

# 1. ВТОРОЙ ОТБОРОЧНЫЙ ЭТАП

## Задачи второго этапа

### 1.1. Задачи на программирование

*Задача 1.1.1. Задача на перевод координат (- баллов)*

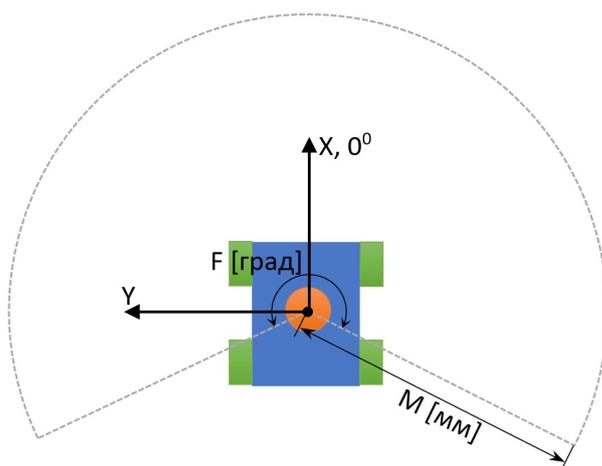


Рис. 1.1: Изображение системы

Робот оборудован LIDAR'ом. Угол его сканирования  $F$  градусов, на эти  $F$  градусов приходится  $P$  измерений LIDAR'а. Максимальное расстояние измерения - до  $M$  миллиметров. Если LIDAR не видит объекта, то соответствующее измерение имеет максимальное значение. Нулевые координаты декартовой системы координат  $OXY$  совпадают с центром LIDAR'а, от которого он измеряет расстояние. Ось  $X$  этой системы координат совпадает с серединой сектора сканирования LIDAR'а. Кроме того, в системе присутствует полярная система координат. Ее полюс совпадает с началом декартовой системы координат и центром лидара. А полярная ось (нулевой луч) совпадает с осью  $X$  декартовой системы координат.

Необходимо перевести координаты каждой найденной точки объекта в декартовы. При этом измерения LIDAR'а, которые не обнаружили объект, переводить в декартовы координаты не требуется.

**Формат входных данных**

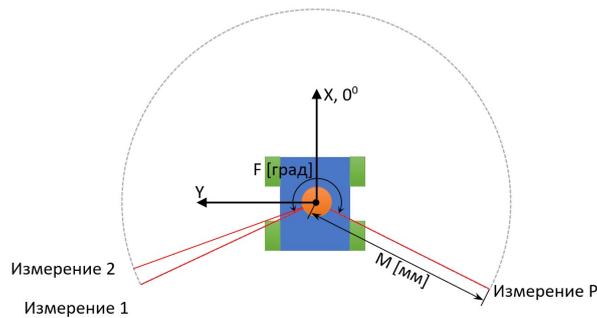


Рис. 1.2: Изображение условия задачи

Первая строка содержит целое число —  $F$  — угол в градусах ( $F \in [1; 360]$ ).

Вторая строка содержит целое число —  $P$  — количество измерений ( $P \in [1; 4096]$ ).

Третья строка содержит целое число —  $M$  — дистанция измерения LIDAR'а в миллиметрах ( $M \in [1; 4000]$ ).

Четвертая строка содержит  $P$  целых чисел - показания LIDAR'а.

### Формат выходных данных

Первая строка содержит целое число - количество измерений LIDAR'а, в которых обнаружен объект. Если ни в одном измерении объекты не обнаружены, то строка содержит число 0. Следующие строки присутствуют только если в первой число больше нуля; они содержат через пробел пары чисел с плавающей точкой - координаты  $X$  и  $Y$  точки обнаруженного объекта с точностью до 0.1 мм. Вторая строка содержит координаты из первого измерения лидара, в котором обнаружен объект, третья - из второго измерения, в котором обнаружен объект и т.д.

### Комментарии

Готовые библиотеки для решения задачи использовать запрещено.

Примеры входных данных представлены по ссылке и не выводятся в самом задании.

### Примеры

Примеры тестовых полигонов представлены по [данной](#) ссылке.

