## Исполнитель "Робот"

В этой работе вам предстоит управлять учебным графическим исполнителем "Робот".

Робот живёт на прямоугольном клеточном поле, по периметру огороженном стенами. Между клетками тоже могут встречаться стены.

Возможные состояния клетки:

* пустая;
* помечена для закрашивания;
* закрашена.
* клетка, отмеченная чёрной точкой, куда робот должен прийти после выполнения алгоритма.

### Команды робота

|  |  |
| --- | --- |
| **Команда** | **Описание** |
| move\_left(n=1) | Пройти n клеток влево (по умолчанию n = 1) |
| move\_right(n=1) | Пройти n клеток вправо (по умолчанию n = 1) |
| move\_up(n=1) | Пройти n клеток вверх (по умолчанию n = 1) |
| move\_down(n=1) | Пройти n клеток вниз (по умолчанию n = 1) |
| wall\_is\_above() | если *сверху* стена, возвращает True, иначе — False |
| wall\_is\_beneath() | если *снизу* стена, возвращает True, иначе — False |
| wall\_is\_on\_the\_left() | если *слева* стена, возвращает True, иначе — False |
| wall\_is\_on\_the\_right() | если *справа* стена, возвращает True, иначе — False |
| fill\_cell() | Закрасить текущую клетку |
| cell\_is\_filled() | Возвращает True, если текущая клетка закрашена |
| mov(r, v) | Поместить значение v в регистр r |

### Как запустить Робота

Скачайте и разархивируйте [файлы Робота](http://cs.mipt.ru/python/extra/lab1/robot-tasks-master.zip) в любое место в своей домашней папке.

Увидеть условие первой задачи можно, запустив в этой папке из консоли

python3 task\_1.py

Решение пишется прямо в файл task\_1.py, который можно открыть любым текстовым редактором (например, gedit или PyCharm).

Для быстрой самопроверки ваших решений в этой папке есть файл all\_tasks.py, который запустит все программы подряд и проверит их корректность.

### Как решать задачи для Робота

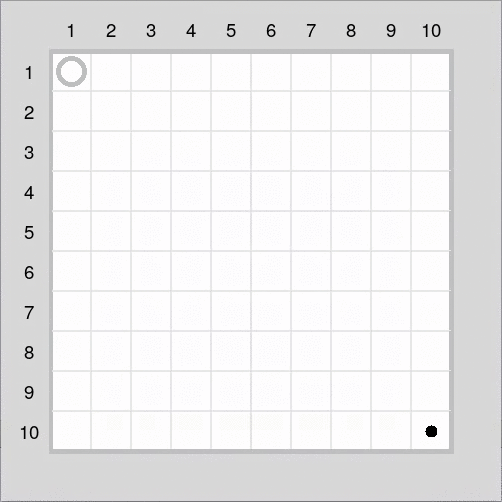
Для решения каждой задачи, если больше ничего не сказано, требуется выполнить условия:

* закончить выполнение в клетке с чёрной точкой;
* закрасить все бирюзовые клетки;
* ни разу не врезаться в стену.

В итоге Робот может оказаться в одном из четырёх состояний.

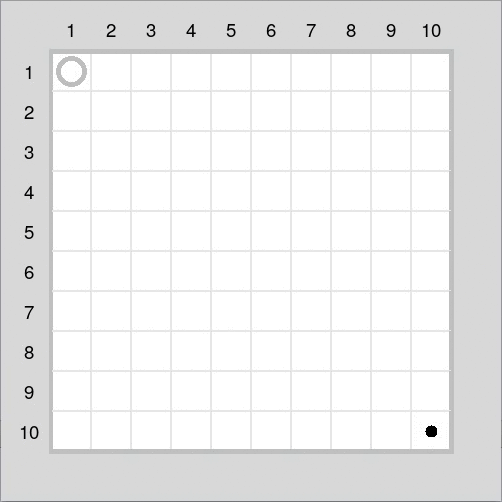
#### **Состояние №1: задача решена**

|  |  |
| --- | --- |
| |  | | --- | | *#!/usr/bin/python3*  **from** pyrob.api **import** \*  @task **def** example1():   **for** i **in** range(9):  move\_right()  move\_down()   **if** \_\_name\_\_ == **'\_\_main\_\_'**:  run\_tasks() | |

qaq

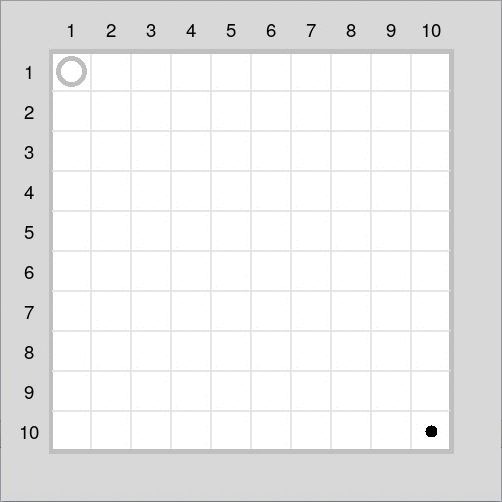
#### **Состояние №2: задача не решена, завершение штатное**

|  |
| --- |
| *#!/usr/bin/python3*  **from** pyrob.api **import** \*   @task **def** example1():   **for** i **in** range(4):  move\_right()  move\_down()   **if** \_\_name\_\_ == **'\_\_main\_\_'**:  run\_tasks() |



