

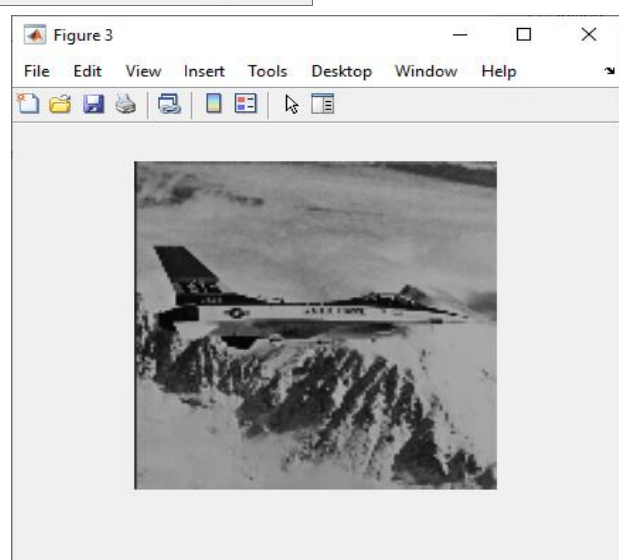
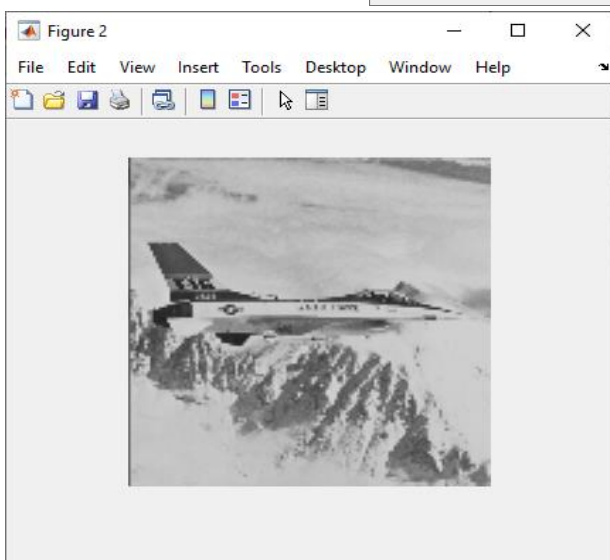
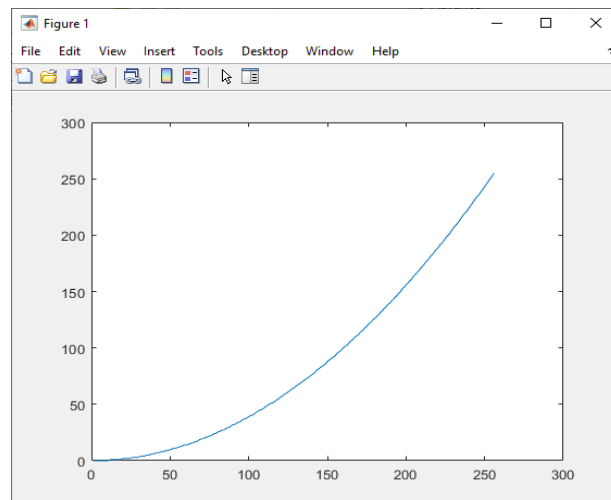
Digital image processing and vision systems – lab #2

Date performed: 16.03.2021	Group 2
Author name: Krzysztof Klimczyk	

1. Source codes and screenshots:

Task 2.2

```
1. load functionsLUT.mat;  
2. figure(1);  
3. plot(square);  
4.  
5. figure(2);  
6. jet = imread("jet.bmp");  
7. imshow(jet);  
8.  