### Решение

#### Задача 1.1.2. Задача на SLAM (- баллов)

Робот оборудован LIDAR'ом. Угол его сканирования  $F$  градусов, на эти  $F$  градусов приходится  $P$  измерений LIDAR'а. Максимальное расстояние измерения - до  $M$  миллиметров. Если LIDAR не видит объекта, то соответствующее измерение имеет максимальное значение. Нулевые координаты декартовой системы координат  $OXY$  совпадают с центром LIDAR'а, от которого он измеряет расстояние. Ось  $X$  этой системы координат совпадает с серединой сектора сканирования LIDAR'а. Кроме того,

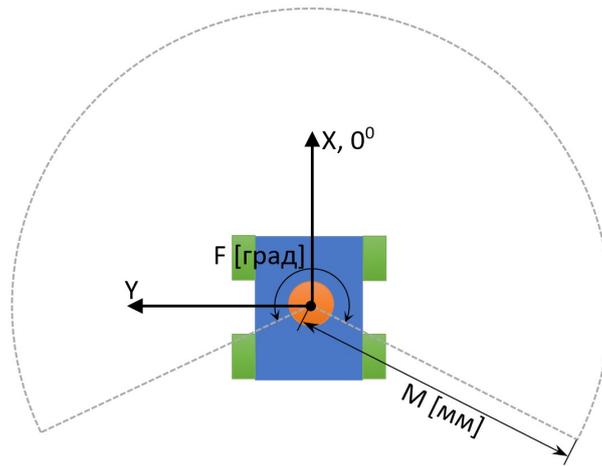


Рис. 1.3: Изображение системы

в системе присутствует полярная система координат. Ее полюс совпадает с началом декартовой системы координат и центром лидара. А полярная ось (нулевой луч) совпадает с осью X декартовой системы координат.

Гарантируется, что в зоне видимости лидара находится одна стена без изгибов или углов. Необходимо определить угол между осью X декартовой системы координат и стеной.

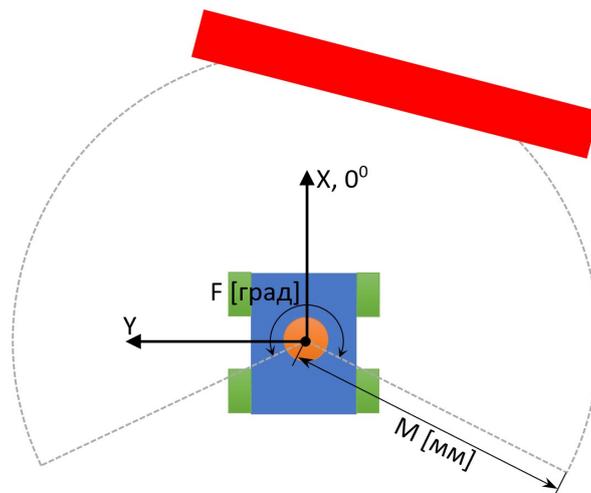


Рис. 1.4: Изображение условия задачи

### Формат входных данных

Первая строка содержит целое число —  $F$  — угол в градусах ( $F \in [1; 360]$ ).

Вторая строка содержит целое число —  $P$  — количество измерений ( $P \in [1; 4096]$ ).

Третья строка содержит целое число —  $M$  — дистанция измерения LIDAR'a в миллиметрах ( $M \in [1; 4000]$ ).

Четвертая строка содержит  $P$  целых чисел - показания LIDAR'a.

### Формат выходных данных

Целое число - угол в градусах между стеной и осью X (направление движения робота).

### Комментарии

Готовые библиотеки для решения задачи использовать запрещено.

Примеры входных данных представлены по ссылке и не выводятся в самом задании.

### Примеры

Примеры тестовых полигонов представлены по [данной](#) ссылке.

