|  |  |  |  |
| --- | --- | --- | --- |
| http://vtp.ifsp.edu.br/wp-content/themes/ifsp_votu/images/logo_ifsp.png | | ANÁLISE E DESENVOLVIMENTO DE SISTEMAS 6º PERÍODO | |
| Aluno:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | |
| **Disciplina:** TÓPICOS EM TECNOLOGIA DA INFORMAÇÃO | | | 2º SEMESTRE DE 2021 |
| **Professora:** IVAN OLIVEIRA LOPES | | | Votuporanga, 10 de DEZEMBRO de 2021 |
| **Valor:** 3.0 pontos | Nota:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Assinatura Professor:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | | | |
| **Observações:**   1. A prova é INDIVIDUAL; 2. Leia atentamente as questões. A interpretação destas FAZ PARTE da prova;   **Boa Prova!** | | | |
| u | | | |
| 1. Fazer um algoritmo para transformar uma imagem em níveis de cinza em uma imagem apenas 4 níveis de cinza;   pkg load image;  clc;  img = imread('marcada.bmp');  img4 = img/64;  img4 = uint8(img4);  img4 = img4\*64;  imshow(img4);   1. Fazer um algoritmo para que uma imagem com 256 níveis de cinza fique mais clara ou mais escura, de acordo com os dados abaixo:   Quando o pixel tiver valor maior que 220, este deverá ficar 30 tons de cinza mais claro;  Quando o pixel tiver valor menor que 120, este deverá ficar 30 tons de cinza mais escuro;  pkg load image;  clc;  img = imread('marcada.bmp');  imgGrey = img/255;  imgGrey = uint8(imgGrey);  imgGrey = imgGrey\*255;  [X,Y] = size(img);  for i = 1:X  for j = 1:Y  if imgGrey(i,j) > 220  imgGrey(i,j) - 30;  elseif imgGrey(i,j) < 120  imgGrey(i,j) + 30;  endif  endfor  end  figure(1),imshow(img);  figure(2),imshow(imgGrey);   1. Fazer um algoritmo para aplicar um filtro passa-baixa (suavização) em uma imagem, usando a seguinte máscara:   pkg load image;  clc;  img = imread('marcada.bmp')  pb\_filter = [1 2 1;2 0 2;1 2 1]  pb\_filter = pb\_filter/12;  imgSuavizada = conv2(img, pb\_filter);  imgSuavizada = uint8(imgSuavizada);  figure(1),subplot(1,2,1),imshow(img);  figure(1),subplot(1,2,2),imshow(imgSuavizada);   1. Na pasta da prova (exemplo: provaIvan, provaPedro, ProvaJamily,....) existem três imagens chamadas imagem A, imagem B e imagem C. Faça um algoritmo para somar as três imagens (gerando uma imagem nova) para que ela fique com a seguinte visualização:     pkg load image;  clc;  imgA = imread('ImagemA.bmp');  imgB = imread('ImagemB.bmp');  imgC = imread('ImagemC.bmp');  [Bx,By] = size(imgB);  imgResultado = imgA;  for i = 1:Bx  for j = 1:By  imgResultado(i,end-By+j)=imgB(i,j);  endfor  end  [Cx,Cy] = size(imgC);  for i = 1:Cx  for j = 1:Cy  imgResultado(end-Cx+i,end-Cy+j)=imgC(i,j);  endfor  end  figure(1),subplot(1,4,1),imshow(imgA),title('Imagem A');  figure(1),subplot(1,4,2),imshow(imgB),title('Imagem B');  figure(1),subplot(1,4,3),imshow(imgC),title('Imagem C');  figure(1),subplot(1,4,4),imshow(imgResultado),title('Imagem Resultante');   1. Na pasta da prova (exemplo: provaIvan, provaPedro, ProvaJamily,....) existe uma imagem chamada marcada.bmp, esta imagem possui uma mensagem escondida. Tal mensagem foi inserida usando a função: c(i,j)=bitset(a(i,j),1,b(i,j)). Extraia a mensagem escondida e cole a imagem resultante aqui: | | | |
| QQQQ | | | |
|  | | | |

pkg load image;

clc;

img = imread('marcada.bmp');

[M,N] = size(img);

imgResultado = zeros(M,N);

for i = 1:M

for j = 1:N

imgResultado(i,j) = bitget(img(i,j),1);

endfor

end

figure(1),imshow(imgResultado);