный шестиугольник $\Rightarrow n = 360^0/6 = 60^0$



1 Каким должно быть значение n в цикле:
1 Повтори 10 [Вперед 10 Направо n], чтобы на экране появилась правильная пятиугольная звезда?

Решение.

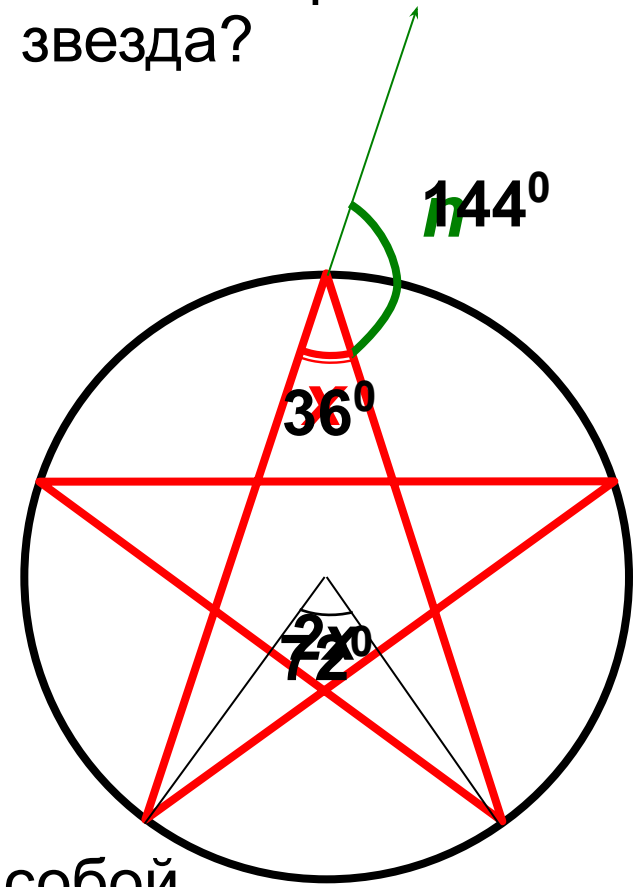
Угол поворота Черепашки в вершине звезды будет равен углу, смежному с внутренним углом, т.е. $n = 180^\circ - x$.

Внутренний угол звезды является вписанным в окружность. Соответствующий ему центральный угол вдвое больше вписанного, т.е. $2 \cdot x$.

Звезда 5-угольная правильная \Rightarrow
вписанных углов 5 и они равны между собой
 \Rightarrow центральных углов 5.

Сумма 5 центральных углов = 360° . $\Rightarrow 2 \cdot x = 360^\circ : 5 = 72^\circ$
 $\Rightarrow x = 36^\circ \Rightarrow$

$n = 180^\circ - 36^\circ = 144^\circ$.



1
2 Черепашке был дан для исполнения следующий алгоритм:

повтори N [вперед 10 направо X]

Какими должны быть значения **X** и минимальное **N**, чтобы на экране появился правильный девятиугольник?
(выведите результат через запятую)

Решение

1. Сумма внешних углов многоугольника = 360° ,
все углы правильного многоугольника равны
и \Rightarrow равны его внешние углы

$$\Rightarrow x = \frac{360^0}{9} = 40^0$$

2. **$N \geq 9$** , т.к. иначе ломаная не замкнётся!
(9 сторон в 9-тиугольнике)

Задания с кратким ответом

- Вычислитель

1) **Исполнитель Вычислитель** имеет следующую систему пронумерованных команд:

1. **умножь на два;**
2. **прибавь единицу.**

Первая умножает число на два, а вторая прибавляет к числу единицу.

Алгоритм, преобразующий число 3 в число 26, записывается в виде последовательности команд 1121, что соответствует:

1. умножь на два $3 \cdot 2 = 6$
1. умножь на два $6 \cdot 2 = 12$
2. прибавь единицу $12 + 1 = 13$
1. умножь на два $13 \cdot 2 = 26$

Запишите порядок команд алгоритма, преобразующего **число 3 в число 21**, содержащего **не более пяти команд**, указывая лишь номера команд.

1) Подобные задачи принято решать «от ответа»

Исходная задача: 3 → 21

- умножь на два;
- прибавь единицу.

Решим обратную задачу:
получить из числа 21 число 3.

Применяемые команды также должны быть обратными к заданным командам исполнителя:

- дели на два;
- вычти единицу.

Операция **дели на два** выполняется только для чисел, кратных двойке, иначе она не будет обратной исходной команде **умножь на два**.

Для скорейшего получения из числа 21 числа 3 по возможности применяем операцию деления, а если это невозможно — операцию вычитания.

«Обратная» команда	Выполнение команды
2. вычти единицу	$21 - 1 = 20$
1. дели на два	$20 : 2 = 10$
1. дели на два	$10 : 2 = 5$
2. вычти единицу	$5 - 1 = 4$
2. вычти единицу	$4 - 1 = 3$

Результат решения обратной задачи — полученную последовательность команд — переписываем в обратном порядке для получения ответа исходной задачи.

Исполнитель Делитель

У исполнителя *Делитель* две команды, которым присвоены номера:

1. прибавь пять
2. раздели на два

Первая команда увеличивает число на экране на 5, вторая — уменьшает его в 2 раза.

Запишите порядок команд в алгоритме получения **из числа 1** числа **13**, содержащем **не более 5 команд**, указывая лишь номера команд.

Например, последовательность **11221** соответствует алгоритму, который преобразует число 7 в 8:

1. прибавь пять (12)
2. раздели на два (6)
1. прибавь пять (11)
1. прибавь пять (16)
2. раздели на два (8)

Если таких алгоритмов более одного, то запишите любой из них.

Исполнитель Утроитель

У исполнителя *Утроитель* две команды, которым присвоены номера:

1. умножь на три
2. вычти 2

Первая из них утраивает число на экране, вторая уменьшает его на 2.

Запишите порядок команд в алгоритме получения **из числа 9 числа 19**, содержащем **не более 3 команд**, указывая лишь номера команд.

Например, 21211 — это алгоритм:

1. умножь на три
2. вычти 2
1. умножь на три
2. вычти 2
2. вычти 2, — который преобразует число 2 в 8.

1)

Исполнитель **Удвоитель**

У исполнителя *Удвоитель* две команды, которым присвоены номера:

1. вычти 3

2. умножь на 2

Первая из них уменьшает число на экране на 3, вторая — удваивает его.

Запишите порядок команд в алгоритме получения **из числа 11 числа 13**, содержащем **не более 3** команд, указывая лишь номера команд.

Например, 212111 — это алгоритм:

- умножь на 2
- вычти 3
- умножь на 2
- вычти 3
- вычти 3
- вычти 2, — который преобразует число 5 в 8.

Если таких алгоритмов более одного, то запишите любой из них.

Исполнитель Удвоитель

У исполнителя *Делитель* две команды, которым присвоены номера:

- 1) **вычти три**
- 2) **умножь на два**

Первая команда уменьшает число на экране на 3, вторая — увеличивает его в 2 раза.

Запишите порядок команд в алгоритме получения **из числа 16 числа 34**, содержащем **не более 5 команд**, указывая лишь номера команд.

Например, последовательность 22121 соответствует алгоритму, который преобразует число 2 в 7:

2. умножь на два (4)
2. умножь на два (8)
1. вычти три (5)
2. умножь на два (10)
1. вычти три (7).

Если таких алгоритмов более одного, то запишите любой из них.

2)

Исполнитель Вычислитель

У исполнителя *Вычислитель* три команды, которым присвоены номера:

1. вычти 1
2. умножь на 3
3. прибавь 3

Первая из них уменьшает число на экране на 1, вторая — утраивает его, а третья увеличивает на 3.

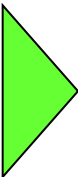
Запишите порядок команд в алгоритме получения из числа 5 числа 23 за наименьшее число команд.

Например, 211 — это алгоритм:

2. умножь на 3
1. вычти 1

1. вычти 1, — который преобразует число 7 в 19.

