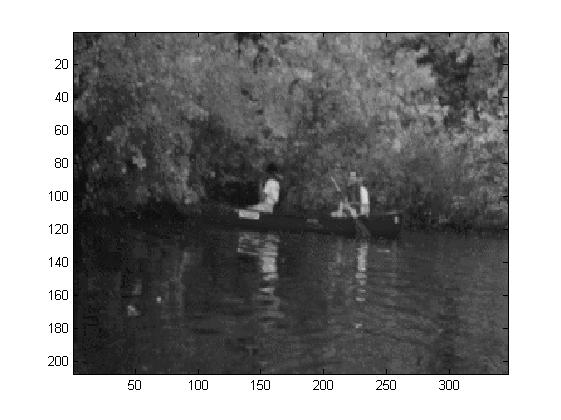
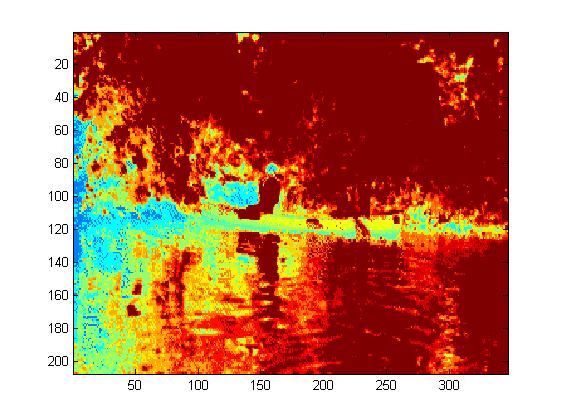
* ***Question 1:* What happens in the image apart from the color change?**
* ***Answer:*** When we load *canoe256* in a script and *Canoe* is store in image then for so much extra color the output image dose not come as a visible image. But, when we using *showgrey* function then we get a clear image figure and we can see from the figure image of a boat with two boatmen or fisherman in a canal.



Before use *showgrey* After use *showgrey*

**Figure: Question 1**

* ***Question 2:*** **Why does a pattern appear in the background of the telephone image? How many grey-levels are needed to get a (subjectively) acceptable result in this case?**
* ***Answer:*** By using *phonecalc256* from the image-m, we can can find out a telephone image figure in the pattern as an output. By using *showgrey (p,32)* function, we find a clear and clearly visible image.



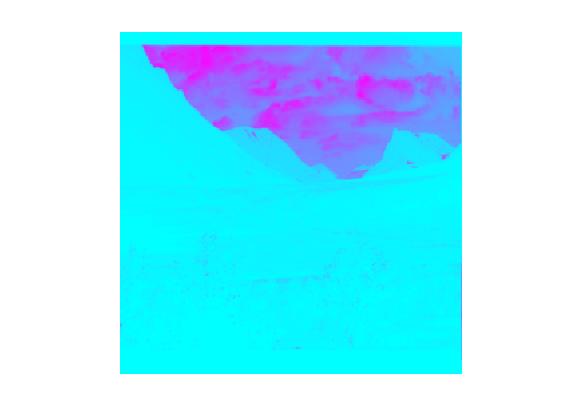
**Figure: Question 2**

* ***Question 3:*** **What does the image show? Why it is difficult to interpret the information in the original image?**
* ***Answer:*** From this figure we can find a view of night sky with a mountain area. We also find white points in the whole figure. For the white points, it is so much difficult to find information from the main figure.

******

**Figure: Question 3**

* ***Question 4:*** **Interpret the plots in terms of image content. Which colormap makes the visualization more clear and why?**
* ***Answer:*** We can see two plots from the figure. If we use *colormap(hot)* and *showgray* function we can find one sun shining area but, if we use *colormap(cool)* and use *showgray* function, we can find out a image of without sun and iceberg area. So, finally we must say that if we use *colormap(hot)*, then we find out a more visible plot or image files.

Using *colormap(hot)* Using *colormap(cool)*

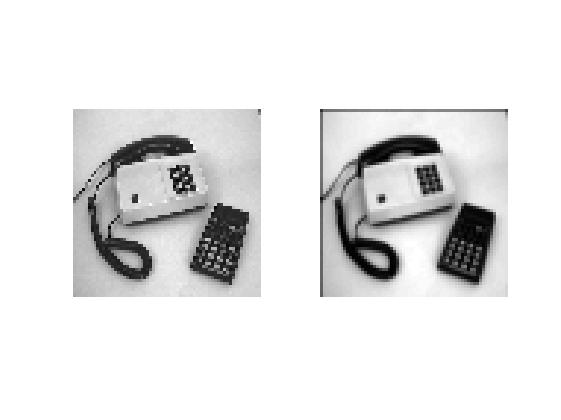
**Figure: Question 4**

* ***Question 5:*** **Repeatedly apply the subsampling operator to some of the images mentioned above. What are the results and your conclusion?**
* ***Answer:*** We use *rawsubsample* and *nallo256* applied for getting this output. We print the script by using subplot three times. We find out a clear and visible output from the fist subplot. If we use *rawsubsample* for second subplot, then the figure becomes fuzzy. For third times, it becomes more fuzzy.