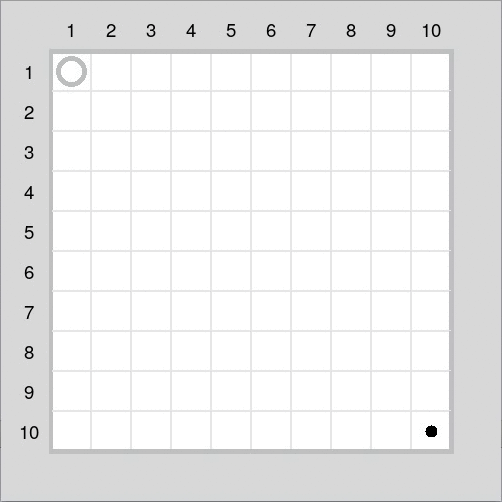
#### **Состояние №3: ошибка времени выполнения**

|  |
| --- |
| *#!/usr/bin/python3*  **from** pyrob.api **import** \*   @task **def** example1():   **for** i **in** range(9):  move\_right()  move\_down()   **if** i == 6:  z = 1 / 0   **if** \_\_name\_\_ == **'\_\_main\_\_'**:  run\_tasks() |



#### **Состояние №4: робот врезался в стену**

|  |
| --- |
| *#!/usr/bin/python3*  **from** pyrob.api **import** \*   @task **def** example1():   **for** i **in** range(10):  move\_right()  move\_down()  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  run\_tasks() |



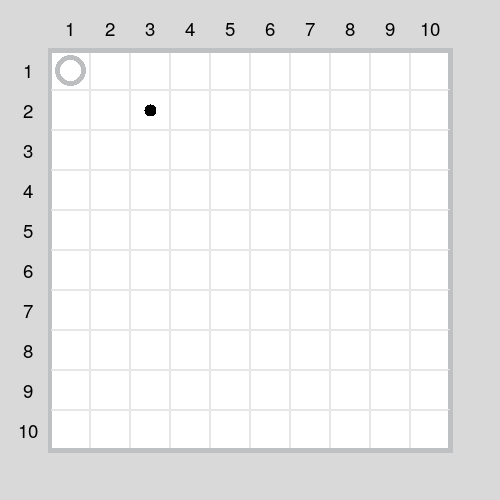
### 

### 

### Просто последовательности команд

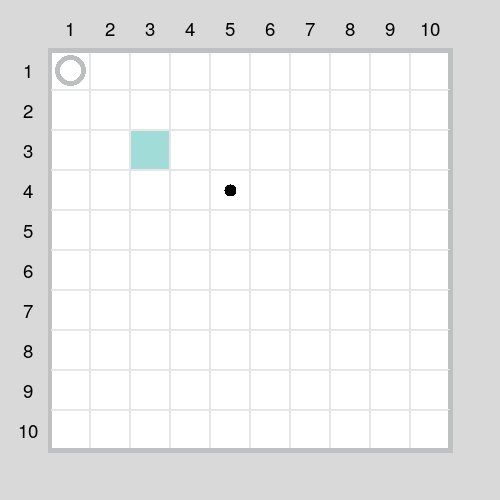
#### Задача №1: task\_1\_1

Дойти до конечной точки.



#### Задача №2: task\_1\_2

Дойти до конечной точки, закрасить одну клетку.



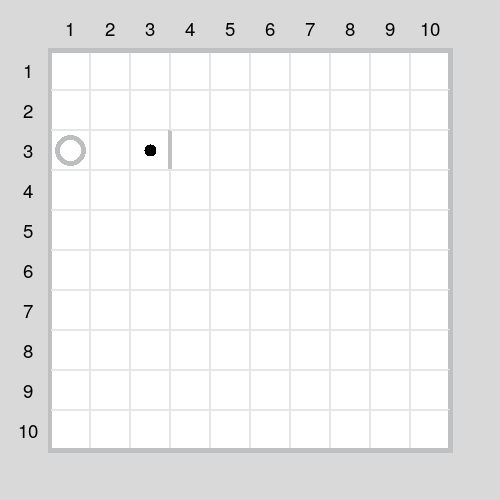
### 

### 

### Условное исполнение

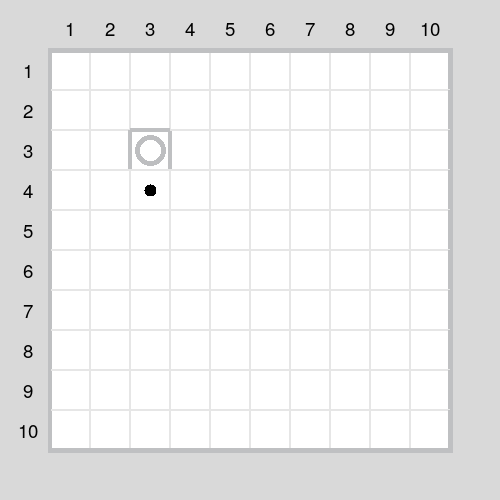
#### Задача №3: task\_3\_1

Дойти до стены. Расстояние до стены не известно.



#### Задача №4: task\_3\_3

С трёх сторон стены. Выйти в свободную сторону. Положение выхода не известно.