Решение



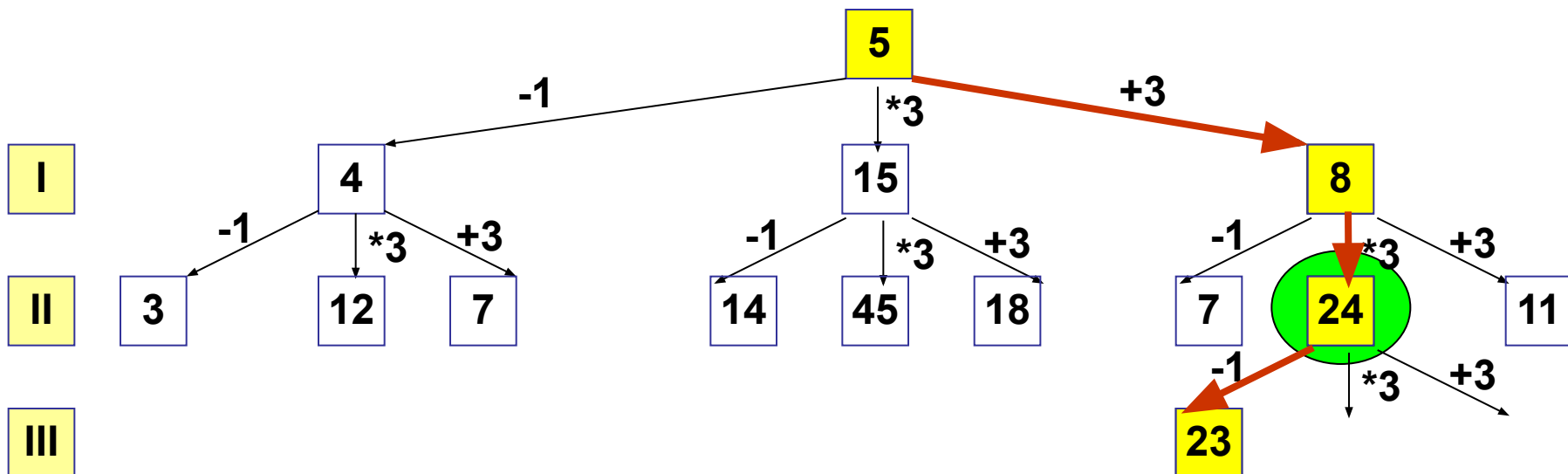
2)

Исполнитель Вычислитель

1. вычти 1
2. умножь на 3
3. прибавь 3

Для решения данной задачи
полезно построить дерево. Строим!

Запишите порядок команд в алгоритме получения
из числа 5 числа 23 за наименьшее число команд.



Исполнитель Конструктор

У исполнителя Конструктор две команды, которым присвоены номера:

1. **приписать 2**
2. **разделить на 2.**

Первая из них приписывает к числу на экране справа цифру 2, вторая – делит его на 2.

Запишите порядок команд в алгоритме получения из числа 1 числа 16, содержащем не более 5 команд, указывая только номера команд

(например, 22212 – это алгоритм:

разделить на 2

разделить на 2

разделить на 2

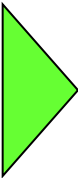
приписать 2

разделить на 2,

который преобразует число 8 в число 6.)

Если таких алгоритмов более одного, запишите любой из них.

Решение



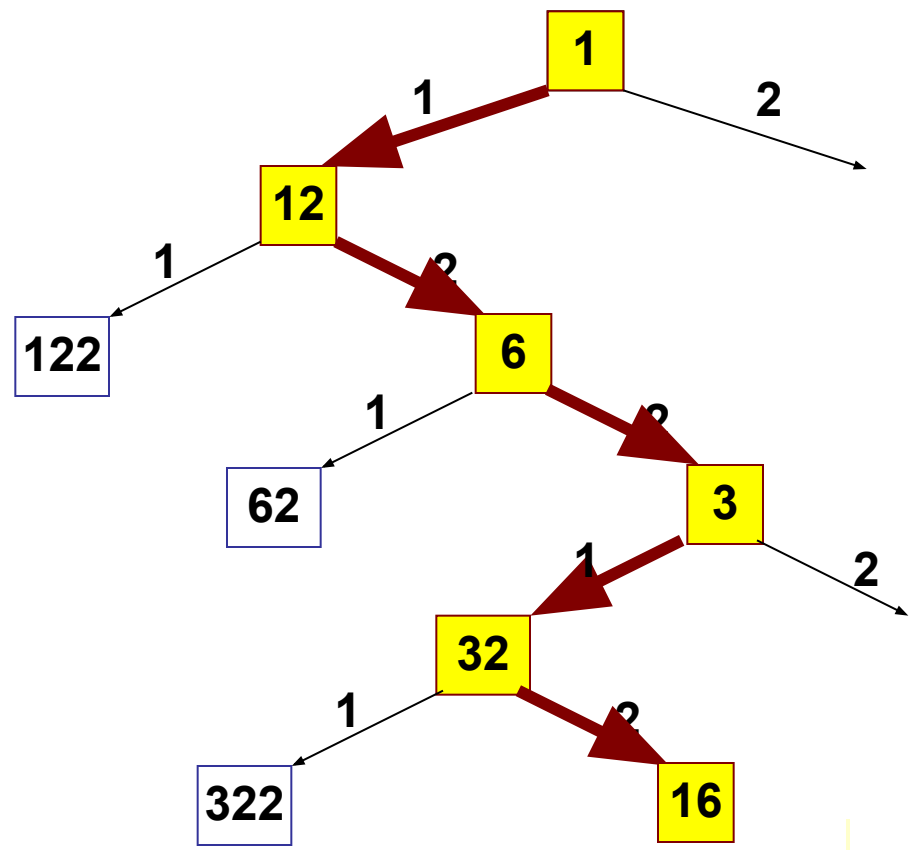
1.P Исполнитель Конструктор

СКИ:

1. приписать 2
2. разделить на 2.

Получить из числа 1 число 16 (максимум за 5 команд).

- I
- II
- III
- IV
- V



Задания с развёрнутым ответом

- Робот

23. Исполнитель *Робот* умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки.

У *Робота* есть четыре команды перемещения:

вверх
вниз
влево
вправо.

При выполнении любой из этих команд *Робот* перемещается на одну клетку соответственно:
вверх ↑, вниз ↓, влево ←, вправо →.

Между соседними (по сторонам) клетками может стоять стена, через которую *Робот* пройти не может. Если *Робот* получает команду передвижения через стену, то он разрушается.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится *Робот*:

сверху свободно
снизу свободно
слева свободно
справа свободно.

Эти команды можно использовать с условием «если», имеющим следующий вид:

если <условие> то
последовательность команд
все

Последовательность команд — это одна или несколько любых команд *Робота*. Например, для передвижения на одну клетку вправо, если справа нет стенки, можно использовать такой алгоритм:

если справа свободно то
вправо
все

В одном условии можно использовать несколько команд, используя логические связки: «и», «или», «не». Например,

если (справа свободно) и (не снизу свободно)
то
вправо
все

Для повторения последовательности команд можно использовать цикл «пока», имеющий следующий вид:

нц пока <условие>
последовательность команд
кц

Например, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

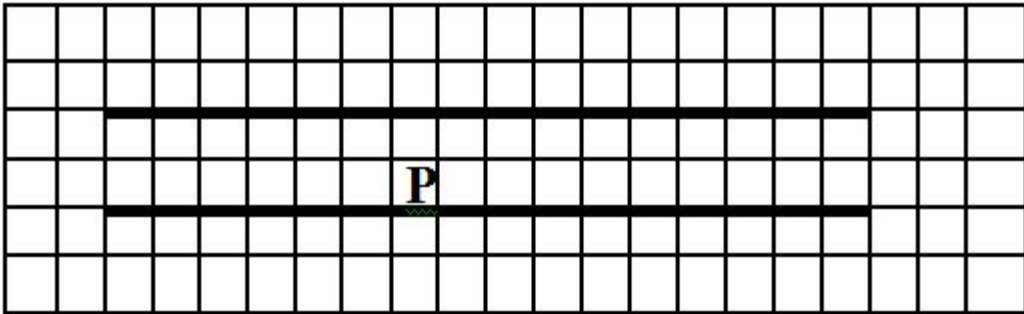
нц пока справа свободно
вправо
кц

У *Робота* есть команда закрасить, закрашивающая клетку, в которой находится *Робот* в настоящий момент.

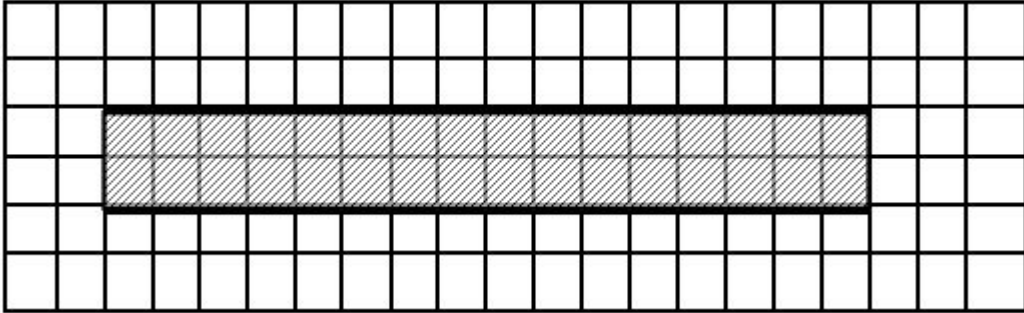
1)

Выполните задание

На бесконечном поле имеются две длинные горизонтальные стены одинаковой длины. Длина стен неизвестна. Расстояние между ними — две клетки. *Робот* находится в одной из клеток, расположенной между стен. Начальное положение *Робота* неизвестно. Одно из возможных положений *Робота* приведено на рисунке (*Робот* обозначен буквой «Р»):



Напишите для *Робота* алгоритм, закрашивающий все клетки, расположенные между стенами. *Робот* должен закрасить только клетки, удовлетворяющие данному условию. Например, для рисунка выше *Робот* должен закрасить следующие клетки:



Решение 

Конечное расположение *Робота* может быть произвольным. Алгоритм должен решать задачу для произвольного размера стены и любого допустимого начального положения *Робота*.