### Решение

#### Задача 1.1.3. Задача на оптимизацию работы лидара (- баллов)

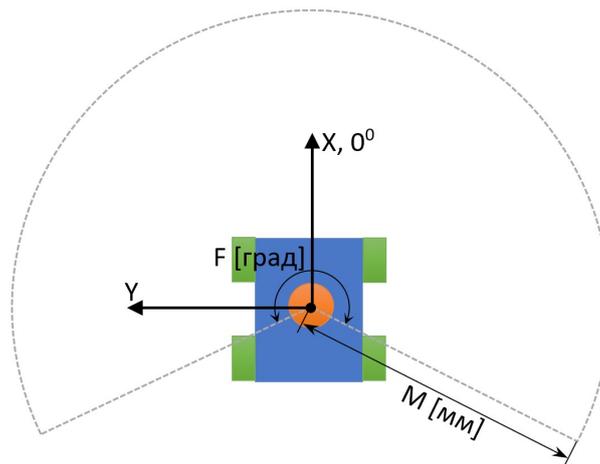


Рис. 1.5: Изображение системы

Робот оборудован LIDAR'ом. Угол его сканирования  $F$  градусов, на эти  $F$  градусов приходится  $P$  измерений LIDAR'а. Максимальное расстояние измерения - до  $M$  миллиметров. Если LIDAR не видит объекта, то соответствующее измерение имеет максимальное значение. Нулевые координаты декартовой системы координат  $OXY$  совпадают с центром LIDAR'а, от которого он измеряет расстояние. Ось  $X$  этой системы координат совпадает с серединой сектора сканирования LIDAR'а. Кроме того, в системе присутствует полярная система координат. Ее полюс совпадает с началом декартовой системы координат и центром лидара. А полярная ось (нулевой луч) совпадает с осью  $X$  декартовой системы координат.

Контроллер робота не успевает обрабатывать все показания LIDAR'а с требуемым быстродействием. Гарантируется, что LIDAR видит один объект из следующего списка: параллелепипед / куб (тип 1), цилиндр (тип 2), прямая стена (тип 3), стена с выпуклым углом (тип 4), стена с вогнутым углом (тип 5). Гарантируется, что объект занимает не менее  $1/10$  от всего числа измерений (от числа  $P$ ).

### Формат входных данных

Первая строка содержит целое число —  $F$  — угол в градусах ( $F \in [1; 360]$ ).

Вторая строка содержит целое число —  $P$  — количество измерений ( $P \in [1; 4096]$ ).

Третья строка содержит целое число —  $M$  — дистанция измерения LIDAR'a в миллиметрах ( $M \in [1; 4000]$ ).

Четвертая строка содержит  $P$  целых чисел - показания LIDAR'a.

### Формат выходных данных

Целое число - тип объекта, который зафиксировал LIDAR.

### Комментарии

Готовые библиотеки для решения задачи использовать запрещено.

Примеры входных данных представлены по ссылке и не выводятся в самом задании.

### Примеры

Примеры тестовых полигонов представлены по [данной](#) ссылке.

### Решение

#### **Задача 1.1.4. Задача на локализацию по маякам и пеленгам (-баллов)**

На полигоне находится три "маяка" с известными координатами. Робот оборудован устройством, способным определить направления на маяки. При этом азимут измеряется относительно направления движения робота вращением по часовой стрелке. То есть, маяк, расположенный ровно по направлению движения робота имеет азимут 0 градусов; маяк, расположенный строго справа от робота - азимут 90 градусов; маяк, расположенный строго слева от робота - азимут 270 градусов (не -90 градусов!).

Даны координаты "маяков" и азимуты на маяки относительно направления движения робота. Необходимо определить координаты робота и его направление. Допустимая погрешность  $\pm 1$  мм и  $\pm 1$  градус.

### Формат входных данных

Первые 3 строки содержат через пробел по два целых числа —  $x_i, y_i$  — координаты маяков в миллиметрах ( $x_i \in [-5000; 5000], y_i \in [-5000; 5000]$ ).

Четвертая строка содержит через пробел три целых числа —  $a_1, a_2, a_3$  — азимуты на маяки относительно направления движения робота ( $a_i \in [0; 359]$ ).