9. figure(3);  
10. B = intlut(jet,square);  
11. imshow(B);
```



```

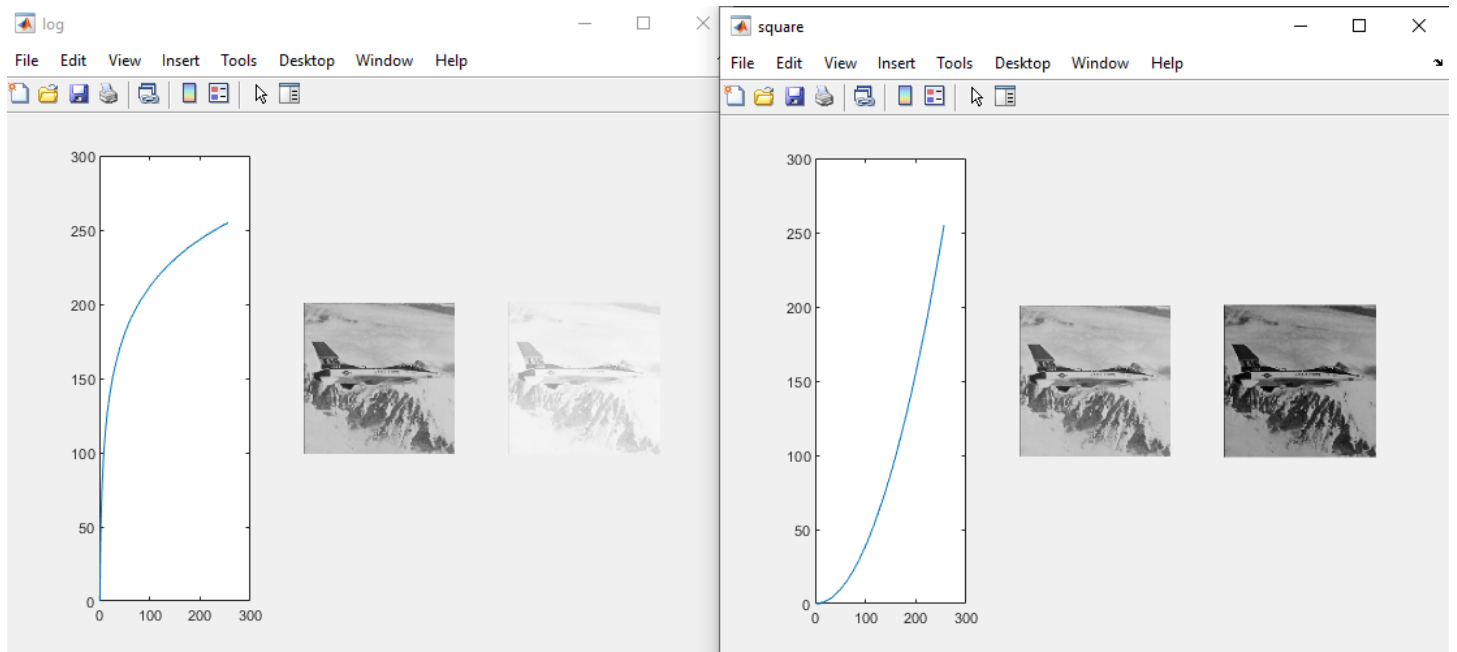
1. load functionsLUT.mat;
2. jet = imread("jet.bmp");
3.
4. figure('Name','log','NumberTitle','off');
5. LUT(jet,log);
6.
7. figure('Name','square','NumberTitle','off');
8. LUT(jet,square);
9.
10. figure('Name','invlog','NumberTitle','off');
11. LUT(jet,invlog);
12.
13. figure('Name','inverse','NumberTitle','off');
14. LUT(jet,inverse);
15.
16. figure('Name','squareroot','NumberTitle','off');
17. LUT(jet,squareroot);
18.
19. figure('Name','saw','NumberTitle','off');
20. LUT(jet,saw);
21.