Алгоритм напишите в текстовом редакторе и сохраните в текстовом файле.

1)

Возможный вариант алгоритма:

содержани
е

алг Закраска

нач

если сверху свободно то

вверх

все

нц пока ~~сверху не свободно~~ не сверху свободно

влево

кц

вправо

нц пока ~~сверху не свободно~~ не сверху свободно

закрасить

вправо

кц

вниз

влево

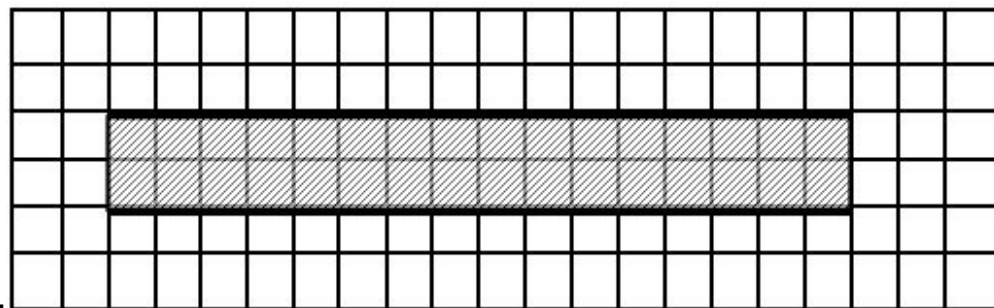
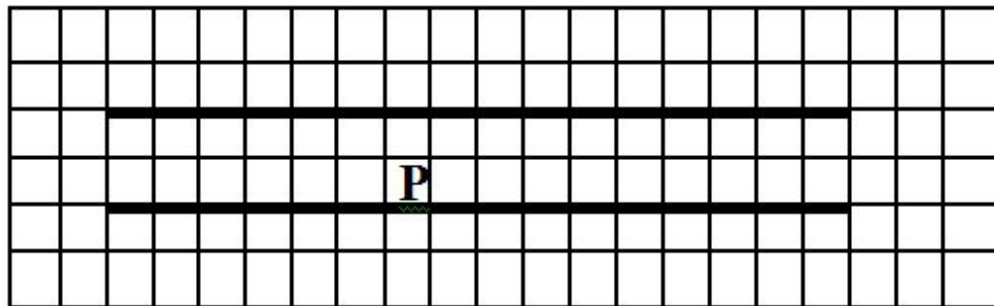
нц пока ~~снизу не свободно~~ не снизу свободно

закрасить

влево

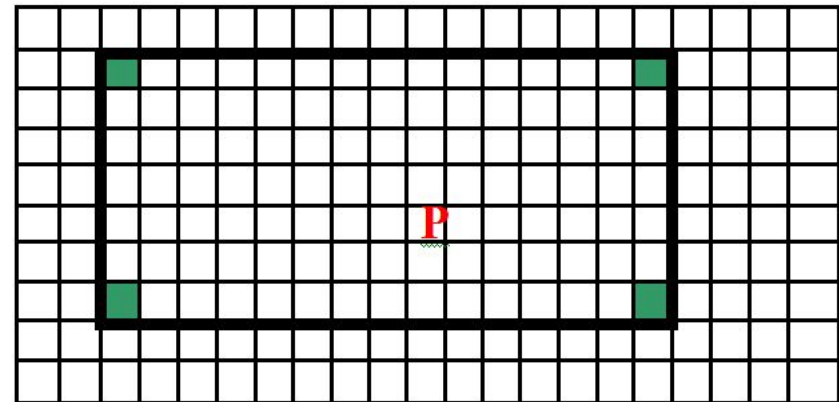
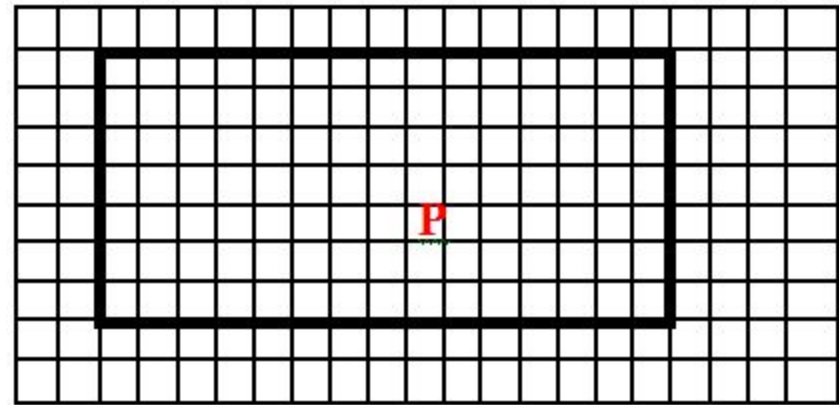
кц

кон

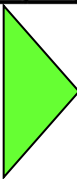


- На бесконечном поле имеется прямоугольник из стен. Длина противоположных стен прямоугольника неизвестна. Расстояние между противоположными стенами — не менее двух клеток. *Робот* находится в одной из клеток, расположенной внутри прямоугольника из стен. **Начальное положение *Робота* неизвестно.** Одно из возможных положений *Робота* приведено на рисунке (*Робот* обозначен буквой «Р»).
- Напишите для *Робота* алгоритм, закрашивающий все внутренние угловые клетки прямоугольника из стен. *Робот* должен закрасить только клетки, удовлетворяющие данному условию. Например, для исходного рисунка *Робот* должен закрасить клетки, заштрихованные на рисунке ниже.
- Конечное расположение *Робота* может быть произвольным. Алгоритм должен решать задачу для произвольного размера стены и **любого допустимого начального расположения *Робота*.** Алгоритм напишите в текстовом редакторе и сохраните в текстовом файле.

Выполните задание



Решение



Возможный вариант алгоритма:

нц пока слева свободно
влево

кц

нц пока сверху свободно
вверх

кц

закрасить

нц пока справа свободно
вправо

кц

закрасить

нц пока снизу свободно
вниз

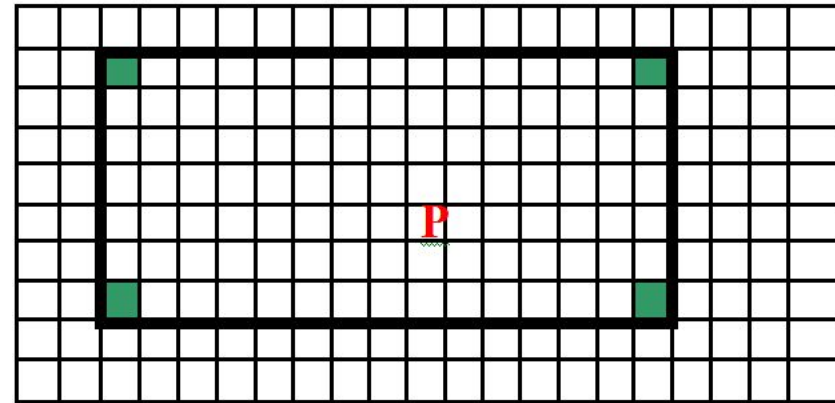
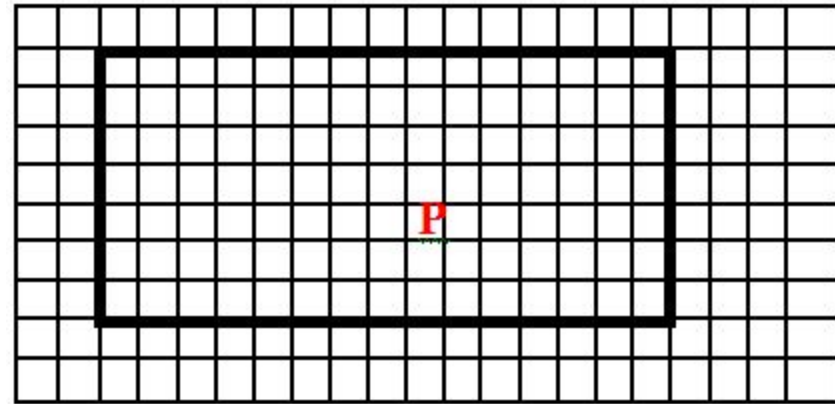
кц

закрасить

нц пока слева свободно
влево

кц

закрасить



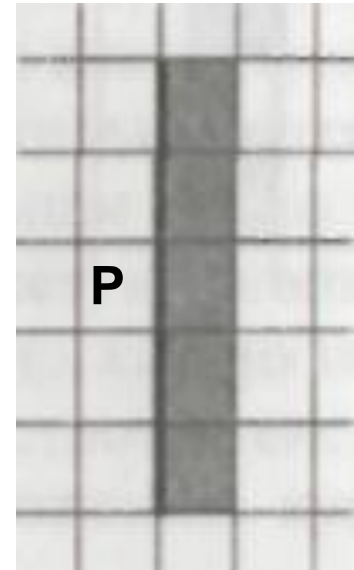
Выполните задание

На бесконечном поле клеток имеется вертикальная стена. Длина стены неизвестна. Робот размещён в одной из клеток непосредственно справа от стены. Точная начальная позиция Робота неизвестна. Возможное начальное положение Робота приведено на рисунке буквой **P**.