### Формат выходных данных

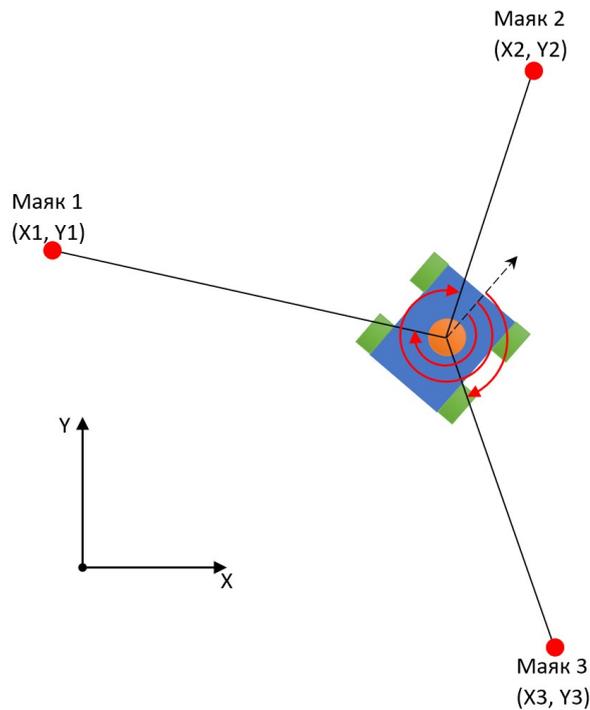


Рис. 1.6: Структура позиций объектов в задаче

Строка, содержащая через пробел три числа: координату  $X$  робота в миллиметрах, координату  $Y$  робота в миллиметрах, угол в градусах между осью  $X$  системы координат и направлением движения робота.

### Комментарии

Готовые библиотеки для решения задачи использовать запрещено.

Примеры входных данных представлены по ссылке и не выводятся в самом задании.

### Примеры

Примеры тестовых полигонов представлены по [данной](#) ссылке.

### Решение

#### ***Задача 1.1.5. Задача на управление мотором по определенному закону (- баллов)***

Робот оборудован колесами диаметром  $N$  мм. Его приводы способны развивать скорость вращения колес до  $M$  об/мин. Робот перевозит хрупкий груз, который разрушается при ускорении или торможении выше  $P$  м/с<sup>2</sup>. Гарантируется, что приводы могут развить ускорение робота выше  $P$  м/с<sup>2</sup>. Расстояние перемещения робота -  $L$  метров.

Необходимо вычислить минимальное время проезда робота  $T$  и скорости робота через каждые 0.1 секунды.

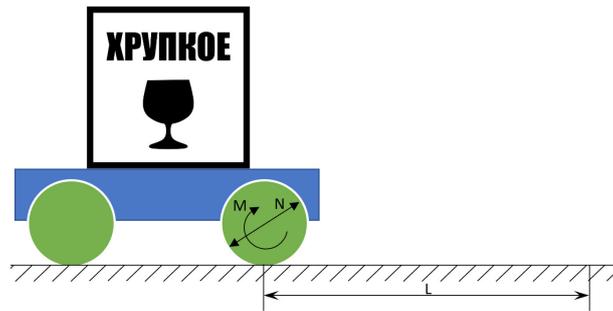


Рис. 1.7: Изображение системы

### Формат входных данных

Первая строка содержит целое число —  $N$  — диаметр колес в миллиметрах ( $N \in [1; 250]$ ).

Вторая строка содержит целое число —  $M$  — максимальная скорость приводов в оборотах в минуту ( $M \in [1; 400]$ ).

Третья строка содержит число с плавающей точкой —  $P$  — максимально допустимое ускорение / торможение, при котором разрушается груз ( $P \in [0.001; 5.0]$ ).

Четвертая строка содержит число с плавающей точкой и двумя знаками после запятой —  $L$  — расстояние, на которое необходимо перевезти груз (в метрах) ( $L \in [0.01; 100.00]$ ).

### Формат выходных данных

$T \cdot 10 + 1$  строк.

Первая строка содержит минимальное время проезда в секундах, с точностью до 0.1 секунды. Следующие  $T \cdot 10$  строк содержат скорости робота в каждый интервал времени в м/с (с точностью до 0.001 м/с). При этом вторая строка содержит скорость в момент старта, то есть число 0 м/с, третья - скорость в момент времени 0.1 с, четвертая - скорость в момент времени 0.2 с и т.д. Последняя строка содержит скорость робота в момент времени  $T$  с, то есть 0 м/с.

### Комментарии

Готовые библиотеки для решения задачи использовать запрещено.

Примеры входных данных представлены по ссылке и не выводятся в самом задании.

### Примеры

Примеры тестовых полигонов представлены по [данной](#) ссылке.