22. figure('Name','exponential','NumberTitle','off');
23. LUT(jet,exponential);

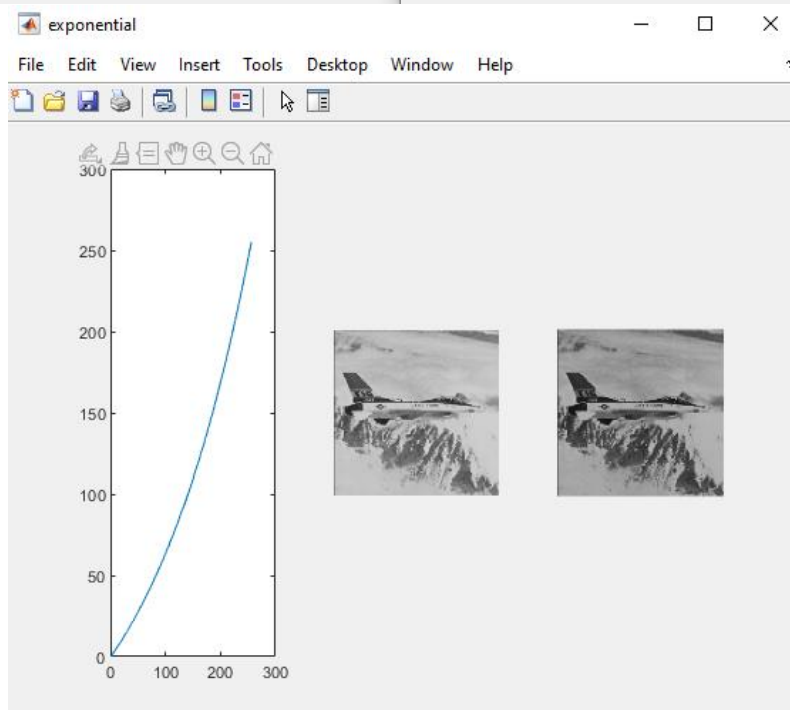
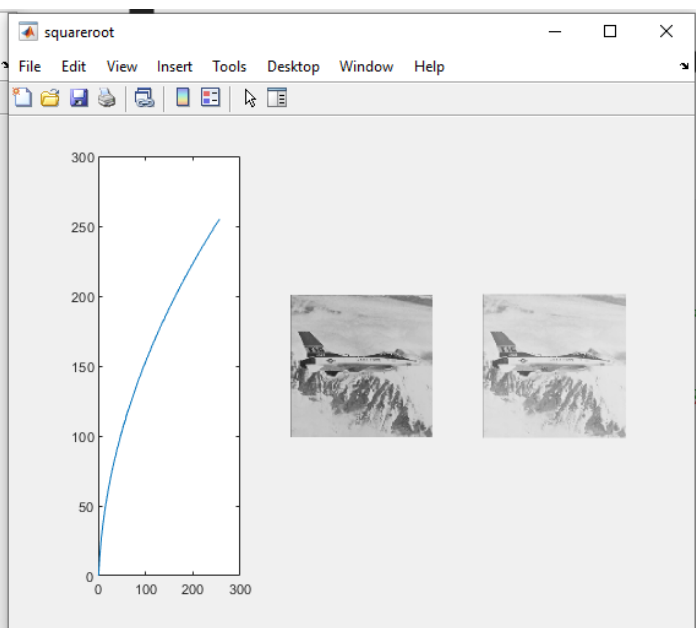
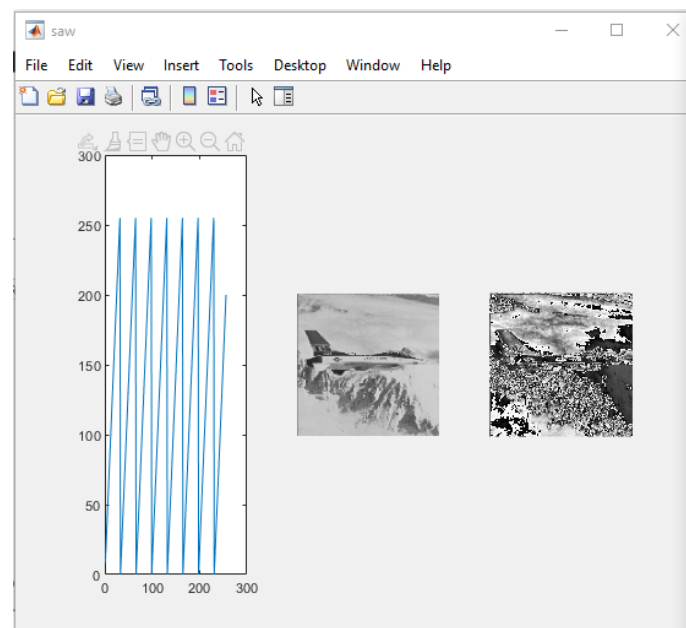