а) Напишите для Робота алгоритм, закрашивающий все клетки, расположенные справа от стены и прилегающие к ней. Например, для приведённого начального значения **P** Робот должен закрасить следующие клетки (см. рис.):

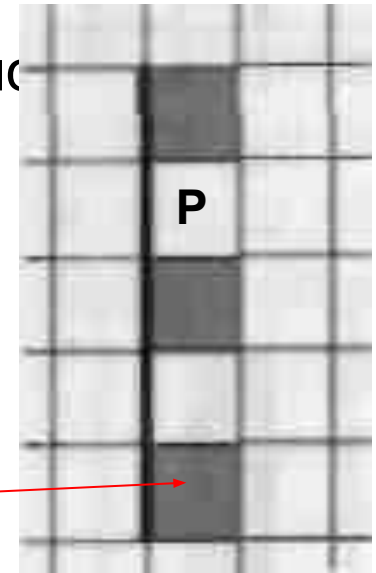
Конечное положение Робота может быть произвольным. Алгоритм должен решать задачу для произвольного размера стены и любой допустимой начальной позиции Робота.

Запишите алгоритм в текстовом редакторе и сохраните на рабочем столе в текстовом файле с именем «алгоритм».

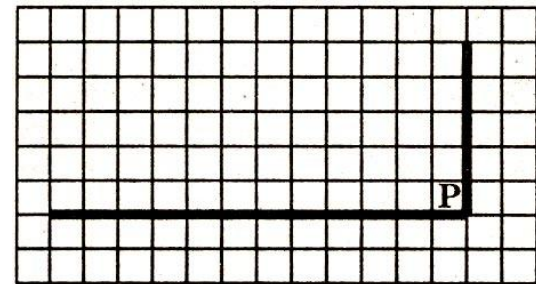


Выполните задание

- На бесконечном поле клеток имеется вертикальная стена. Длина стены неизвестна. Робот размещён в одной из клеток непосредственно справа от стены. Точная начальная позиция Робота неизвестна. Возможное начальное положение Робота приведено на рисунке (символом **P**).
- а) Напишите для Робота алгоритм, закрашивающий клетки через одну, **начиная с самой нижней клетки**, расположенные справа от стены и прилегающие к ней. Например, для начального положения **P** Робот должен закрасить следующие клетки: _____
- Конечное положение Робота может быть произвольным. Алгоритм должен решать задачу для произвольного размера стены и любой допустимой начальной позиции Робота.
- Запишите алгоритм в текстовом редакторе и сохраните на рабочем столе в текстовом файле с именем «алгоритм».

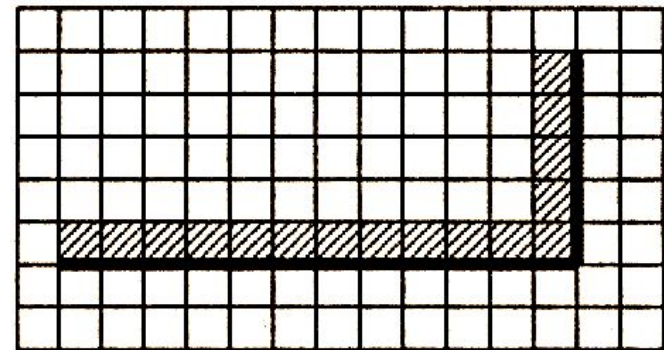


На бесконечном поле имеется горизонтальная стена. Длина стены неизвестна. От правого конца стены вверх отходит вертикальная стена также неизвестной длины. Робот находится в углу между вертикальной и горизонтальной стеной. На рисунке указана один из возможных способов расположения стен и Робота (расположение Робота обозначено буквой «Р»):



5
)

Напишите алгоритм для Робота, закрашивающий все клетки, расположенные выше горизонтальной стены и левее вертикальной стены и прилегающие к ним. Робот должен закрасить только клетки, удовлетворяющие заданному условию. Например, для приведённого рисунка Робот должен закрасить следующие клетки:



- Конечное расположение Робота может быть произвольным. Алгоритм должен решать задачу для произвольного размера поля и любого допустимого расположения стены внутри прямоугольного поля. При исполнении алгоритма **Робот не должен разрушиться**.
- Алгоритм напишите в текстовом редакторе и сохраните в текстовом файле. Название файла и каталог для сохранения вам сообщат организаторы экзамена.

- На бесконечном поле имеется длинная горизонтальная стена. Длина стены неизвестна. *Робот* находится в одной из клеток, расположенной непосредственно сверху от стены. Одно из возможных положений *Робота* приведено на рисунке (*Робот* обозначен буквой «Р»).



- Напишите для *Робота* алгоритм, закрашивающий все клетки, расположенные ниже стены и прилегающие к ней, причём через одну, **начиная с последней (?)**. Например, для приведённого рисунка *Робот* должен закрасить следующие клетки

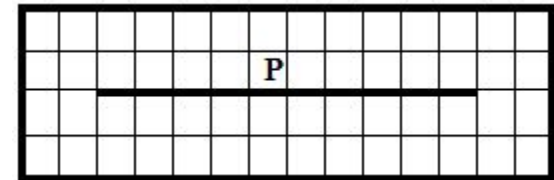


(?)

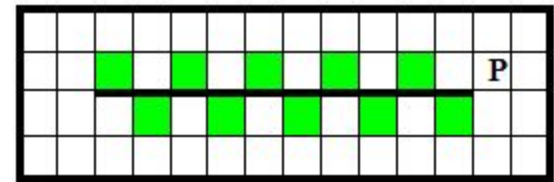


(?)

- На бесконечном поле имеется длинная горизонтальная стена (длина стены равна **чётному** числу клеток). Длина стены неизвестна. *Робот* находится в одной из клеток, расположенной **непосредственно сверху от стены**.



- Напишите для *Робота* алгоритм, закрашивающий все клетки, расположенные выше и ниже стены и прилегающие к ней, причём через одну, **начиная с первой нижней правой**.



ПОКА «Экзотика»

- **Задания с выбором одного ответа**
 Черепашка № 9, Кузнечик №№ 11, 21, 2, 31,
2, 3, 4,
 Черепашка (?) 1.11.1, 1.21.1, 1.2, 2, Робот (?) 1,
Карандаш №№ 11, 2,
- **Задания с кратким ответом**
 Черепашка №№ 10 (10 (I 10 (I сп.) 10 (I сп.),
10 (10 (I сп.), 10 (II 10 (I сп.), 10 (II сп.) 10
 (I сп.), 10 (II сп.), 11 10 (I сп.), 10 (II сп.),
11, 12
 Попрыгунчик №№ 11, 2, Чертёжник
- **Задания с развёрнутым ответом**
Плиточник Плиточник: задание 1) Плиточник:
 задание 1), алгоритм Плиточник: задание 1),
 алгоритм, задания 2)-3)

Исполнитель *Кузнечик* живёт на числовой оси

Начальное положение — точка 0.

Система команд *Кузнечика*:

вперёд 3 — *Кузнечик* прыгает вперёд на 3 единицы;

назад 2 — *Кузнечик* прыгает назад на 2 единицы;

закрась — текущая позиция *Кузнечика* закрашивается в красный цвет.

Условия могут быть следующими:

чётное — проверка того, что текущее положение — чётное число,

положительное — проверка того, что текущее положение — число > 0 ,

отрицательное — проверка того, что текущее положение — число < 0 .

1) Кузнечик выполнил следующий алгоритм 2 раза:

вперёд 3 назад 2

ЕСЛИ **чётное** ТО назад 2 закрась ИНАЧЕ вперёд 3

вперёд 3

Определите, сколько точек на числовой прямой будет закрашено в результате выполнения этого алгоритма.

1) 1 2) 2 3) 3 4) 0

вперёд 3 назад 2

ЕСЛИ **чётное**

ТО назад 2 закрась

ИНАЧЕ вперёд 3 вперёд 3

1) Кузнечик выполнил следующий алгоритм 2 раза:

вперёд 3 назад 2

ЕСЛИ **чётное**

ТО назад 2 закрась

ИНАЧЕ вперёд 3 вперёд 3

Определите, **сколько точек на числовой прямой будет закрашено** в результате выполнения этого алгоритма.