*Решение***Задача 1.1.6. Регулировка автоматизированного поезда (- баллов)**

Автоматизированный поезд с ПИД-контроллером движется вдоль заданного пути

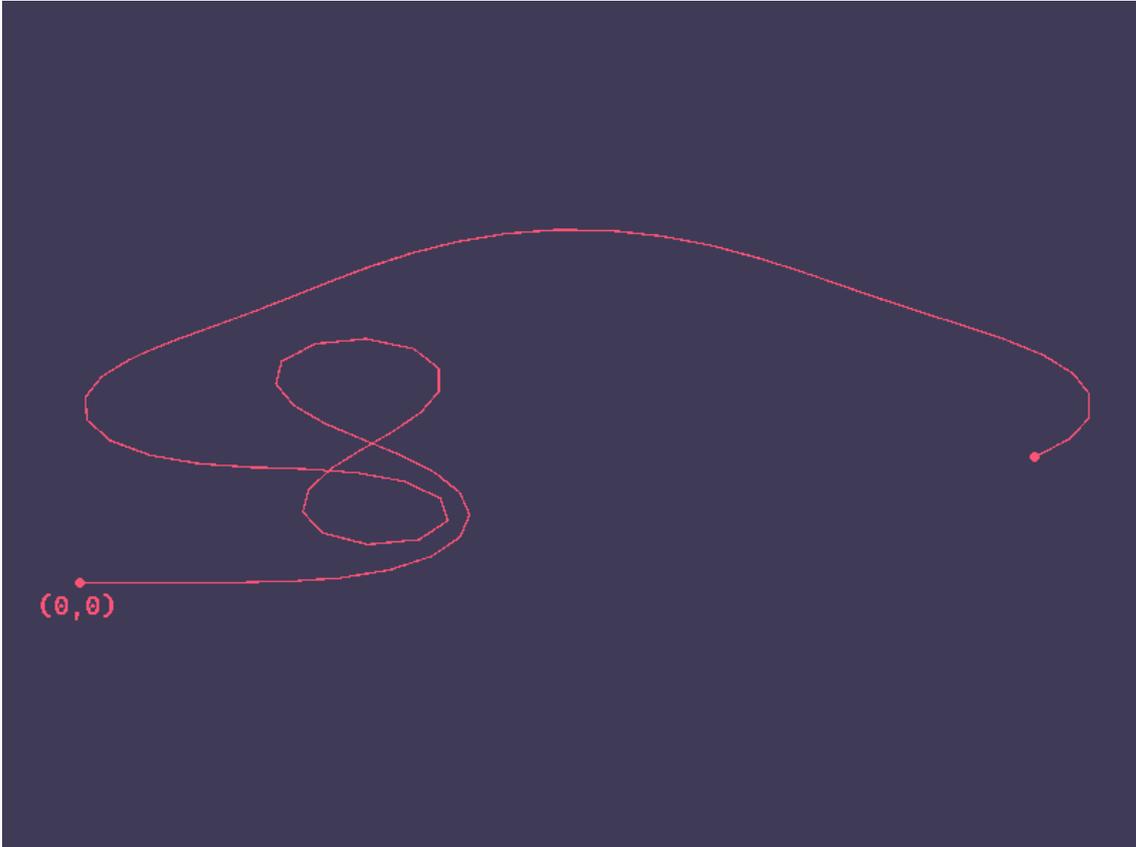


Рис. 1.8: Пример движения поезда

Известен путь движения поезда в виде  $N$  последовательных точек, расстояние между которыми равно  $\Delta x = const = 1$  м.

Изначальная скорость поезда равна  $V$ , масса поезда равна  $M$ . Движение между соседними точками можно считать линейно-направленным и равноускоренным.

Ускорение поезда происходит контролируемо с коллинеарной направлению движения силой  $\vec{F}$  ( $0 \leq |\vec{F}| \leq F_{max}$ ).

Для упрощения считается, что других сил на поезд не оказывается, и изменение направления движения поезда не влияет на модуль его скорости (трение отсутствует). Изменение  $\vec{F}$  применяется ПИД-контроллером мгновенно.