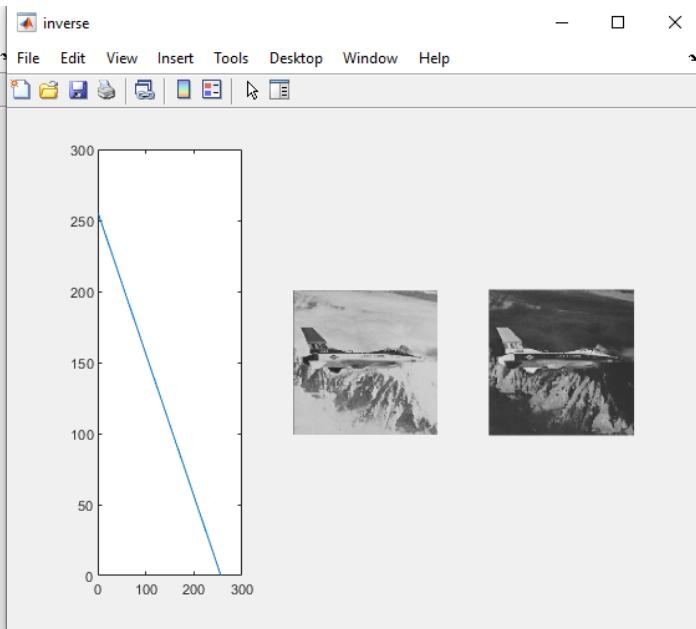
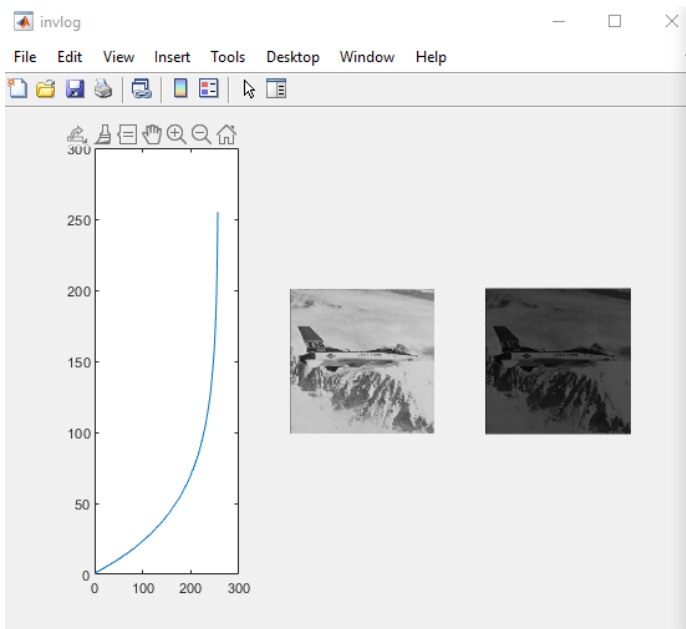
```

```

1. function LUT(image,reencoding)
2. A = intlut(image,reencoding);
3. subplot(1,3,1);
4. plot(reencoding);
