1. Сначала пользователю предлагается ввести количество (palitra) оттенков красного, зеленого и синего цвета, участвующие в палитре.
2. После ввода происходит создание палитры map – матрицы Nx3, где N – общее количество цветов в палитре. Палитра map образуется из всевозможных сочетаний palitra оттенков красного, зеленого и синего цвета. Сами интенсивности оттенков распределены в отрезке [0, 256] c шагом, равным 256/(palitra – 1) .  
   Например, если ввести число 2, то образуются все возможные сочетания из 2 оттенков красного цвета, 2 оттенков зеленого цвета и 2 оттенков синего цвета. Интенсивность оттенков каждого цвета будет иметь значение 0 или 256. Сама палитра будет выглядеть таким образом:
3. Затем с помощью функции zeros создаются пустые массивы, в которые будут заноситься данные:

1) images – матрица, где каждая строка характеризует соотвествующее изображение. Размер матрицы составляет N\*10000, где N – количество изображений. Строка хранит номер цвета палитры, которым закрашен определенный пиксель. Сам номер цвета берется из матрицы map как номер строки.   
Например, элементы матрицы images принимают значения (всего 10000 элементов). Также дана палитра map, расположенная выше.

Данное условие можно трактовать следующим образом:

- пиксель, лежащий на координате (0, 0) закрашен в цвет палитры №4, то есть в синий цвет, так как 4 строка матрицы map имеет значения (0 0 1), соответствующая (0, 0, 256) по цветовой модели RGB;

- пиксель, лежащий на координате (0, 1) закрашен в цвет палитры №7, то есть в белый цвет, так как 7 строка матрицы map имеет значения (1 1 1), соответствующая (256, 256, 256) по цветовой модели RGB и т. д.

2) notpreparedimages – трехмерный массив. Первая координата отвечает за порядковый номер изображения, вторая и третья – координаты пикселя изображения разрешением 100\*100.  
Значения матрицы хранят номер цвета пикселя определенного изображения из палитры map .

1. Работа с каждым эталонным изображением.
2. Читаем из файла полноцветное квадратное изображение и переводим его в массив.
3. Изменяем размер полученного изображения (массива) на размер 100\*100 пикселей, используя в качестве метода масштабирования Ближайшего соседа.
4. Преобразуем Полноцветное изображение в палитровое, используя предопределенную палитру map.
5. Выводим полученное изображение в папку, чтобы убедиться, что изображение стало палитровым с размером 100\*100.
6. Сохраняем изображение в соответствующую строку массива notpreparedimages .
7. Переводим массив в вектор-строку размером 10000 и сохраняем в в соответствующую строку массива images.
8. Описываем эталонные портреты писателей для последующего вывода, если тестовое изображение будет распознано.  
     
   Пример описания:  
   answers{3} = '3. Брюсов';
9. Работа с тестовым изображением.
10. Читаем из файла полноцветное изображение и переводим его в массив.
11. Изменяем размер полученного изображения (массива) на размер 100\*100 пикселей, используя в качестве метода масштабирования Ближайшего соседа.
12. Преобразуем полноцветное изображение в палитровое, сохранив саму палитру в свою матрицу map.
13. Переводим массив в вектор-строку размером 10000 и сохраняем ее в соответствующую строку массива images.
14. Вычисляем степень соответствия эталонных изображений тестовому посредством подсчета совпадающих по цвету пикселей.  
    Берем каждую строку массива images и сравниваем значения каждого элемента массива с соответствующим элементом вектора testimage. В случае совпадения значений увеличиваем счетчик совпадающих по цвету пикселей у определенного эталонного изображения.
15. Находим среди эталонных изображений то, которая максимально похожа на тестовое (значение счетчика совпадающих по цвету пикселей является наибольшим).
16. Назначаем порог схожести, позволяющий определить, было ли распознано тестовое изображение или нет.
17. Если порог превышен, то выводим в консоли соответствующее описание портрета писателя, в окне выводим палитровое изображение.
18. Иначе в консоли выводим «не удалось распознать».