При движении поезд сходит с рельс, если его модуль скорости  $|\vec{V}|$  превышает максимальную скорость для текущего участка  $V_{max0}$ , задаваемую следующим образом:

$$V_{max0} = V_{max} * 2^{-(1 - \cos\alpha)} * V_{factor} \quad (1.1)$$

$$V_{factor} = const = 500 \quad (1.2)$$

Где  $V_{max}$  - максимальная скорость для прямого участка пути ( $\alpha = 0$ ).

Угол  $\alpha$  является углом между текущим ( $\vec{BC}$ ) и предыдущим ( $\vec{AB}$ ) направлениями движения поезда ( $P$ ) на сегментах пути:

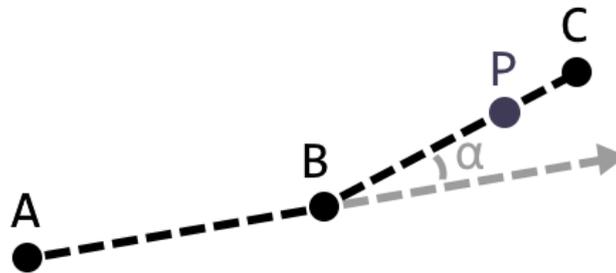


Рис. 1.9: Угол между сегментами

Движение поезда происходит с автоматизированным регулированием [стандартного ПИД-контроллера](#), коэффициенты  $C_P$ ,  $C_I$ ,  $C_D$  которого известны.

Ошибка контроллера  $E_0$  в момент времени равна:

$$E_0 = \frac{3}{4} * V_{max0} - V_0 \quad (1.3)$$

Где  $V_0$  - текущая скорость поезда.  $E_0$  напрямую численно передается для регулирования  $\vec{F}$  в ПИД-контроллер:

$$C_{out} = C_P * e(t) + C_I * \int e(t) * dt + C_D * \frac{d}{dt} e(t) \quad (1.4)$$

ПИД-контроллер производит все обновления с частотой  $FREQ = const = 1000$ Гц.

### Формат входных данных

Первая строка содержит 1 целое число и 4 вещественных числа соответственно через пробел:  $N, V, F_{max}, M, P_{max}$ , где:

- $N$  — количество точек пути поезда ( $0 < N < 10^4$ );
- $V$  — начальная скорость поезда в м/с ( $0 < V < 10^2$ );
- $F_{max}$  — модуль максимально применимой ПИД-контроллером силой в ньютонах ( $1 < F_{max} < 10^3$ );
- $M$  — масса поезда в килограммах ( $1 < M < 10^3$ );
- $P_{max}$  — максимальный импульс поезда ( $P_{max} = M * V_{max}$ ) для движения по прямой ( $\alpha = 0$ ) в кг\*м/с ( $1 < P_{max} < 10^4$ ).

Вторая строка содержит 3 вещественных числа соответственно через пробел:  $C_P, C_I, C_D$ , где:

- $C_P$  — пропорциональная составляющая ПИД-контроллера ( $0 \leq C_P < 10^2$ );
- $C_I$  — интегрирующая составляющая ПИД-контроллера ( $0 \leq C_I < 10^2$ );
- $C_D$  — дифференцирующая составляющая ПИД-контроллера ( $0 \leq C_D < 10^2$ );

Далее идут  $N$  строк, состоящие из 2 вещественных числа через пробел:  $X, Y$ :

- $X$  — координата пути по оси  $X$  в метрах ( $-10^6 < X < 10^6$ );
- $Y$  — координата пути по оси  $Y$  в метрах ( $-10^6 < Y < 10^6$ ).

### Формат выходных данных

Первая строка — единственное слово *DONE*, если симуляция произведена успешно. Слово *FAIL*, если происходит нарушение указанных требований к скорости поезда;

Вторая строка — одно вещественное число с точностью до сотых — время  $T$  в секундах в рамках допустимой ошибки, за которое поезд пройдет указанный или доступный путь:

$$T_{correct} * 0.95 \leq T \leq T_{correct} * 1.05 \quad (1.5)$$

### Примеры

Примеры тестовых путей движения поезда представлены по [данной](#) ссылке.