5. subplot(1,3,2);
6. imshow(image);
7. subplot(1,3,3);
8. imshow(A);
9. end

```





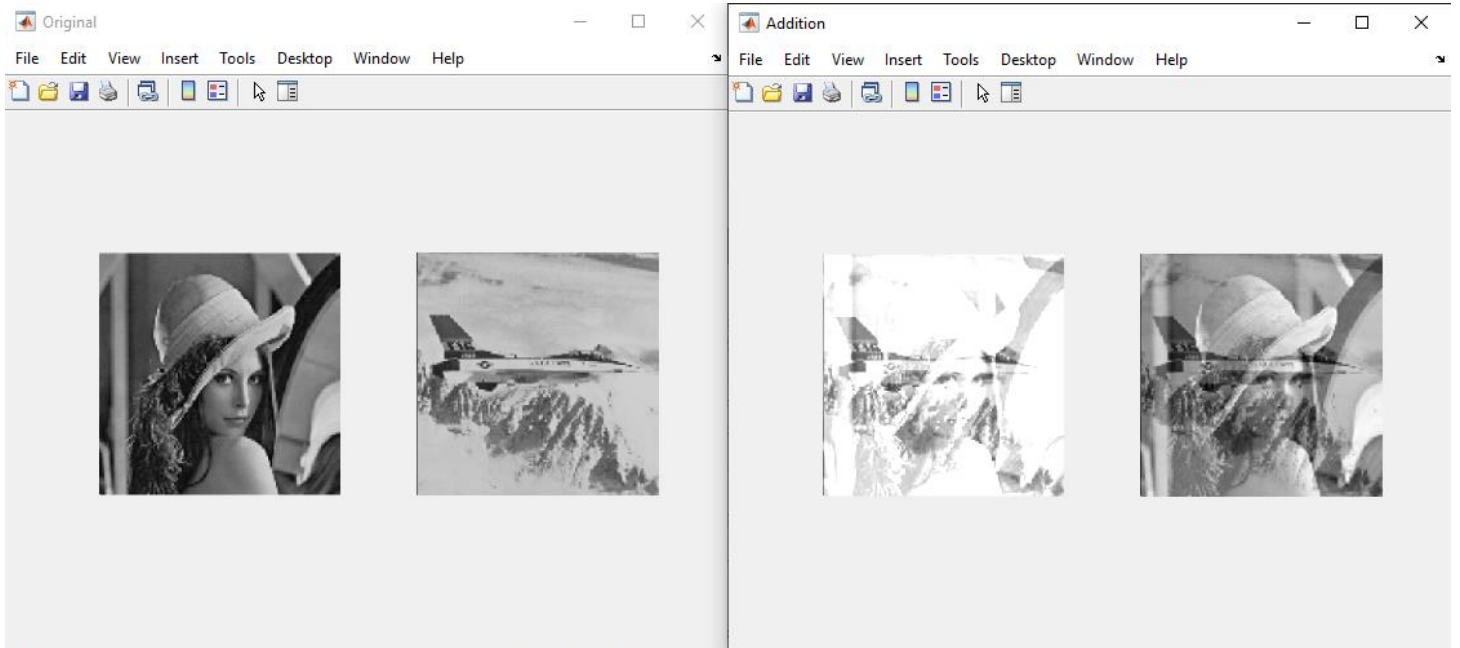
Task 2.3

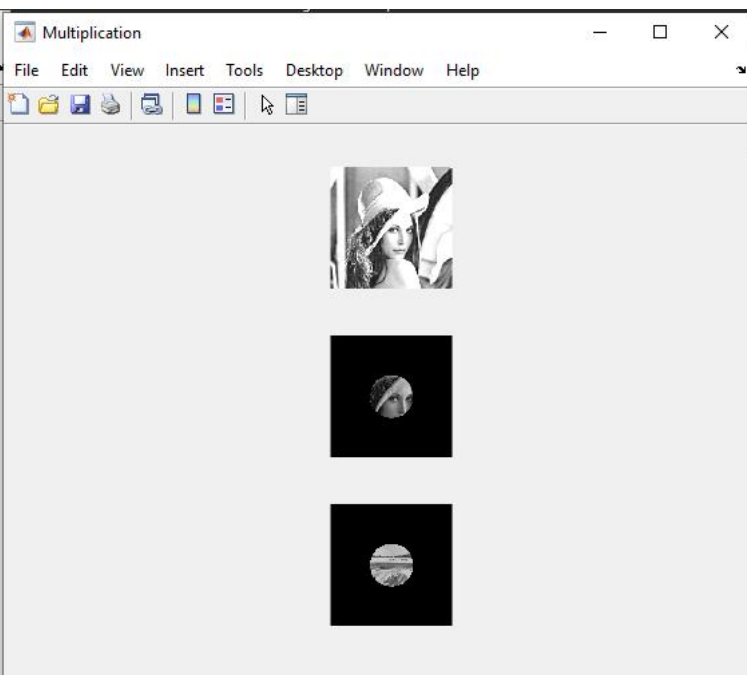