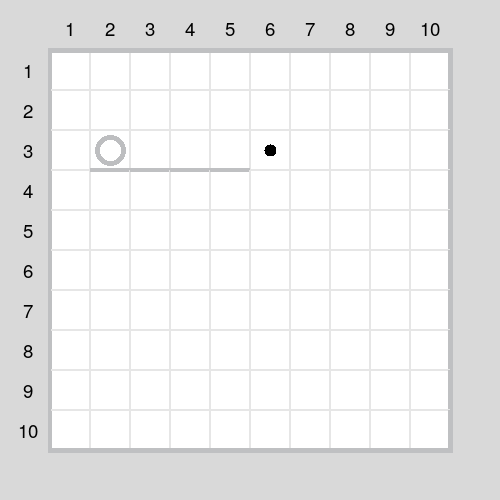
### 

### 

### Циклы с условием

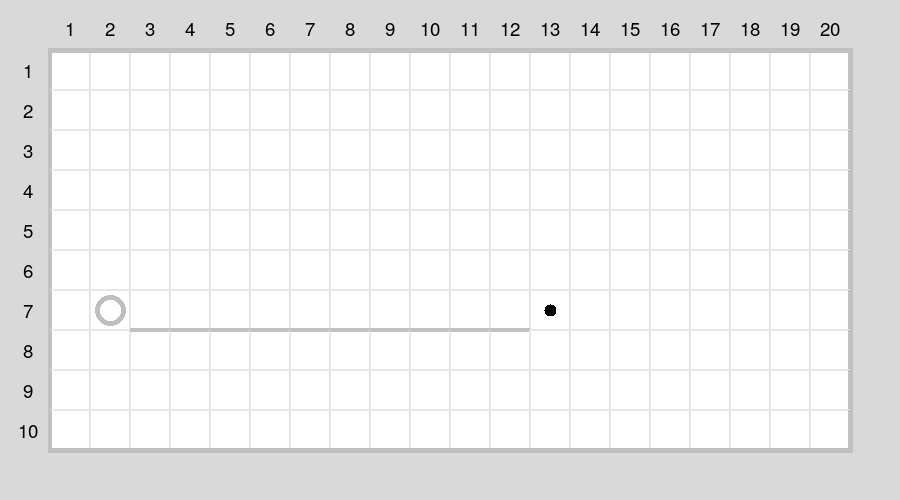
#### Задача №5: task\_5\_2

Дойти до конца стены. Расстояние не известно.



#### Задача №6: task\_5\_3

Дойти до конца стены. Расстояние не известно.

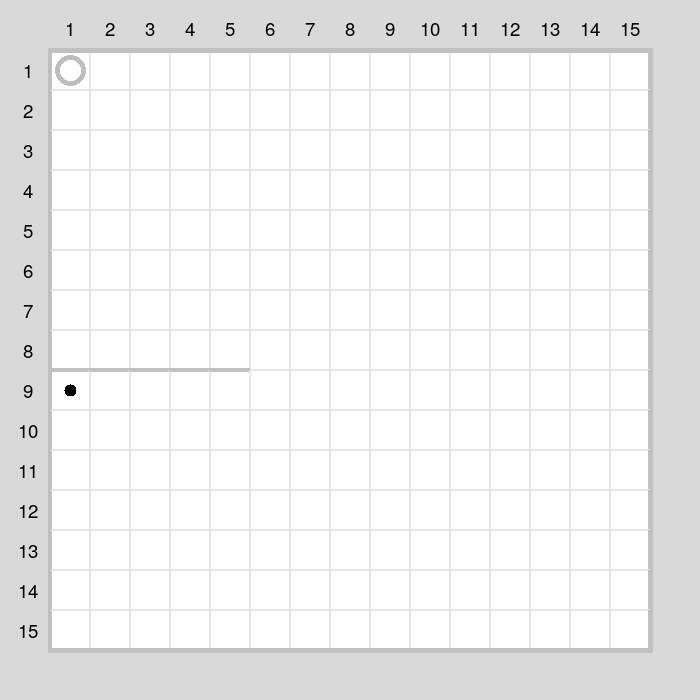


#### 

#### 

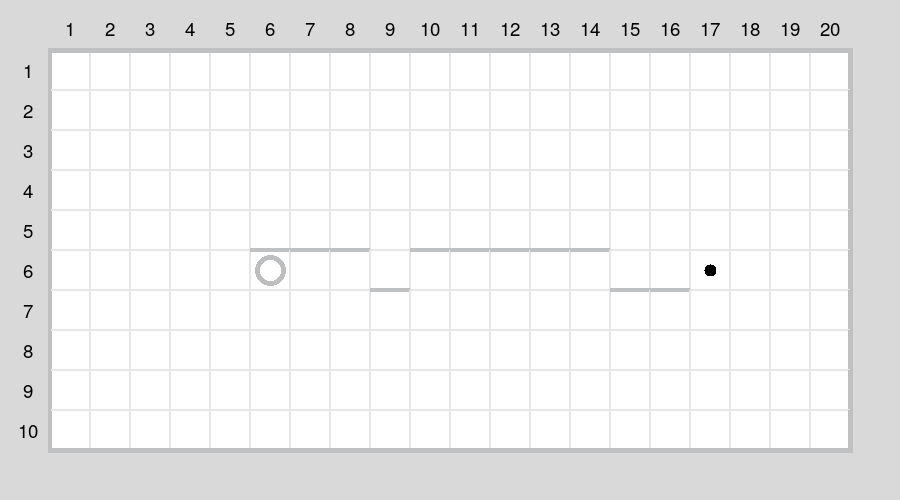
#### Задача №7: task\_5\_4

Обойти стену. Размеры стены и расстояние до неё неизвестны. Стена одна.



#### Задача №8: task\_5\_7

Выйти из коридора. Есть проёмы сверху или снизу.

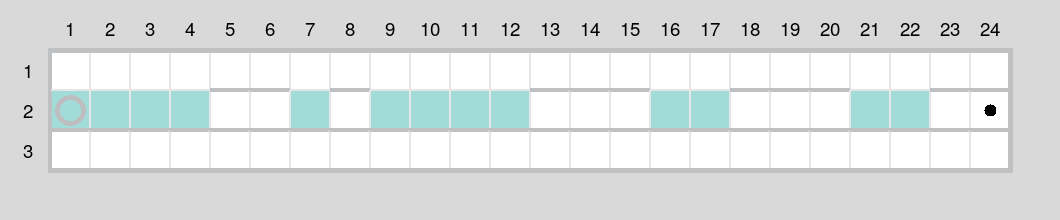


### 

### Условные действия на каждой итерации цикла

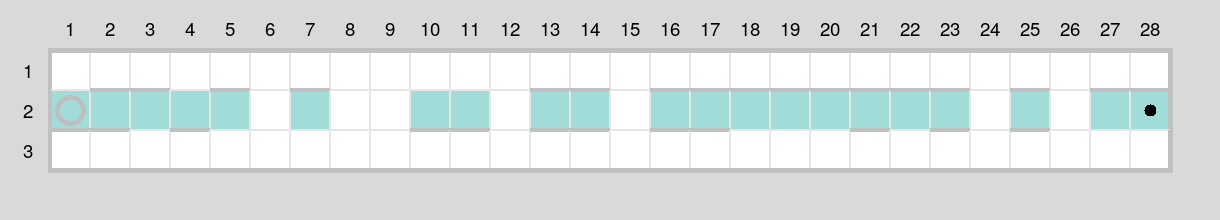
#### Задача №9: task\_8\_2

Закрасить клетки. Расстояние до стены не известно.