- 1) 1 2) 2 3) 3 4) 0

Решение.

- 1) $0+3-2=1$ чётное? – нет \Rightarrow ИНАЧЕ $\Rightarrow 1+3+3=7$
2) $7+3-2=8$ чётное? – да \Rightarrow ТО $\Rightarrow 8-2=6$ закрась

2) Кузнечик выполнил следующий алгоритм 2 раза:

вперёд 3 : назад 1

ЕСЛИ **отрицательное**

ТО вперёд 2

ИНАЧЕ назад 1 : закрась

КОНЕЦ

назад 1 : назад 1

Определите, сколько точек на числовой прямой будет закрашено в результате выполнения этого алгоритма.

- 1) 1 2) 2 3) 3 4) 0

Известно, что алгоритм, который выполнил *Кузнечик*, состоит из 6 записей. Первой была запись:

Повтори 35 [Вперёд 2 Назад 1] Остальные записи — это команды **Назад 6**.

На какую одну команду можно заменить этот алгоритм, чтобы *Кузнечик* оказался в той же точке, что и после выполнения алгоритма?

- 1) Назад 5 3) Вперёд 1
- 2) Вперёд 5 4) Назад 1

Известно, что начальное положение *Кузнечика* — точка 0 на координатной оси. Кузнечик выполнил алгоритм:

Повтори 3 [Вперёд 2 Назад 1 Повтори 2 [Назад 2]]

Повтори 4 [Вперёд 3]

В какой точке на координатной оси окажется *Кузнечик* после выполнения алгоритма?

- 1) 0 2) 12 3) 3 4) Назад 11

Черепашка (?) – только название

Исполнитель *Черепашка* ползёт по клеткам бесконечной вертикальной клетчатой доски, переходя по одной из команд:

вверх, вниз, вправо, влево — в соседнюю клетку в указанном направлении. *Черепашка* выполнила следующую программу:

вправо вправо влево вверх вверх вниз влево вверх, — в результате которой перешла из клетки **a** в клетку **b**.

Укажите **наименьшее возможное число команд в программе**, переводящей *Черепашку* из начальной клетки **a** в конечную клетку **b**.

- 1) 1 2) 2 3) 3 4) 4

Черепашка (?) – только название

Исполнитель *Черепашка* ползёт по клеткам бесконечной вертикальной клетчатой доски, переходя по одной из команд:

вверх, вниз, вправо, влево — в соседнюю клетку в указанном направлении. *Черепашка* выполнила следующую программу:

вправо вниз вправо вниз вправо вверх влево вверх
— в результате которой перешла из клетки *a* в клетку *b*.

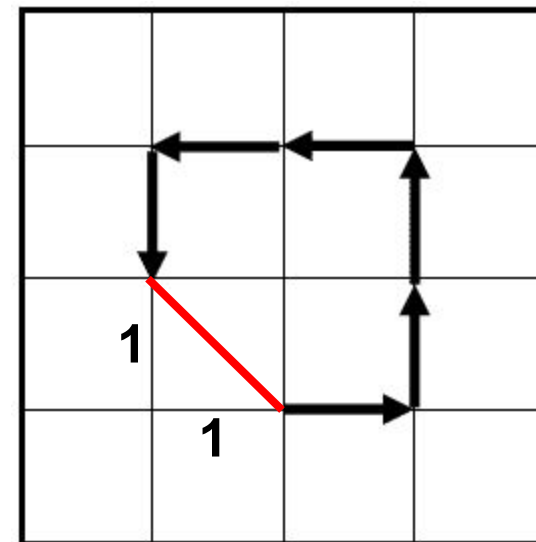
Укажите **наименьшее возможное число команд в программе**, переводящей *Черепашку* из начальной клетки *a* в конечную клетку *b*.

- 1) 1 2) 2 3) 3 4) 4

Черепашка (?)

Исполнитель *Черепашка* перемещается на экране компьютера, оставляя след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существуют четыре команды:

1. **вверх**
2. **вниз**
3. **вправо**
4. **влево,**



— при выполнении каждой из которых он движется в соответствующем направлении **на 1 см.**

Он выполнил следующую программу:

вправо вверх вверх влево влево вниз.

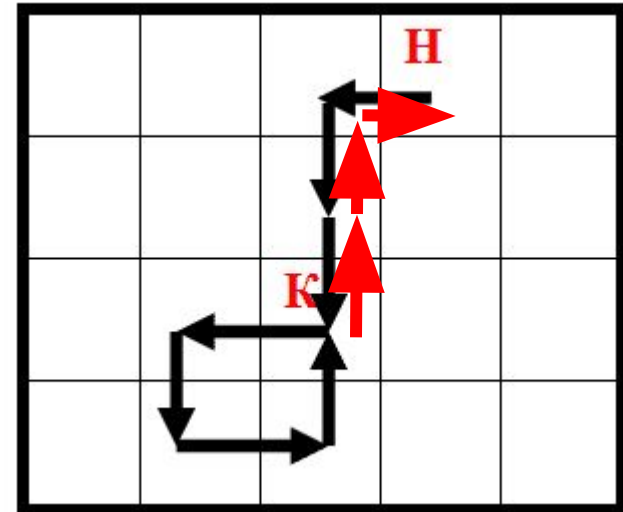
Укажите расстояние (в см) от начальной до конечной точки.

- 1) 6 2) $\sqrt{6}$ 3) $\sqrt{2}$ 4) $\sqrt{18}$

Робот (?)

Исполнитель *Робот* действует на клеточной доске, между соседними клетками которой могут стоять стены. *Робот* передвигается по клеткам доски и может выполнять команды:

1. **вверх**
2. **вниз**
3. **вправо**
4. **влево, —**



переходя на соседнюю клетку в направлении, указанном в скобках. Если в этом направлении между клетками стоит стена, то *Робот* разрушается. *Робот* успешно выполнил программу **4224231**. Какую последовательность из трёх команд должен выполнить *Робот*, чтобы вернуться в ту клетку, где он был перед началом выполнения программы, и не разрушиться вне зависимости от того, какие стены стоят на поле?

- 1) 113 2) 131 3) 311 4) 133

Исполнитель КАРАНДАШ

перемещается по плоскости, оставляя за собой след в виде линии.

Система команд исполнителя:

ВПЕРЁД K , где K — количество шагов; и

НАПРАВО P , где P — поворот на P градусов по часовой стрелке.

Для повторения последовательности команд применяется команда **ПОВТОРИ A** , где A — количество повторений.

(С Черепашкой по смыслу очень похоже!)

1)

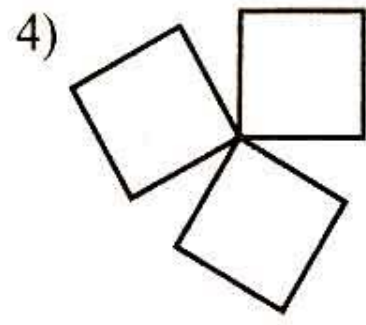
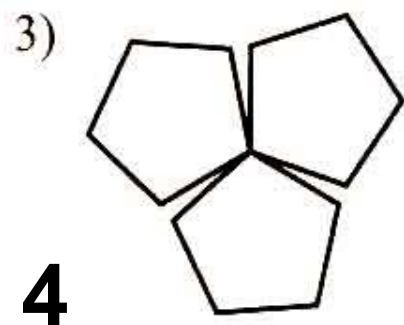
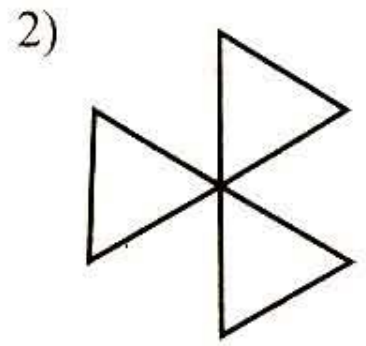
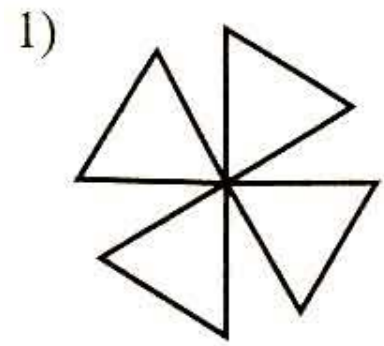
Исполнитель КАРАНДАШ

(и почему не Черепашка?)

КАРАНДАШ получил для
исполнения следующий
алгоритм:

**ПОВТОРИ 3 (ПОВТОРИ 4
(ВПЕРЁД 10 НАПРАВО 90)
НАПРАВО 120)**

Какая фигура будет
нарисована (см. рис.)?