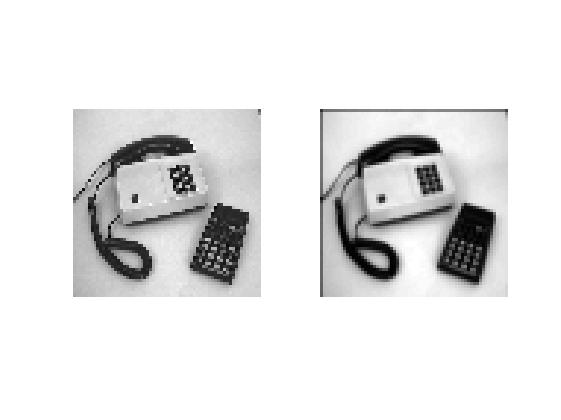
**Figure: Question 5**

* ***Question 6:*** **Describe in which ways the results are similar and different. Explain the reasons behind the differences.**
* ***Answer:*** By using *rawsubsample* and *binsubsample* with using *showgrey* function and sub plotting, we get the similar and different image or output files. We can find out similar image or output with using *phonecalc256* and *rawsubsample* but the output image becomes fuzzy. But, when we use *binsubsample*, as an output we also get a more fuzzy image but that image was different from first image and also get a bold or black border image.



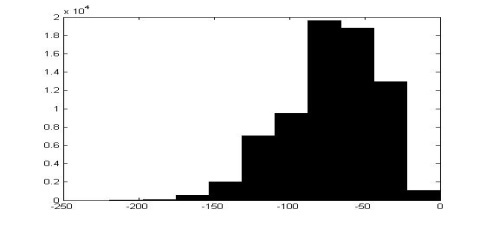
**Figure: Question 6**

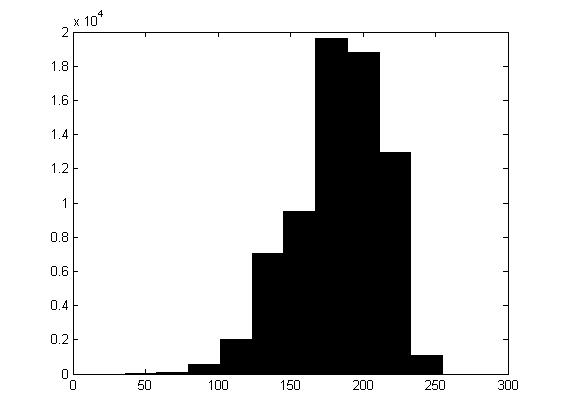
* ***Question 7:*** **What will the results be if you repeatedly apply these two types of operators to a textured image?**
* ***Answer:*** If we applied two operators at a same time then we get two image files as per output file. From first output file that image file was fuzzy. From second image file we saw that image was so much clear more than first image but in second image upper side and the left side has a black bold border.



**Figure: Question 7**

* ***Question 8:* Related to neg1 and neg2 – why are the histograms different but images similar? Related to *nallo* – explain how the above transformation functions affect the image histograms.**
* ***Answer:*** At firstwe applied *hist(neg1(:))* then we get a plot of graph where graph’s plot stay X-axis and Y-axis. This graph started from the right side of the frame corner and finished left side of the frame corner and we get a image of plot in the middle of the axis by using *hist(neg2(:))*. If we use *nallo256* and with using *showgrey* function we get a grey color image. If this output with connect with *hist(neg1(:))*, then we get a type of graph. If this graph connect with *hist(neg2(:))*, then we get a another type of graph.

**Before use *hist(neg1(:))*,*hist(neg2(:))* Before use *hist(neg2(:))***

**After use *hist(neg1(:))*, *hist(neg2(:))***

**Figure: Question 8**

* ***Question 9:* Why was value 1 added to the image before the look-up?**
* ***Answer:*** If we increase the value of Canoe and we increase the value-1, then get a clear image as an output. If someone wants to get more clear image then increasing the value of Canoe. But, in this process the image size was smaller before the image size.



**Figure: Question 9**

* ***Question 10:* What would be the best transformation function and why?**
* ***Answer:*** The best transformation function is colormap. Because of for this function the output comes with a sketch image and that image output was a little bit clear.