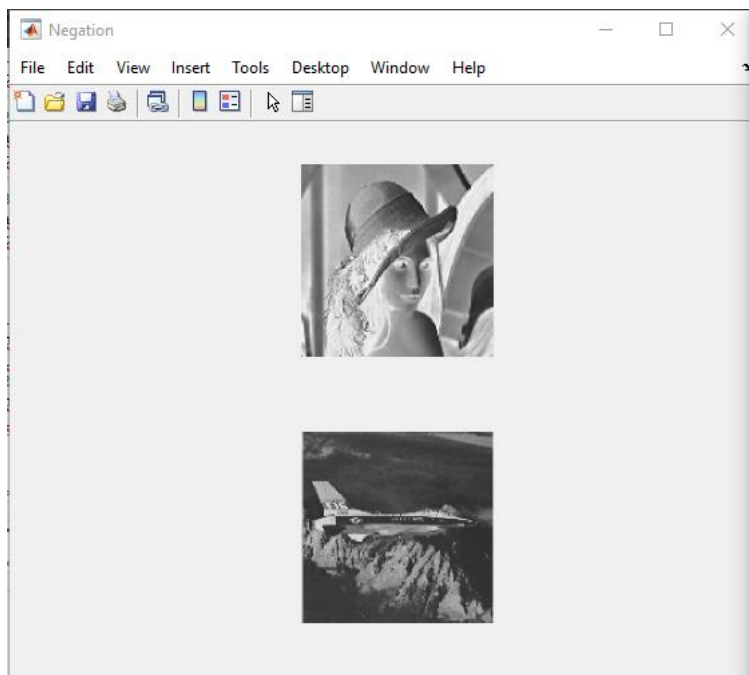
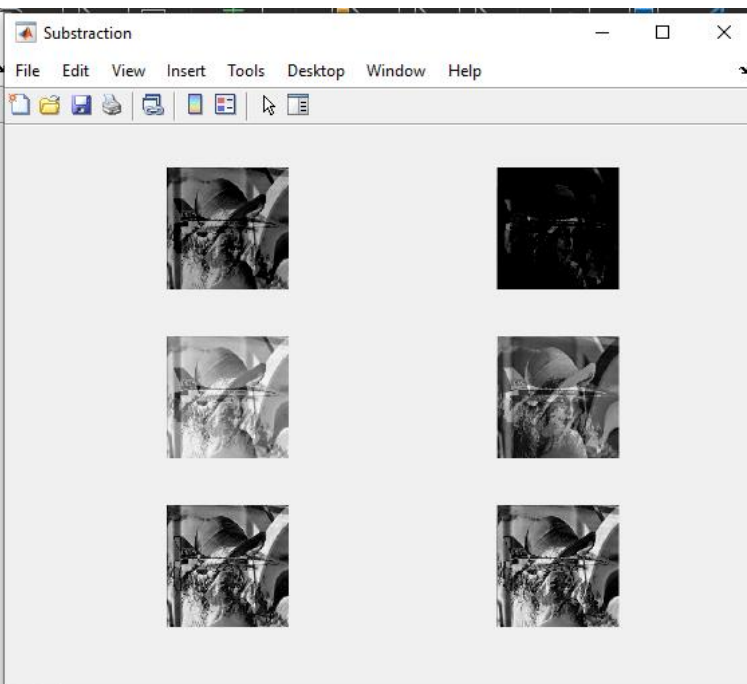
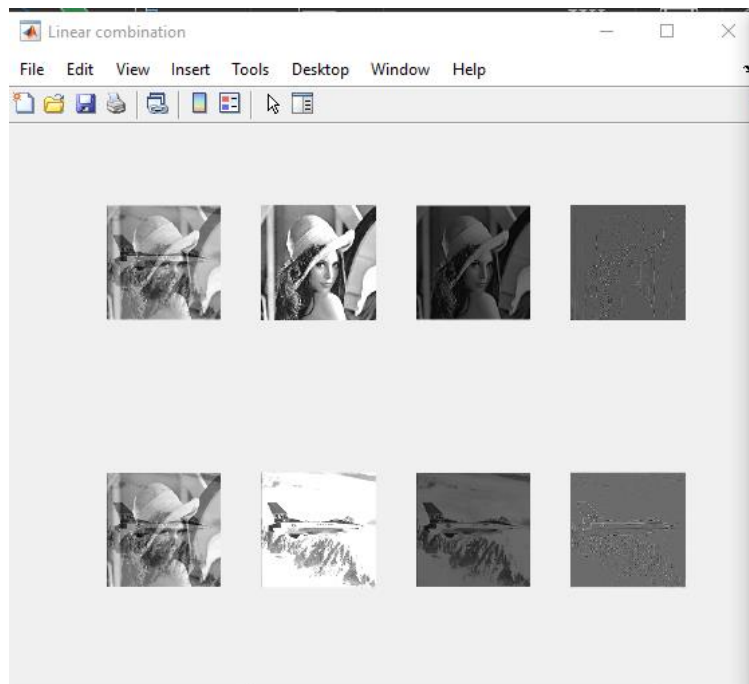
```
1. lena = imread("lena.bmp");
2. jet = imread("jet.bmp");
3. mask = imread("circle.bmp");
4.