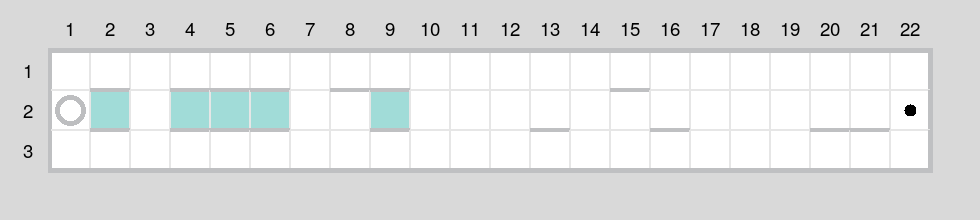
#### Задача №10: task\_8\_3

Закрасить клетки. Расстояние до стены не известно.



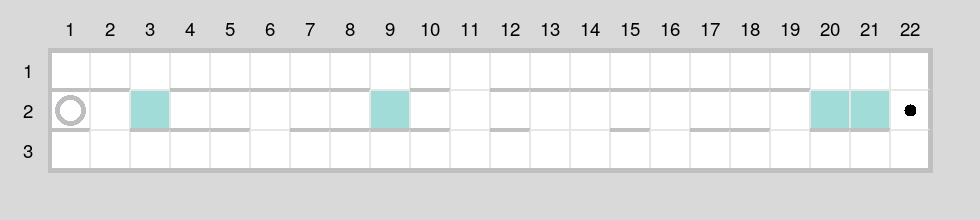
#### Задача №11: task\_8\_4

Закрасить клетки. Расстояние до стены не известно.



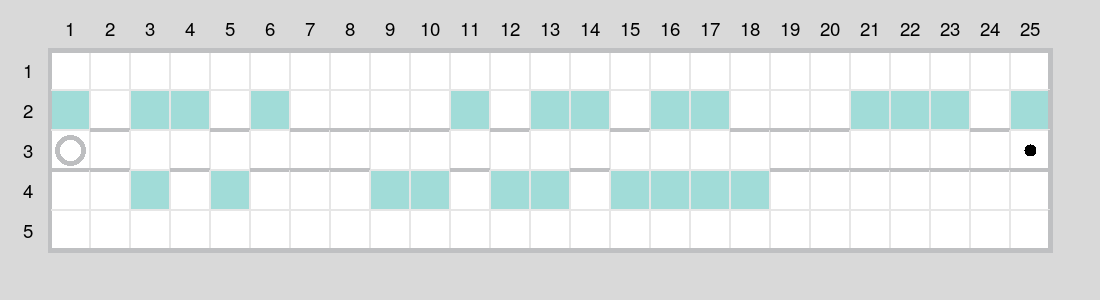
#### Задача №12: task\_8\_6

Закрасить клетки. Расстояние до стены не известно.



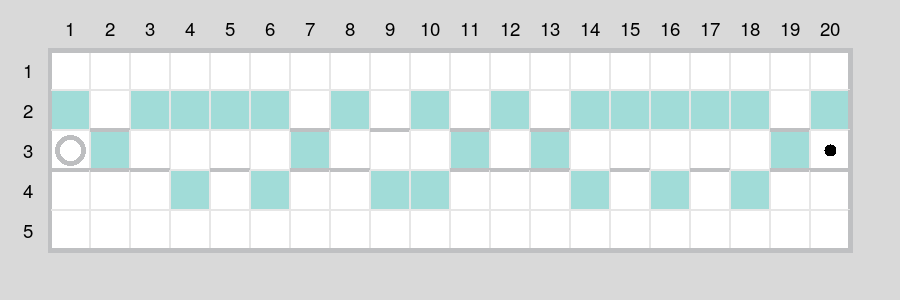
#### Задача №13: task\_8\_10

Закрасить клетки. Расстояние до стены не известно.



#### Задача №14: task\_8\_11

Закрасить клетки. Расстояние до стены не известно.

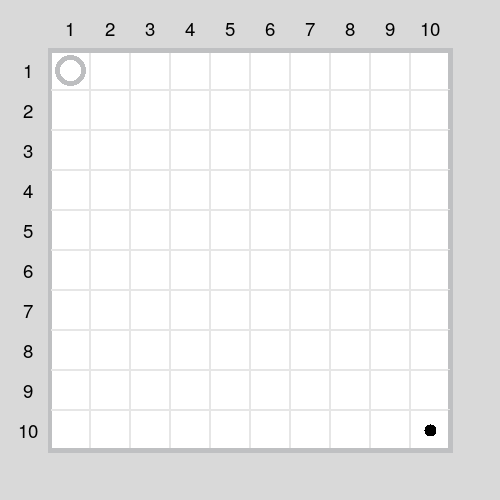


### 

### Циклы, вложенные в условия

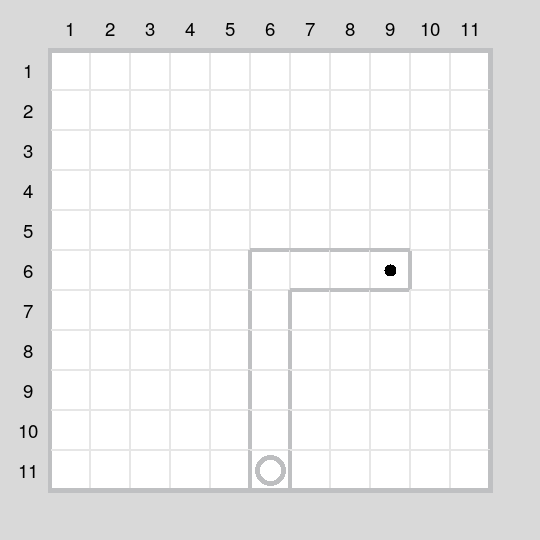
#### Задача №15: task\_8\_21

Перейти в противоположный угол. В начальный момент робот находится в углу, но не известно, в каком.



