A comparison between the compression ratios the files as follows

| Random.png | checker |
|---|---|
| 11:1 | 427:1 |

## Step 1:  read and flatten the image

```
BufferedImage image = ImageIO.read(new File("D:\\T7\\New folder\\data compression\\proj\\Image
Files\\checker.png"));


    int width = image.getWidth();

    int height = image.getHeight();

    int[] pixels = new int[width * height];    int k =0;


    for (int y = 0; y < height; y++) {

      for (int x = 0; x < width; x++) {

        int pixel = image.getRGB(x, y) & 0xFF; // Extract the blue channel, which represents intensity in
grayscale

        pixels[k++] = pixel;

      }

    }
```

## Step 2:  compress the flattened pixels

### ➢ Create dictionary of all pixels values with unique

```
        HashMap<String, Integer> dictionary = new HashMap<>();

    for (int i = 0; i < 256; i++) {

      dictionary.put(String.valueOf((char)i), i);

    }
```

- **Iterate over each pixel and concatenate it with the next pixel**

```
for (int pixel : pixels)

{

  char next = (char) pixel;

  String currentPlusNext = current + next;
```

- **If the Concatenated pixel already exist I will output nothing then update the current to current + next**

```
if (dictionary.containsKey(currentPlusNext))

          current = currentPlusNext;
```

- **If the Concatenated pixel doesn’t exist I will get the unique key of the current pixel to be the new output then add it to the compressed array**

```
else {

      int output = dictionary.get(current);

       compressed.add( output );
```

- **Update the dictionary with the new concatenated pixel ’’currentPlusNext’’ then move the current to start with next**

```
dictionary.put(currentPlusNext, dictSize++);

 current = String.valueOf( next );
```

## Step 3: serialize the compressed array

## Step 4: decompress

- **Create dictionary of all pixels values with unique**

```
    HashMap<Integer, String> dictionary = new HashMap<>();
for (int i = 0; i < 256; i++) {
  dictionary.put(i, String.valueOf((char) i));    System.out.println( (char)i);
}
```

- **Iterate over each element in the compressed array**

```
  for (int code : compressed) {
   String entry;
```

- **If the value already exists in the dictionary I will retrieve its label ''entry''**

```
if (dictionary.containsKey(code))
        {
           entry = dictionary.get(code);
        }
```

- **If the value doesn't exist this mean that the value is unknown to me ''ab?'' so entry will be the current + the first letter of the current**

```
else if (!dictionary.containsKey(code))
        {
           entry = current + current.charAt(0);
        }
```

- **add entry to the decompressed list which will be the image pixel**

```
decompressed.append(entry);
```

- **update the dictionary with the (current and the first letter of the next)**

```
dictionary.put(dictSize++, (current + entry.charAt(0) ) );
```

- **move the current to start with next**

```
current = entry;
```