5. figure('Name','Original','NumberTitle','off');
6. subplot(1,2,1);
7. imshow(lena);
8. subplot(1,2,2);
9. imshow(jet);
10.
11. figure('Name','Addition','NumberTitle','off');
12. subplot(1,2,1);
13. Add = imadd(lena,jet);
14. imshow(Add);
15. subplot(1,2,2);
16. Add = imadd(lena,jet,'uint16');
17. imshow(Add,[]);
18.
19. figure('Name','Linear combination','NumberTitle','off');
20. subplot(2,4,1);
21. Lin = imlincomb(1,lena,1,jet,'uint16');
22. imshow(Lin,[]);
23. subplot(2,4,2);
24. Lin = imlincomb(1.5,lena);
25. imshow(Lin);
26. subplot(2,4,3);
27. Lin = imlincomb(0.5,lena);
28. imshow(Lin);
29. subplot(2,4,4);
30. J = uint8(filter2(fspecial('gaussian'), lena));
31. Lin = imlincomb(1,lena,-1,J,64);
32. imshow(Lin,[]);
33.
34. subplot(2,4,5);
35. Lin = imlincomb(1,jet,1,lena,'uint16');
36. imshow(Lin,[]);
37. subplot(2,4,6);
38. Lin = imlincomb(1.5,jet);
39. imshow(Lin);
40. subplot(2,4,7);
41. Lin = imlincomb(0.5,jet);
42. imshow(Lin);
43. subplot(2,4,8);
44. J = uint8(filter2(fspecial('gaussian'), jet));
45. Lin = imlincomb(1,jet,-1,J,128);
46. imshow(Lin,[]);
47.
48. figure('Name','Subtraction','NumberTitle','off');
49. subplot(3,2,1);
50. Sub = imsubtract(jet,lena);
51. imshow(Sub);
52. subplot(3,2,2);
53. Sub = imsubtract(lena,jet);
54. imshow(Sub);
55.
56. lena16 = int16(lena);
57. jet16 = int16(jet);
58.
59. subplot(3,2,3);
```

```

60. Sub = imsubtract(jet16,lena16);
61. imshow(Sub,[]);
62. subplot(3,2,4);
63. Sub = imsubtract(lena16,jet16);
64. imshow(Sub,[]);
65.
66. subplot(3,2,5);
67. Sub = imabsdiff(jet,lena);
68. imshow(Sub,[]);
69. subplot(3,2,6);
70. Sub = imabsdiff(lena,jet);
71. imshow(Sub,[]);
72.
73. figure('Name','Multiplication','NumberTitle','off');
74. subplot(3,1,1);
75. Mult = immultiply(lena,2);
76. imshow(Mult);
77. subplot(3,1,2);
78. mask = logical(mask);
79. Mult = immultiply(lena,mask);
80. imshow(Mult);
81. subplot(3,1,3);
82. mask = logical(mask);
83. Mult = immultiply(jet,mask);
84. imshow(Mult);
85.