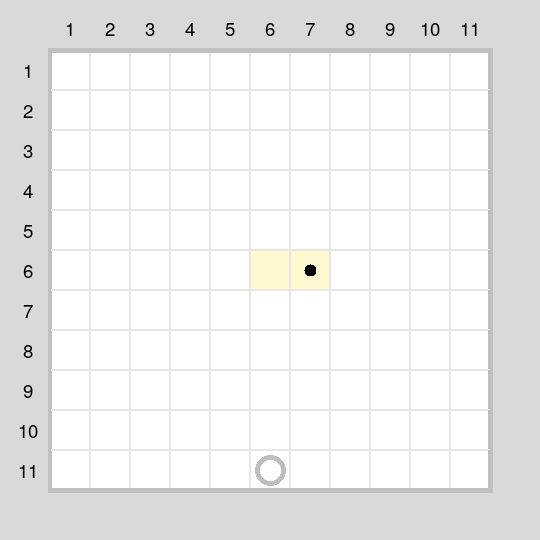
#### Задача №16: task\_8\_22

Дойти до конца тупика. Тупик имеет форму буквы Г (влево или вправо). Размеры тупика не известны.



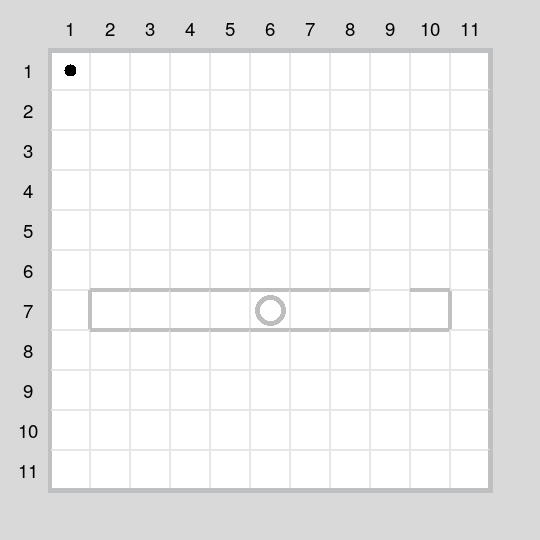
#### Задача №17: task\_8\_27

Перейти на вторую закрашенную клетку. Клетка может быть как справа, так и слева.



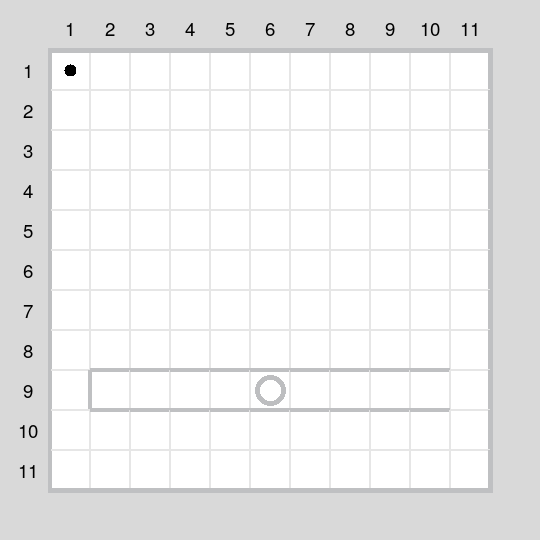
#### Задача №18: task\_8\_28

Выйти из ловушки. Где находится выход, не известно.



#### Задача №19: task\_8\_29

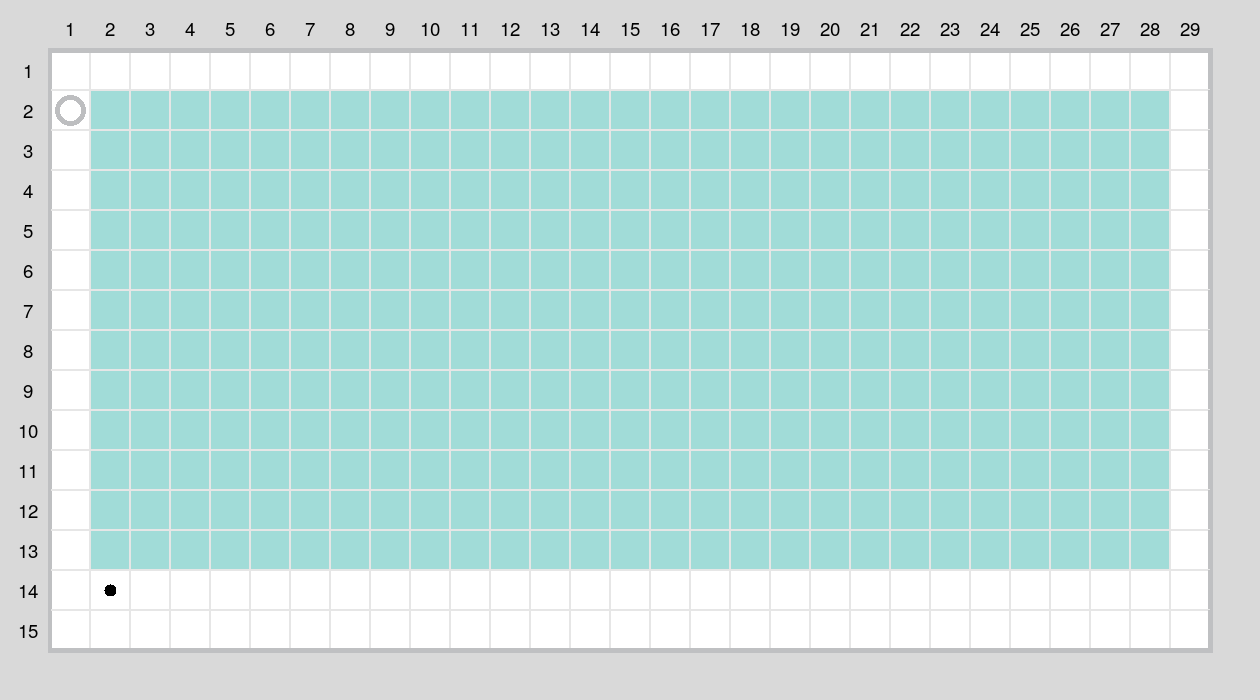
Выйти из ловушки. Выход может находиться как справа, так и слева. Выхода может не быть, в этом случае остановиться в правом тупике.



