1. Загрузите данные **ex6data1.mat** из файла.

```
load('ex7data2.mat');
```

2. Реализуйте функцию случайной инициализации K центров кластеров.

```
function centroids = kMeansInitCentroids(X, K)

centroids = zeros(K, size(X, 2));

randidx = randperm(size(X, 1));
centroids = X(randidx(1:K), :);


end
```

3. Реализуйте функцию определения принадлежности к кластерам.

```
function idx = findClosestCentroids(X, centroids)
K = size(centroids, 1);

idx = zeros(size(X,1), 1);

m = size(X,1);

for i = 1:m
distance_array = zeros(1,K);
for j = 1:K
distance_array(1,j) = sqrt(sum(power((X(i,:)-centroids(j,:)),2)));
end
[~, d_idx] = min(distance_array);
idx(i,1) = d_idx;
end

end
```

4. Реализуйте функцию пересчета центров кластеров.

```
function centroids = computeCentroids(X, idx, K)

[m n] = size(X);

centroids = zeros(K, n);

for k=1:K
centroids(k, :) = mean(X(idx==k, :));
end

end
```

5. Реализуйте алгоритм K-средних.

```
function [centroids, idx] = runkMeans(X, initial_centroids, ...
max_iters, plot_progress)
if ~exist('plot_progress', 'var') || isempty(plot_progress)
plot_progress = false;
end

if plot_progress
figure;
hold on;
end

[m n] = size(X);
K = size(initial_centroids, 1);
centroids = initial_centroids;
previous_centroids = centroids;
idx = zeros(m, 1);

for i=1:max_iters
fprintf('K-Means iteration %d/%d...\n', i, max_iters);
if exist('OCTAVE_VERSION')
fflush(stdout);
end
idx = findClosestCentroids(X, centroids);
if plot_progress
plotProgresskMeans(X, centroids, previous_centroids, idx, K, i);
previous_centroids = centroids;
fprintf('Press enter to continue.\n');
pause;
end
centroids = computeCentroids(X, idx, K);
end

if plot_progress
hold off;
end

end
```

6. Постройте график, на котором данные разделены на K=3 кластеров (при помощи различных маркеров или цветов), а также траекторию движения центров кластеров в процессе работы алгоритма

```
load('ex7data2.mat');

K = 3;
initial_centroids = [3 3; 6 2; 8 5];

idx = findClosestCentroids(X, initial_centroids);
fprintf('Closest centroids for the first 3 examples: %d %d %d', idx(1:3));
```

```
Closest centroids for the first 3 examples: 1 3 2
```

```
centroids = computeCentroids(X, idx, K);
fprintf('Centroids computed after initial finding of closest centroids: \n %f %f \n %f
```

```
Centroids computed after initial finding of closest centroids:
 2.428301 3.157924
 5.813503 2.633656
 7.119387 3.616684
```

```
max_iters = 10;
initial_centroids = [3 3; 6 2; 8 5];

figure('visible','on'); hold on;
plotProgresskMeans(X, initial_centroids, initial_centroids, idx, K, 1);
xlabel('Press ENTER in command window to advance','FontWeight','bold','FontSize',14)
[~, ~] = runkMeans(X, initial_centroids, max_iters, true);
```

```
K-Means iteration 1/10...
Press enter to continue.
K-Means iteration 2/10...
Press enter to continue.
K-Means iteration 3/10...
Press enter to continue.
K-Means iteration 4/10...
Press enter to continue.
K-Means iteration 5/10...
Press enter to continue.
K-Means iteration 6/10...
Press enter to continue.
K-Means iteration 7/10...
Press enter to continue.
K-Means iteration 8/10...
Press enter to continue.
K-Means iteration 9/10...
Press enter to continue.
K-Means iteration 10/10...
Press enter to continue.
```

```
set(gcf,'visible','off'); hold off;
```

7. Загрузите данные **bird_small.mat** из файла.

8. С помощью алгоритма K-средних используйте 16 цветов для кодирования пикселей.

9. Насколько уменьшился размер изображения? Как это сказалось на качестве?

```
%  Load an image of a bird
A = double(imread('bird_small.png'));
A = A / 255; % Divide by 255 so that all values are in the range 0 - 1

% Size of the image
img_size = size(A);
X = reshape(A, img_size(1) * img_size(2), 3);
K = 16;
max_iters = 10;
initial_centroids = kMeansInitCentroids(X, K);
% Run K-Means
[centroids, ~] = runkMeans(X, initial_centroids, max_iters);
```

```
K-Means iteration 1/10...
K-Means iteration 2/10...
K-Means iteration 3/10...
K-Means iteration 4/10...
K-Means iteration 5/10...
```

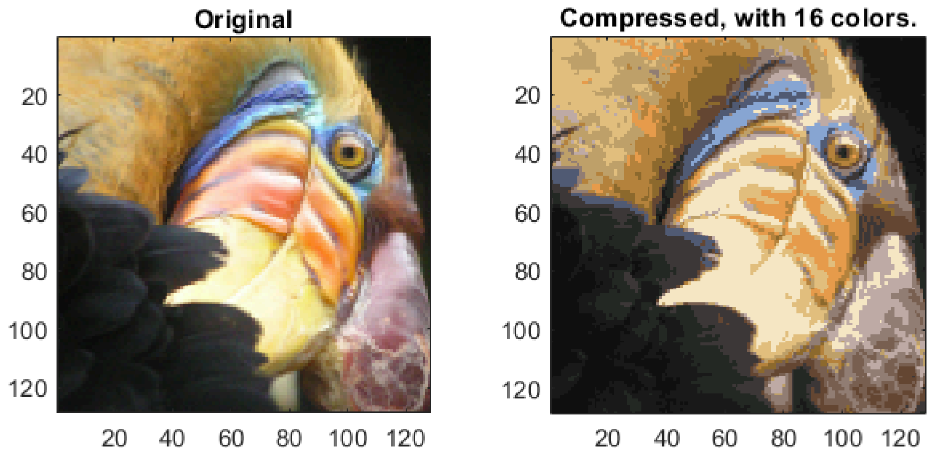
```
K-Means iteration 6/10...
K-Means iteration 7/10...
K-Means iteration 8/10...
K-Means iteration 9/10...
K-Means iteration 10/10...
```

```
idx = findClosestCentroids(X, centroids);
X_recovered = centroids(idx,:);

% Reshape the recovered image into proper dimensions
X_recovered = reshape(X_recovered, img_size(1), img_size(2), 3);

% Display the original image
figure;
subplot(1, 2, 1);
imagesc(A);
title('Original');
axis square

% Display compressed image side by side
subplot(1, 2, 2);
imagesc(X_recovered)
title(sprintf('Compressed, with %d colors.', K));
axis square
```



Original

Compressed, with 16 colors.