2)

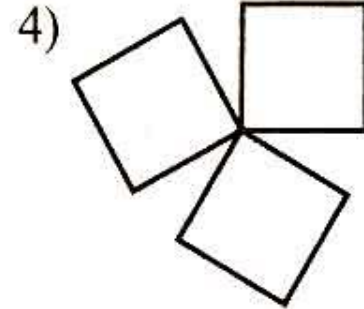
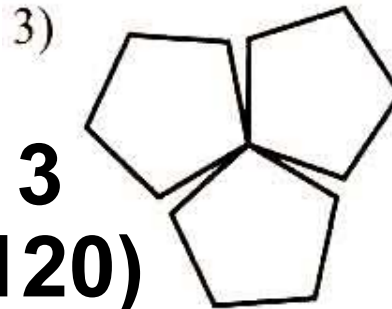
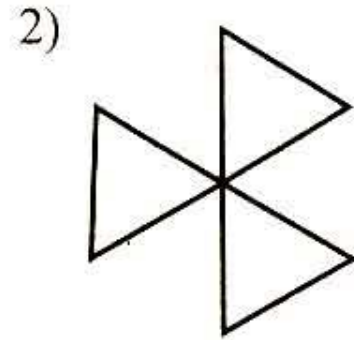
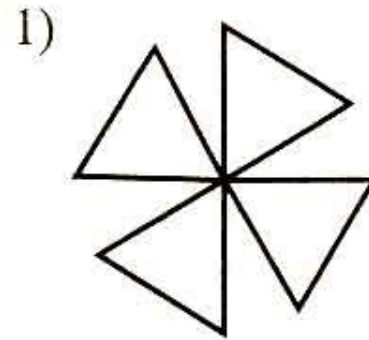
Исполнитель КАРАНДАШ

(и вновь не Черепашка!)

- КАРАНДАШ получил для исполнения следующий алгоритм:

- ПОВТОРИ 3 (ПОВТОРИ 3 (ВПЕРЁД 10 НАПРАВО 120) НАПРАВО 120)**

- Какая фигура будет нарисована (см. рис.)?





Исполнитель ПОПРЫГУНЧИК

перемещается по числовой оси по командам ВПЕРЁД P (BP) и НАЗАД P (HP), где P — количество делений оси.

Исполнитель выполнил **21** команду **H2** и **B3**, причём команд **H2** было **на 5 больше**.

Какой одной командой можно было бы заменить программу, чтобы исполнитель оказался в той же точке?

Решение.

$$\left. \begin{array}{l} X_{H2} + Y_{B3} = 21 \\ X_{H2} = Y_{B3} + 5 \end{array} \right\} \Rightarrow \begin{array}{l} 2Y_{B3} + 5 = 21 \Rightarrow 2Y_{B3} = 16 \Rightarrow Y_{B3} = 8 \\ \Rightarrow X_{H2} = 8 + 5 = 13 \end{array}$$

Т.о.,

8_{B3} : Вперед $3 \cdot 8 = 24$

13_{H2} : Назад $2 \cdot 13 = 26$

$\Rightarrow 26 - 24 = 2$, т.е. **НАЗАД 2**



Исполнитель ПОПРЫГУНЧИК

- Исполнитель ПОПРЫГУНЧИК перемещается по числовой оси по командам **ВПЕРЁД P(ВР)** и **НАЗАД P(НР)**, где P — количество делений оси.
- Исполнитель выполнил 15 команд Н2 и В3, причём команд Н2 было на 5 больше. Какой одной командой можно было бы заменить программу, чтобы исполнитель оказался в той же точке?

Решение.

$$\left. \begin{array}{l} X_{Н2} + Y_{В3} = 15 \\ X_{Н2} = Y_{В3} + 5 \end{array} \right\} \Rightarrow \begin{array}{l} 2Y_{В3} + 5 = 15 \Rightarrow 2Y_{В3} = 10 \Rightarrow Y_{В3} = 5 \\ \Rightarrow X_{Н2} = 5 + 5 = 10 \end{array}$$

Т.о.,

5_{В3}: Вперед 3*5=15

10_{Н2}: Назад 2*10=20

} $\Rightarrow 20 - 15 = 5$, т.е. **НАЗАД 5**

Исполнитель ПЛИТОЧНИК

может перемещаться по клетчатой плоскости по командам:

ВВЕРХ, ВНИЗ, ВЛЕВО, ВПРАВО.

Между соседними клетками может находиться бордюр, пересечь который ПЛИТОЧНИК не может. Поэтому при движении необходимо проверять отсутствие бордюра или края плоскости командами:

**СВЕРХУ СВОБОДНО, СНИЗУ СВОБОДНО,
СЛЕВА СВОБОДНО, СПРАВА СВОБОДНО.**

Эти команды можно использовать вместе с условием ЕСЛИ, которое имеет следующий вид:

ЕСЛИ <условие> ТО <последовательность команд> КОНЕЦ.

В одном условии можно использовать несколько команд, применяя логические связки И, ИЛИ, НЕ.

Для повторения последовательности команд можно использовать цикл ПОКА, который имеет вид:

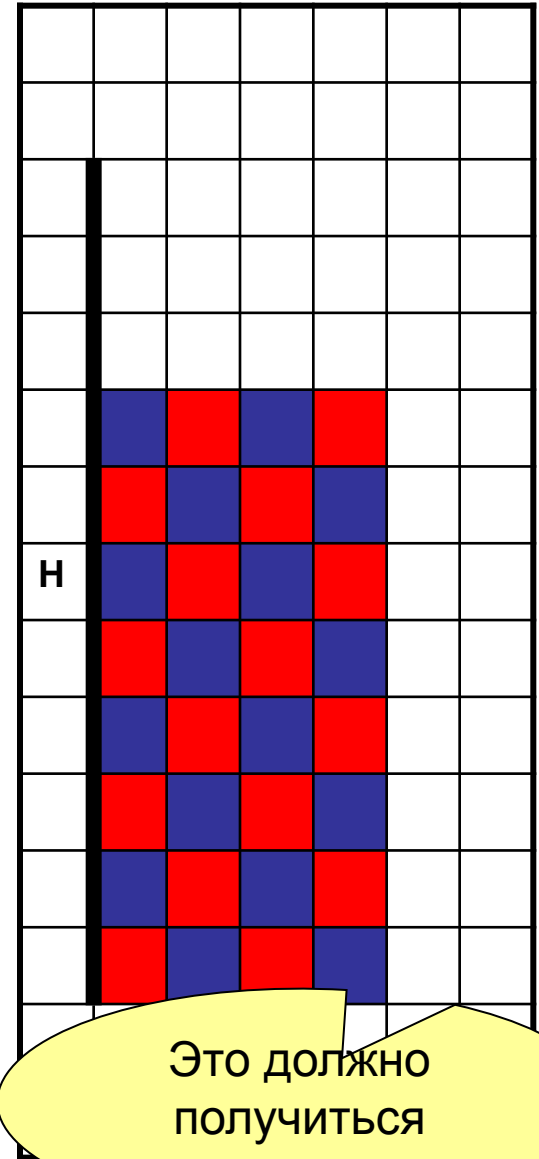
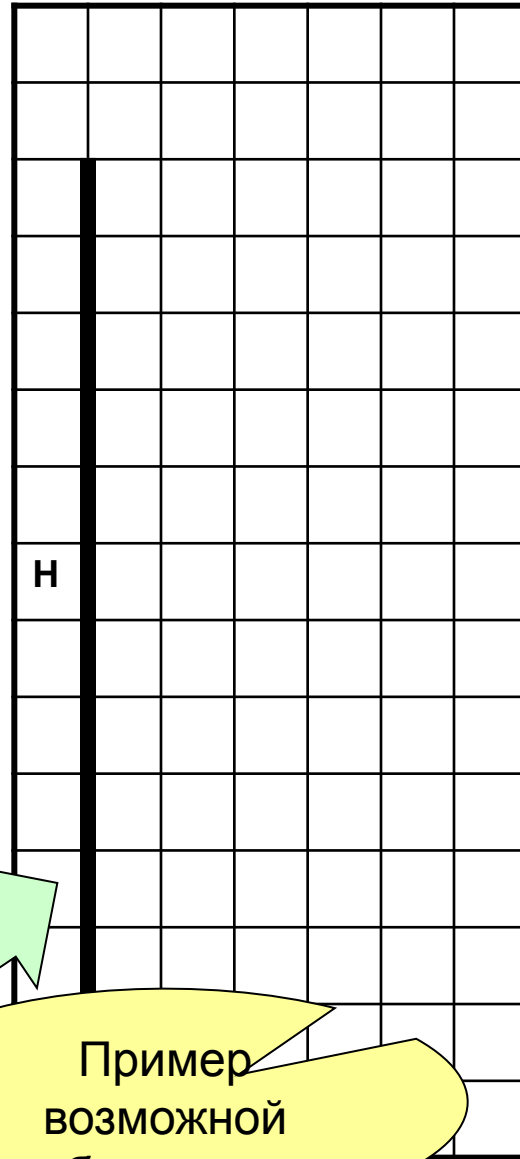
ПОКА <условие> ДЕЛАТЬ <последовательность команд> КОНЕЦ.