### Вложенные циклы

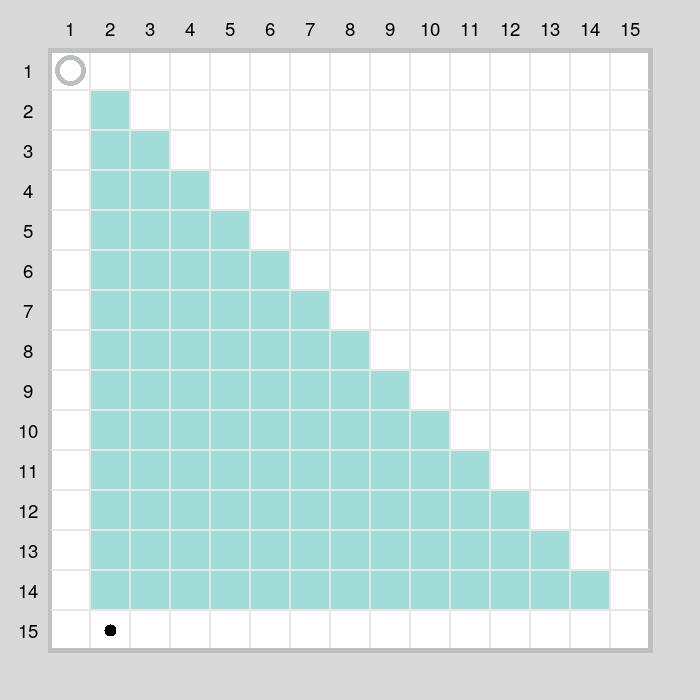
#### Задача №20: task\_4\_3

Закрасить отмеченные клетки.



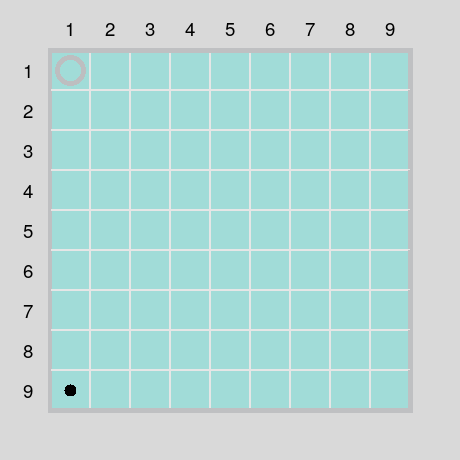
#### Задача №21: task\_4\_11

Закрасить отмеченные клетки.



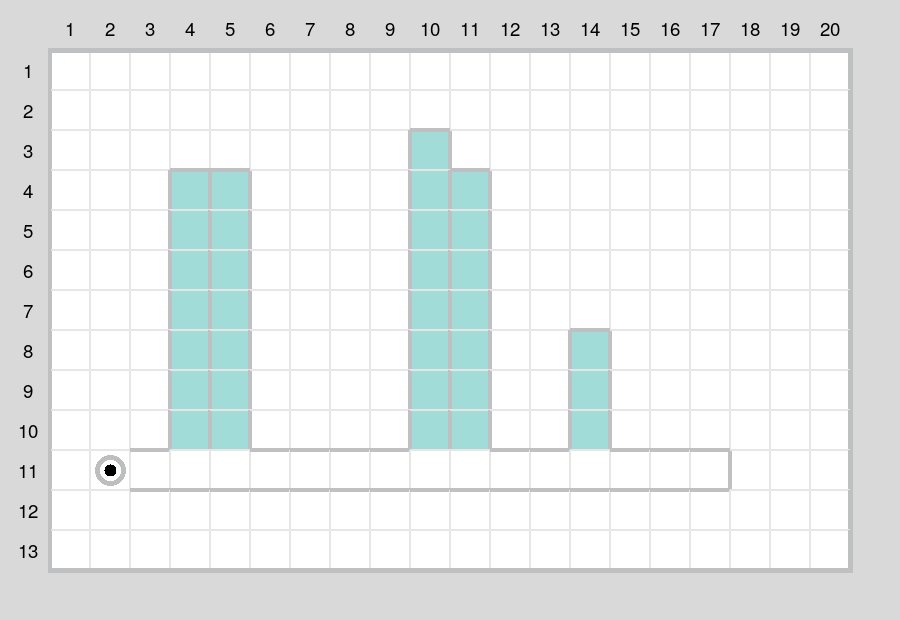
#### Задача №22: task\_5\_10

Закрасить всё поле. Размеры поля неизвестны.



#### Задача №23: task\_6\_6

Закрасить коридоры и вернуться. Количество и длины коридоров не известны.