******

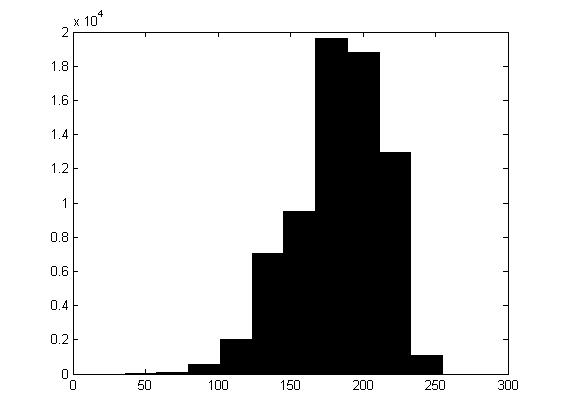
**Figure: Question 10**

* ***Question 11:*** **What are the effects of the operation? Explain in which cases it is a good technique. Why do you need the parameter a and how does it affect the result? What is a suitable value for a?**
* ***Answer:*** The effect of the operation is mainly sketch of the output image file. Besides we see a black dark shade in the lower part of the image file. For applied logarithmic function to *nallo256float* we mainly get this output. This is a good technique to get a sketch output file with a bold or dark line. I need declare the parameter because of affect the results. Suitable value of a is *nallo256.*



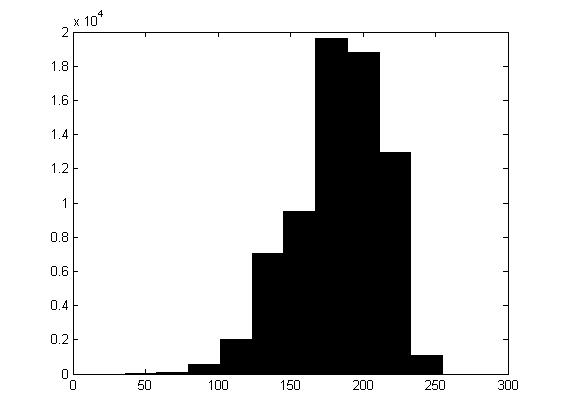
**Figure: Question 11**

* ***Question 12:*** **How do you expect the resulting image and histogram will look like? What did you actually get and why is it different? How does the histogram equalized image and its histogram depend on the number of accumulators?**
* ***Answer:*** I expect the resulting of image and histogram is as like a graph of some plotting area. I don’t get any specific image just get a graph. Histogram depends on the number because it is a graph image and graph image always depends on horizontal and vertical axis.



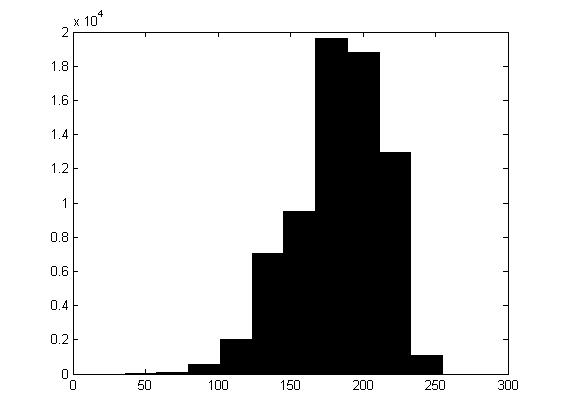
**Figure: Question 12**

* ***Question 13:* How are the results compared to the transformation functions used in previous sections? How will the results be for different kinds of images? What happens if you apply *histeq* to the similar image *nallo256what*?**
* ***Answer:*** If I will use different kind of images as an input then we get same output as like the Figure-13. If I apply *histeq* to the similar image *nallo256what*, then the output was same.



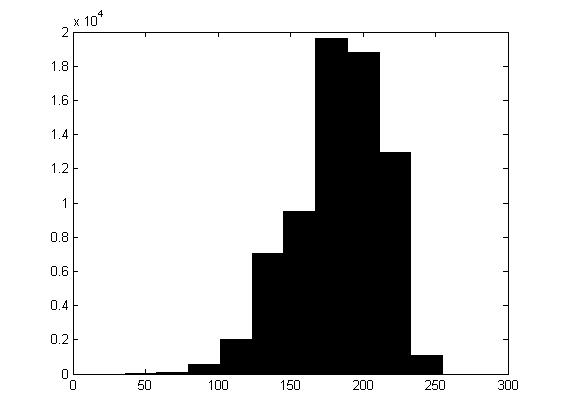
**Figure: Question 13**

* ***Question 14:*** **What happens if you histograms equalize the image *phonecalc256*? Think of what results to expect, before you perform the actual operation. What kinds of images are suitable for histogram equalization?**
* ***Answer:*** If I equalize the image *phonecalc256*, I will get the same graph of plotting. I expect as a result of some kind of plotting which is situated horizontal and vertical axis. Basically graph type images are suitable for histogram equalization.

******

**Figure: Question 14**

* ***Question 15:*** **Would it be possible to improve the results by exploiting the fact that *nallo256float* contains image data stored as floating-point values, unlike-*nallo256* that contains integers?**
* ***Answer:*** If I equalize the image *nallo256float*, I will get the same graph of plotting. I expect as a result of some kind of plotting which is situated horizontal and vertical axis. Basically graph type images are suitable for histogram equalization. Yes, it is possible to improve the results by exploiting the fact that *nallo256float* contains image data stored as floating point values, unlike-*nallo256* that contains integers.

******

**Figure: Question 15**

**........The End........**