Также у ПЛИТОЧНИКА есть две команды:

УЛОЖИТЬ_К и **УЛОЖИТЬ_С**, — которые укладывают красные и синие плитки соответственно.

1) Выполните задание ПЛИТОЧНИК

Справа от бордюра, отступив вниз 3 клетки, нужно уложить 4 ряда красных и синих плиток в шахматном порядке. ПЛИТОЧНИК находится в клеточке слева от бордюра, длина бордюра более 5 клеток, и до конца плоскости он не доходит. Алгоритм записать в текстовом редакторе.



РИСУНКА НЕТ!!!

Творим сами 😊

Пример
возможной
обстановки

Это должно
получиться

Возможный алгоритм ПЛИТОЧНИК

ПОКА НЕ СПРАВА СВОБОДНО ДЕЛАТЬ
ВВЕРХ

КОНЕЦ

ВПРАВО : ВНИЗ : ВНИЗ : ВНИЗ : ВНИЗ

ПОКА НЕ СЛЕВА СВОБОДНО ДЕЛАТЬ

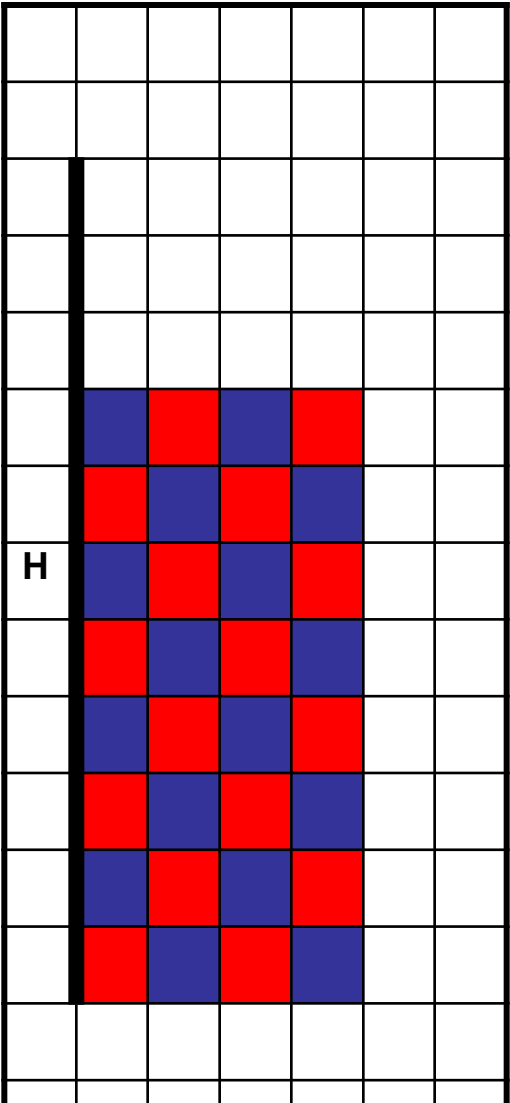
УЛОЖИТЬ_С : ВПРАВО :

УЛОЖИТЬ К : ВПРАВО : УЛОЖИТЬ_С

: ВПРАВО : УЛОЖИТЬ_К

ВНИЗ : ВЛЕВО : ВЛЕВО : ВЛЕВО

ЕСЛИ НЕ С



В сборнике пишут
НЕКОРРЕКТНО:
слева НЕ свободно!!!

КОНЕЦ

Выполните задания

- 2) ПЛИТОЧНИК должен уложить через одну красные и синие плитки снизу от бордюра, который находится на плоскости, пересекая её по горизонтали, не доходя до краёв. Длина бордюра неизвестна. ПЛИТОЧНИК находится в одной из клеток, расположенной непосредственно над бордюром.
- 3) ПЛИТОЧНИК должен уложить через одну красные и синие плитки слева от бордюра, который находится на плоскости, пересекая её по вертикали, не доходя до краёв. Длина бордюра неизвестна. ПЛИТОЧНИК находится в одной из клеток, расположенной непосредственно слева от бордюра. Его начальное положение также неизвестно.

[Плиточник по смыслу похож на ПАРКЕТЧИКА...](#)

[О некоторых средах программирования...](#)

- http://prog.narod.ru/probl_parket.htm - задачи к Паркетчику
- <http://prog.narod.ru/parket.htm>
- **Паркетчик умеет:**
- Перемещаться по клеточному полю.
- Распознавать цвет плитки.
- Убирать и ставить плитки.
- Распознавать наличие стен в соседних клетках.
- Выполнять несложные математические действия с целыми числами.
- **Система команд Паркетчика**
- 1. Шаг вправо
- 2. Шаг влево
- 3. Шаг вверх
- 4. Шаг вниз
- 5. Положить (X) - в текущую клетку ложит плитку цвета X.
- 6. Снять плитку - без комментариев.
- 7. A:=X - переменной A присваивает значение X.
- 8. Запросить A - требует ввести с клавиатуры значение переменной A - пользователь вводит числовое значение и нажимает Enter.
- 9. Сообщить A - выводит на экран значение A.
- 10. Пауза X - приостанавливает выполнение программы на X миллисекунд.
- 11. Перейти на (X,Y) - переходит на ячейку с координатами X,Y.
- 12. Стоп - остановка программы.
- Каждая команда заканчивается знаком (;) - точка с запятой. Группа команд, которая должна выполняться как один оператор, заключается в фигурные скобки: {оператор} - обычно используется в циклах и ветвлениях. Перед закрывающей скобкой } ставить точку с запятой (;) необязательно.
- Начинается описание программы с объявления переменных. Переменные у Паркетчика двух типов: целые и цвет. С переменными целого типа все ясно. Переменные типа цвет могут принимать только два значения: **красная, зеленая**. Если переменных нет, объявление их можно пропустить.
- За переменными идет раздел описания подпрограмм. Если их нет - этот раздел можно так же пропустить. Дальше идет главная часть программы, которая начинается с зарезервированного слова Программа - оно обязательно. Исполнительная часть программы заключается в скобки - {...}.

Плиточник по смыслу похож на ПАРКЕТЧИКА, на которого разработана среда программирования

- <http://yudenisov.livejournal.com/8463.html>
- «Сегодня скачал и опробовал работу некоторых программ для обучения информатики в школе: «Роботландию», «Лого Миры» и «Паркетчик». Первые две распространяются под лицензией «demoware», а последняя — «freeware». Для некоторых программ есть версии под DOS и Windows. Перехожу к более детальному описанию пакетов.
- **Роботландия.** Набор программ для операционной системы MS-DOS, для обучения информатики в младших классах. Будет интересна для школьников с 3 по 6 класс. Содержит в себе тренажёры, обучающие игры и примеры из школьного курса информатики. Интерфейс простой, графика примитивная. В «демонстрационной версии» часть программ не запускается — требуется ключ для разблокировки. Программа запускается в Windows 7 и ALT Linux только в виртуальной машине DOSBox, и уже в нём работает в «оконном режиме» эмулятора. Замечание: демонстрационную версию программы не удалось скачать с официального сайта. Пришлось задействовать поиск в Интернете.
- **Лого Миры.** Набор программ для операционной системы Microsoft Windows, для обучения программированию в средних классах (от 11 лет и старше). Содержит в себе редактор и среду программирования на языке «Лого» (рисование черепашками). Интерфейс мне показался нестандартным, однако есть нормальная инструкция и демонстрационные примеры к программе. Программа распространяется по лицензии «demoware», без покупки ключа невозможно сохранить и распечатать результаты работы. Работает на всех компьютерах с 32-х разрядной Windows и под управлением wine в Linux. Программа выложена для скачивания на сайте разработчика.
- **Паркетчик.** Программа для обучения программированию с примерами из учебника Гейн. Представляет собой реализацию языка: «упрощённый Лого». Программа широко применяется в школе, поэтому её бесплатную («freeware») версию выложили на сайт издательства «Просвещение», откуда автор её и скачал. Интерфейс программы простой и понятный. Программа существует в версии Windows 32 и MS-DOS, работает одинаково хорошо как в Windows, так и на эмуляторах Wine и DOSBox.
- Установка и настройка программ не должна вызвать серьёзных проблем. Что касается приобретения полных версий, тут автор рекомендует сначала протестировать продукты, а потом принимать решения по их покупке. Цена «Роботландии» составляет где-то в районе

23.