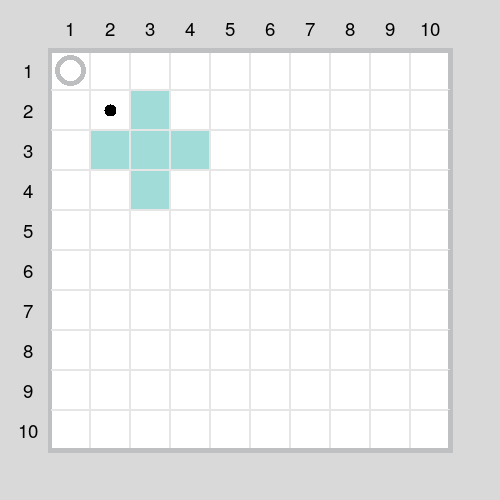


### 

### Создание и использование подпрограмм

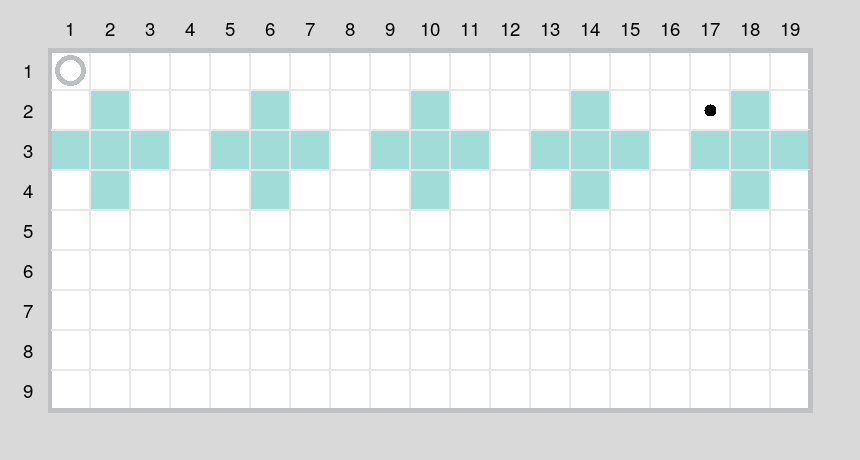
#### Задача №24: task\_2\_1

Закрасить клетки.



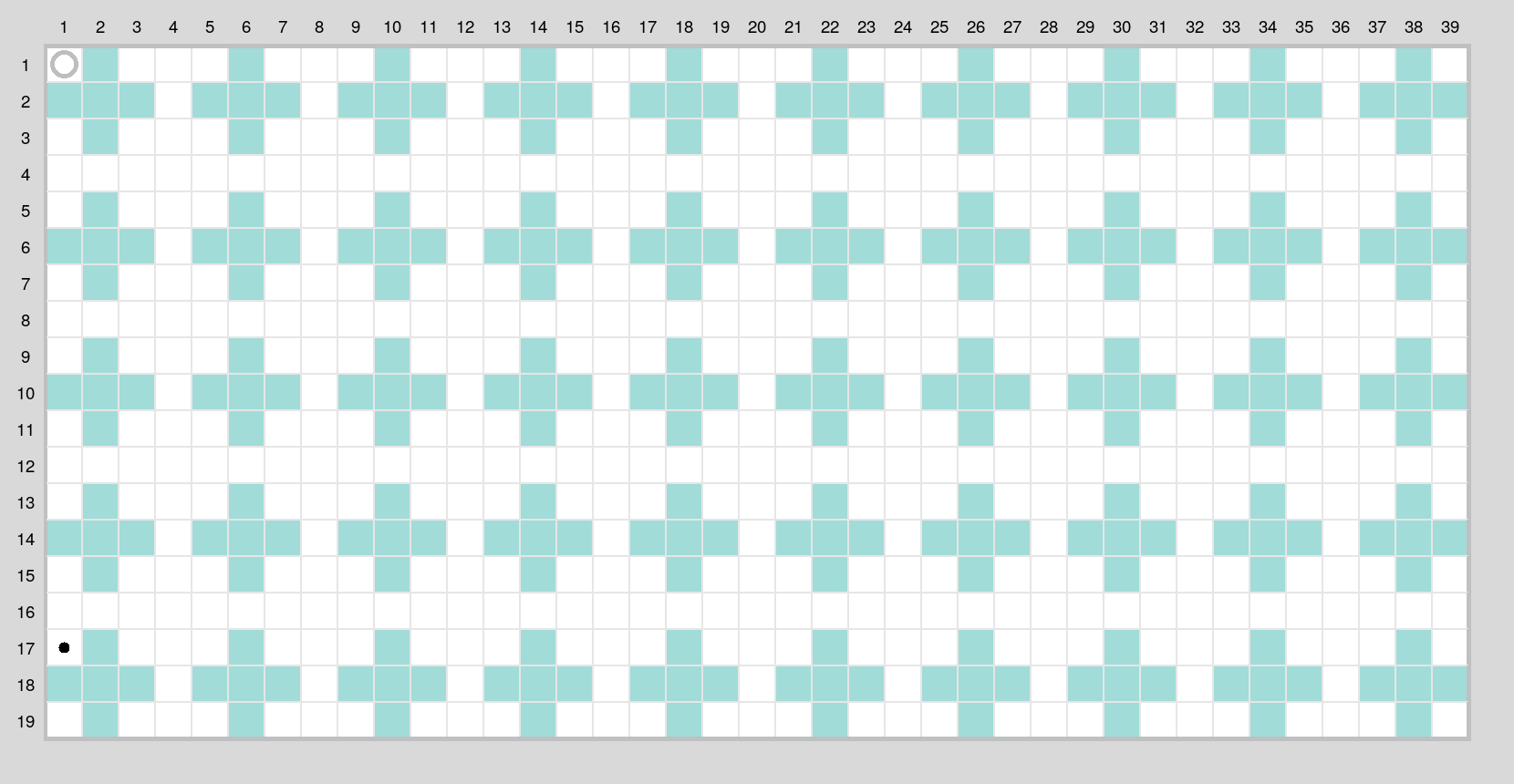
#### Задача №25: task\_2\_2

Закрасить клетки.



#### Задача №26: task\_2\_4

Закрасить клетки.