Предоставленные тесты делятся на пять категорий по мере повышения сложности:

- Для тестов [1; 10] гарантируется, что  $V \gg 0, C_P, C_I, C_D = 0, \forall \alpha (\alpha = const = 0)$ .
- Для тестов [11; 30] гарантируется, что  $V \gg 0, C_P, C_I, C_D = 0$ .
- Для тестов [31; 40] гарантируется, что  $C_I, C_D = 0$ .
- Для тестов [41; 50] гарантируется, что  $C_I = 0$ .
- Для тестов [51; 100] дополнительных условий не предусмотрено.

*Решение*

**Задача 1.1.7. Задача на компьютерное зрение (- баллов)**

Есть изображение с высотой  $H$  пикселей и шириной  $W$  пикселей.

Необходимо вывести наиболее встречающийся в картинке цвет из списка (RED, YELLOW, GREEN, BLUE, PINK, WHITE, BLACK)

Схема перевода из наиболее встречающегося цвета в формате HSV в текстовый вариант можно найти по [данной](#) ссылке. Файл содержит 18 строк, каждая из которых представляет собой границы H S V от начального значения включительно и до конечного включительно, которые нужно проверить по порядку. Первое попадание в границы HSV из файла и будет обозначать конечный цвет.

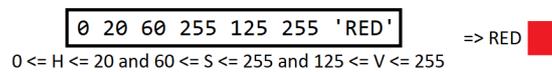


Рис. 1.10: Пример условия для перевода из HSV в текст

H[Hue] - ( $H \in [0; 359]$ ) S[Saturation] - ( $S \in [0; 255]$ ) V[Value] - ( $V \in [0; 255]$ )

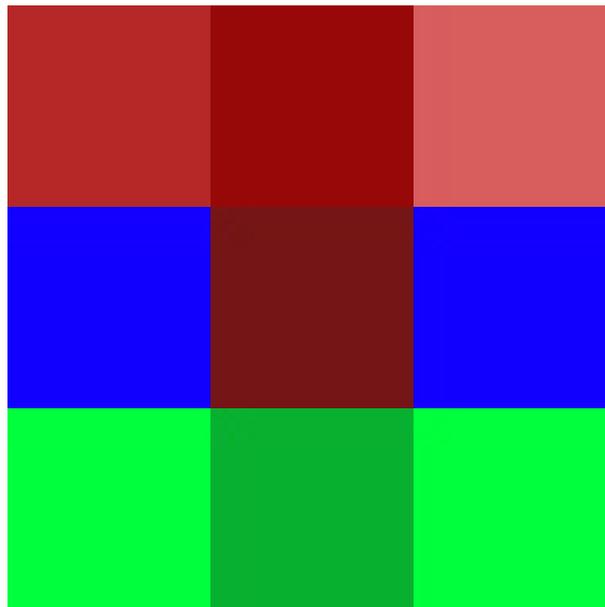


Рис. 1.11: Пример простого изображения

В данном изображении наиболее встречающийся цвет красный(RED), а не зеленый(GREEN) или синий(BLUE), в которых есть по 2 пикселя одинакового цвета

**Формат входных данных**

Первая строка содержит 2 целых числа —  $H$  и  $W$  — высота и ширина изображения в пикселях ( $H \in [50; 4000]$ ,  $W \in [50; 4000]$ ).

Вторая строка содержит через пробел  $H$  на  $W$  кодов цвета в шестнадцатеричной системе счисления. Код выглядит как  $RRGGBB$ , где  $RR$  это шестнадцатеричное число описывающее значение красного канала в пикселе (при переводе в десятичную)

оно принимает значения  $[0, 255]$ ),  $GG$  аналогично описывает зеленый канал в пикселе, а  $BB$  описывает синий.

### **Формат выходных данных**

Одно слово - наиболее встречающийся цвет из списка:

(RED, YELLOW, GREEN, BLUE, PINK, WHITE, BLACK)

### **Комментарии**

Готовые библиотеки для решения задачи использовать запрещено.

Примеры входных данных представлены по ссылке и не выводятся в самом задании.

В данной задаче вам специально дано 5 открытых картинок с ответами, чтобы вы смогли довести алгоритм до идеала. Алгоритм нужно оптимизировать на основе этих изображений.

### ***Примеры***

Примеры тестовых полигонов представлены по [данной](#) ссылке.

### ***Решение***