86. figure('Name','Negation','NumberTitle','off');
87. subplot(2,1,1);
88. Negation = imcomplement(lena);
89. imshow(Negation);
90. subplot(2,1,2);
91. Negation = imcomplement(jet);
92. imshow(Negation);

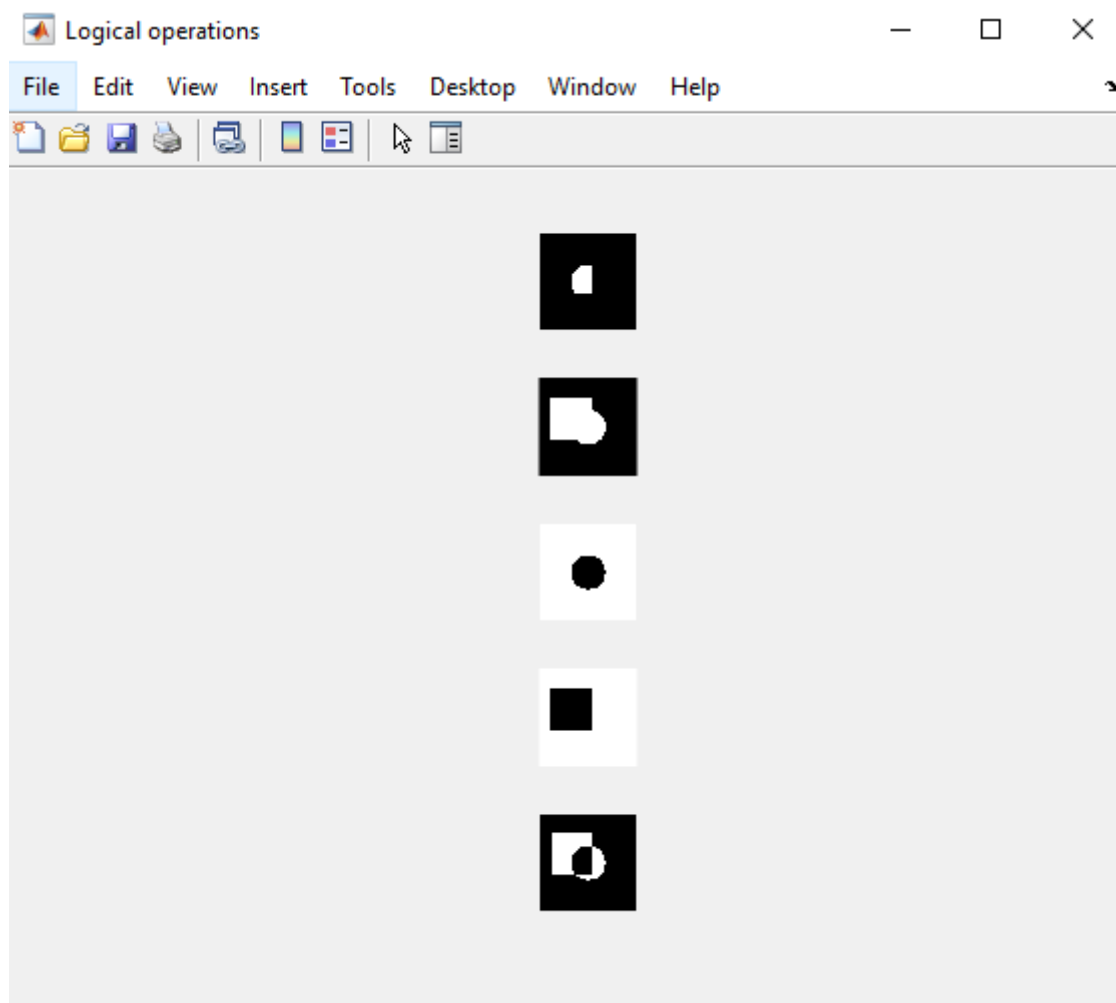
```





Task 2.4

```
1. circle = imread("circle.bmp");
2. square = imread("square.bmp");
3. circle = logical(circle);
4. square = logical(square);
5.
6. figure('Name','Logical operations','NumberTitle','off');
7. subplot(5,1,1);
8. output = and(circle,square);
9. imshow(output);
10. subplot(5,1,2);
11. output = or(circle,square);
12. imshow(output);
13. subplot(5,1,3);
14. output = not(circle);
15. imshow(output);
16. subplot(5,1,4);
17. output = not(square);
18. imshow(output);
19. subplot(5,1,5);
20. output = xor(circle,square);
21. imshow(output);
```



2. Conclusions:

The result of summation is not satisfactory because the image is much brighter than the originals. Values of appropriate pixels are added but there are not rescaled to the proper range. If uint16 format will be used the result have similar brightness to the originals images and it avoids truncating output.

Subtraction is also exposed to exceeds the range and there is a difference between output images depending on arguments order in this function. The int16 data type and rescaling are correcting brightness so the result image is much easier to distinguish between two inputs. Using a function that subtracts absolute values produces the same results, regardless of the order of the arguments.

The boolean() function is not available in base Matlab so I used logical() method instead and it gives the same result.