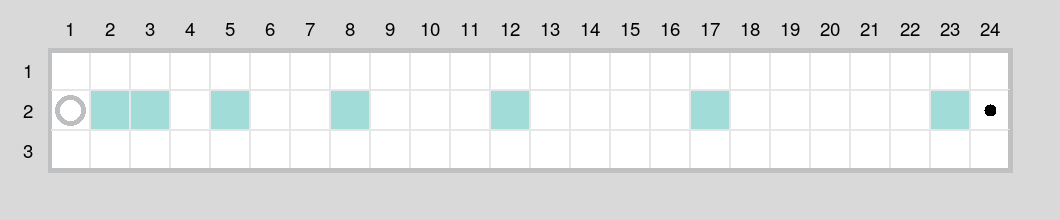


### Использование переменных-флагов и переменных-счётчиков

Обратите внимание на то, что в этих задачах ситуативное поведение Робота не может решить задачу. В задачах требуется запомнить состояние или посчитать количество определённых клеток. Используйте для этого переменные.

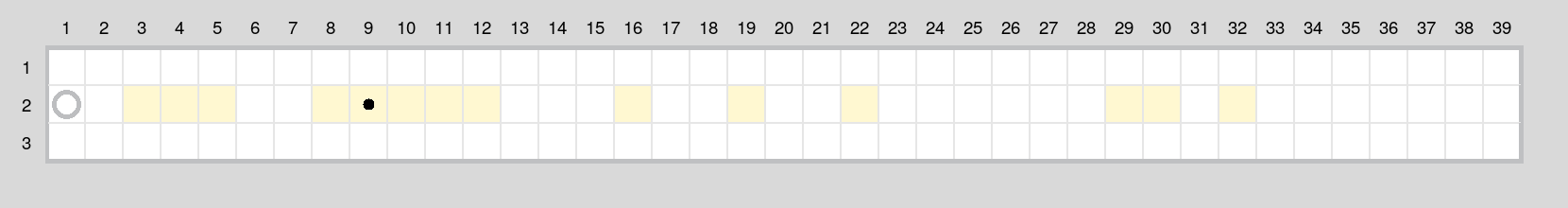
#### Задача №27: task\_7\_5

Закрасить клетки с увеличивающимся интервалом. Расстояние до стены не известно.



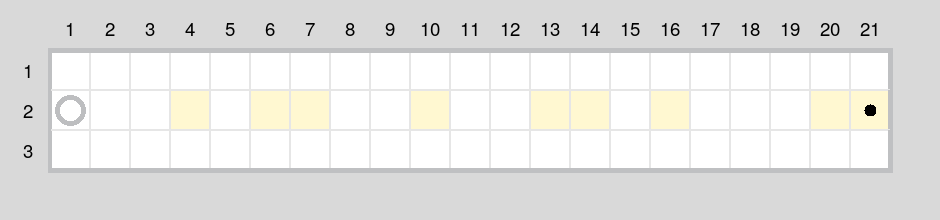
#### Задача №28: task\_7\_6

Остановится на пятой закрашенной клетке. Количество закрашенных клеток не известно, но точно больше пяти.



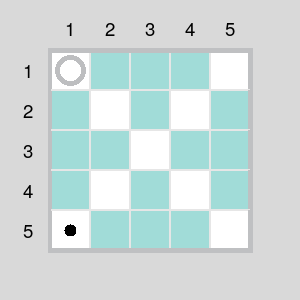
#### Задача №29: task\_7\_7

Остановится на третьей подряд закрашенной клетке. Если нет трёх подряд закрашенных клеток, то остановиться у правой стены. Расстояние до стены не известно.



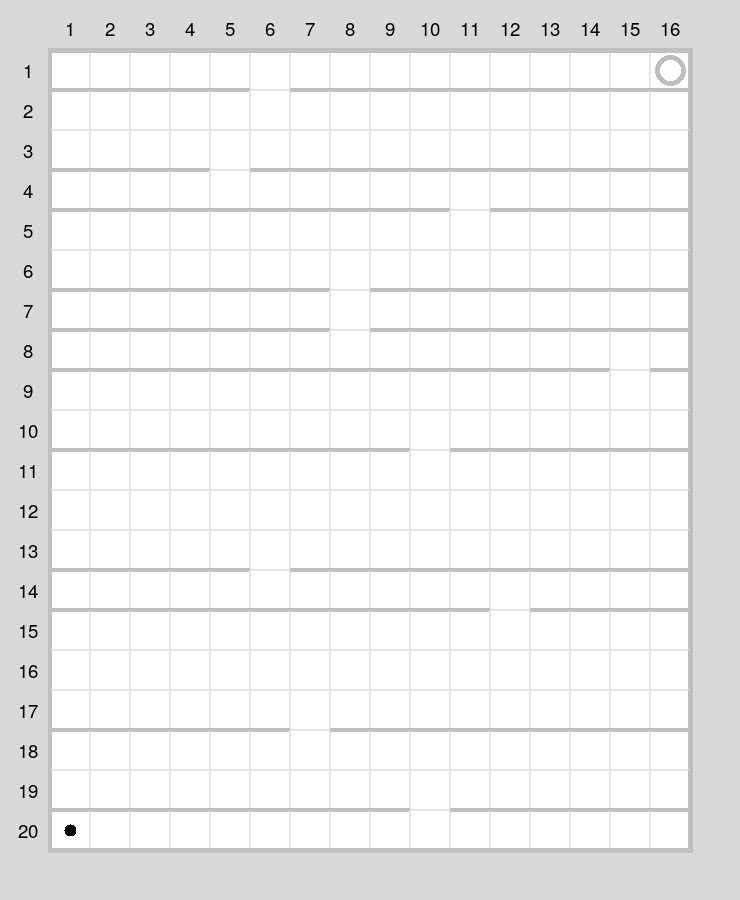
#### Задача №30: task\_9\_3

Закрасить поле «треугольниками». Размер поля не известен, но поле всегда квадратное с нечётным количество клеток по каждой стороне.



#### Задача №31: task\_8\_30

Добраться до нижнего уровня. Количество уровней не известно. Расстояние между стенами не известно. В каждой стене точно есть ровно один проём.



#### Задача №32: task\_8\_18

Закрасить отмеченные клетки. В регистр ax записать количество клеток, которые были закрашены ещё до того, как робот начал двигаться. Количество и размеры коридоров не известны.