И вновь ЧЕРЕПАШКА

- Исполнитель *Черепашка* перемещается на экране компьютера. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существуют две команды:
- **Вперёд n** (где n — целое число), вызывающая передвижение *Черепашки* на n шагов в направлении движения;
- **Налево m** (где m — целое число), вызывающая изменение направления движения на m градусов по часовой стрелке.
- **по** — (перо опустить) выполнение команд, оставляя след в виде линии на экране.
- **пп** — (перо поднять) выполнение команд без следа на экране.
- Запись **Повтори k [Команда1 Команда2 Команда3]** означает, что последовательность команд в скобках повторится k раз.
- Запись на **Листе Программ**. Каждая программа начинается со слова «**Это**», после которого следует имя. На листе программ исполнителя *Черепашка* записывают программы и подпрограммы. (Подпрограмма — это часть программы, содержащая описание определённого набора действий.) Подпрограмму можно вызывать из другой программы. Имя подпрограммы может быть любым. В примере ниже имя подпрограммы — «цифра».

Это цифра

Команда1

Команда2

Конец

Пример вызова подпрограммы «цифра» из программы «число»:

Это число

цифра

Команда1

Конец

В Поле команд записывается окончательная программа.

Выполните задание ЧЕРЕПАШКА

Напишите для *Черепашки* алгоритм, при выполнении которого получится заданная фигура

Листы Программы

это луч

вперед 50

налево 45

повтори 4 [вперед 20 налево 90]

налево 135

вперед 50

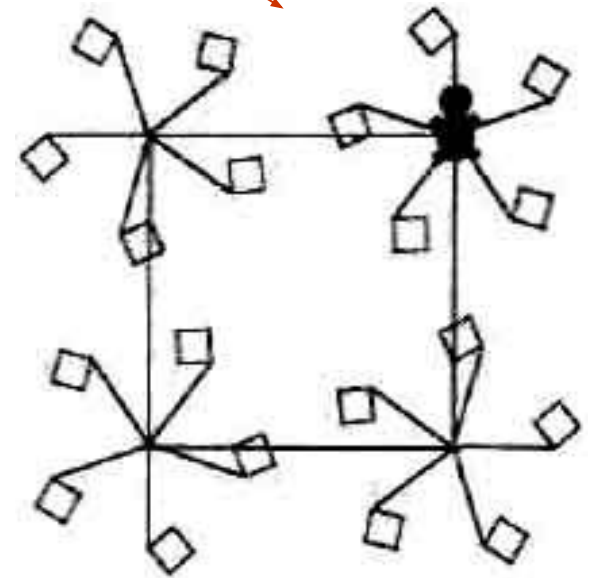
налево 180

налево 72

конец

Поле команд

по повтори 4 [повтори 5 [луч] налево 90 вперед 150]



Чертёжник

- Исполнитель *Чертёжник* умеет перемещаться по листу бумаги. Направление движения *Чертёжника* при необходимости отмечается стрелкой. Ниже приведено описание основных команд *Чертёжника*:
- **Установить_точку**
- **Прыгнуть**
- **Повернуть_налево**
- При выполнении команды **Установить_точку** *Чертёжник* устанавливает точку в том месте, где он находится. При этом данная точка соединяется прямой линией с предыдущей установленной точкой (если она есть на листе).
- При выполнении команды **Прыгнуть** *Чертёжник перемещается на 1 см* по направлению движения, не оставляя за собой линии.
- При выполнении команды **Повернуть_налево** исполнитель разворачивается на 90° против часовой стрелки.
- Команда **впереди_не_край** проверяет истинность условия отсутствия перед *Чертёжником* края бумаги. Эту команду можно использовать с условием «если», имеющим следующий вид:
если <условие>, то:
 последовательность команд
иначе:
 последовательность команд
конец_ветвления
- Последовательность команд — это одна или несколько любых команд *Чертёжника*.
- **Например**, для передвижения на один сантиметр по направлению движения *Чертёжника*, если по ходу его движения нет края листа, можно использовать такой алгоритм:
если впереди_не_край, то:
 Сделать_шаг
конец_ветвления
пока <условие>, повторять:
 последовательность команд
конец_цикла
- **Например**, для рисования прямой линии, пока это возможно, можно использовать следующий алгоритм:
пока впереди_не_край, повторять:
 Сделать_шаг
конец_цикла
- *Выполните задание*
- ***Чертёжник* находится в произвольной точке листа, отстоящей от любого из её краёв более чем на 3 см. Составьте алгоритм, выполнив который он нарисует прямоугольную рамку, отстоящую от края на расстоянии двух шагов.**

Чертёжник

- **Чертёжник** находится в произвольной точке листа, отстоящей от любого из её краёв более чем на 3 см. Составьте алгоритм, выполнив который он нарисует прямоугольную рамку, отстоящую от края на расстоянии двух шагов.

1) Сначала *Чертёжник* может двигаться вперёд (по направлению своего движения, не проводя линии), пока не дойдет до края листа:

пока впереди_не_край, повторять:

Прыгнуть

конец_цикла

2) Теперь *Чертёжник* должен переместиться назад на два шага, чтобы находиться на расстоянии двух шагов до края бумаги:

Повернуть_налево Повернуть_налево Прыгнуть Прыгнуть

3) Развернём *Чертёжника* так, чтобы направление его движения осуществлялось вдоль края листа (при этом дальнейший обход *Чертёжником* листа будет осуществляться по часовой стрелке):

Повернуть_налево

Чертёжник устанавливает точку, которая соединяется прямой с предыдущей точкой (если она есть):

Установить_точку

Далее последовательности команд 1), 2) и 3) следует выполнить 6 раз.

(В результате выполнения этого алгоритма некоторую часть одной из стен *Чертёжника* может закрасить два раза. Чтобы избежать этого, можно сначала установить *Чертёжника* в точку, отстоящую от двух из сторон на расстоянии 2 см, а затем выполнить предложенный алгоритм.

Задание с кратким ответом. Чертёжник

- Исполнитель Чертежник имеет перо, которое можно поднимать, опускать и перемещать. При перемещении опущенного пера за ним остается след в виде прямой линии. У исполнителя существуют следующие команды:
сместиться на вектор (a, b) - исполнитель перемещается в точку, в которую можно попасть из данной, пройдя a единиц по горизонтали и b единиц по вертикали;
Запись Повторить 5 [Команда 1 Команда 2] означает, что последовательность команд в квадратных скобках повторяется 5 раз.
- Чертежник находится в начале координат. Чертежнику дан для исполнения следующий алгоритм:
сместиться на вектор $(5, 2)$
сместиться на вектор $(-3, 3)$
Повторить 3 [Сместиться на вектор $(1, 0)$]
сместиться на вектор $(3, 1)$
На каком расстоянии от начала координат будет находиться исполнитель Чертежник в результате выполнения данного алгоритма?

Источники информации

- **Информатика и ИКТ. 9 класс. Подготовка к ГИА-2011**/Под редакцией Ф.Ф. Лысенко, Л.Н. Евич. – Ростов-на-Дону: Легион-М, 2011. – 272с. – (ГИА-9)
- **Информатика: ГИА: Учебно-справочные материалы для 9 класса (Серия «Итоговый контроль: ГИА»)** / С. М. Авдошин, Р. З. Ахметсафина, О. В. Максименкова, И. Н. Лесовская, М. В. Курак, Н. П. Липкин, С. А. Семикина.— М.; СПб.: Просвещение, 2011.— 252 с: ил.
- **ГИА-2011 : Экзамен в новой форме : Информатика : 9-й кл. : Тренировочные варианты экзаменационных работ для проведения государственной итоговой аттестации в новой форме / авт.-сост. Д.П. Кириенко, П.О. Осипов, А.В. Чернов. — М.: АСТ: Астрель, 2011. — 112,[16] с. — (Федеральный институт педагогических измерений).**
- ДемOVERсии ГИА 2009, 2010 г.г.
- Сайт <http://kpolyakov.narod.ru> Полякова К.Ю.



ГИА-2011

ЭКЗАМЕН В НОВОЙ ФОРМЕ

ИНФОРМАТИКА

Тренировочные варианты
экзаменационных работ
для проведения
государственной
итоговой аттестации
в новой форме

9
КЛАСС

ФЕДЕРАЛЬНЫЙ
ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ
ОФИЦИАЛЬНЫЙ
РАЗРАБОТЧИК КОНТРОЛЬНЫХ ИЗМЕРИТЕЛЬНЫХ МАТЕРИАЛОВ
для ГОСУДАРСТВЕННОЙ ИТОГОВОЙ АТТЕСТАЦИИ

ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ

ГИА-9

Под редакцией Ф.Ф. Лысенко,
Л.Н. Енин

**ИНФОРМАТИКА
и ИКТ**

9 КЛАСС

**ПОДГОТОВКА
К ГИА-9
2011**

Разработано в соответствии
с Федеральным компонентом
государственного стандарта
общего образования

ИТОГОВЫЙ КОНТРОЛЬ

Информатика

Учебно-справочные
материалы
для 9 класса

Экзамен с «Присвоением»