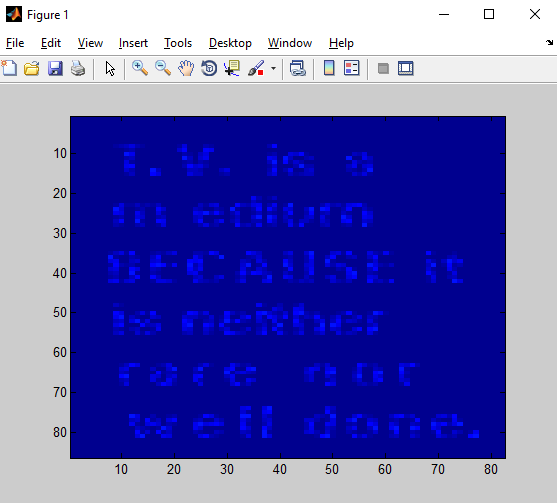
Os exercícios “A” e “B” foram realizados com o MatLab, os exercícios 1,2 e 3 foram realizados com o Octave

EXERCISE “A”

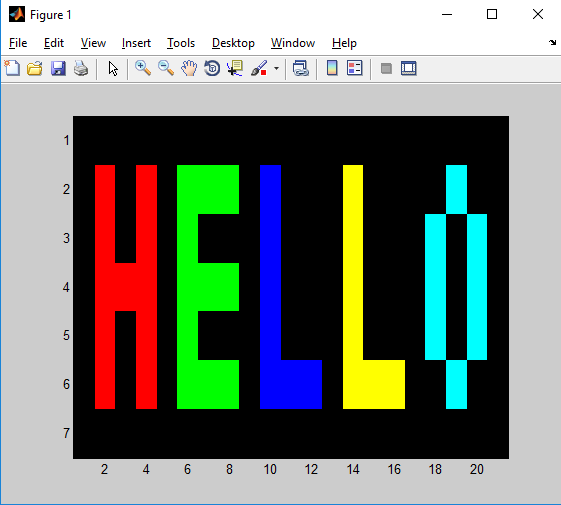
|  |
| --- |
| %Load image blackrectangle.bmp  blackrectangle=imread('blackrectangle.bmp');  %display blackrectangle  image(blackrectangle);  %1 = Red  %2 = Blue  %3 = Blue  blackrectangleRed=blackrectangle([558:643],[622:703],1);  blackrectangleGreen=blackrectangle([558:643],[622:703],2);  blackrectangleBlue=blackrectangle([558:643],[622:703],3);  image(blackrectangleRed)  image(blackrectangleGreen)  image(blackrectangleBlue)  %blackrectangleBlue is the image  %convert to double  text=double(blackrectangleBlue);  %vector A rows 13:80 column 24  vectorA=text(13:80,24);  %vector B rows 13:80 column 45  vectorB=text(13:80,45);  %vector C rows 13:80 column 46  vectorC=text(13:80,46);  %Exponentiate vectorA to vectorB  vectorexp=vectorA.^vectorB;  %Multply  vectorMult=vectorexp.\*vectorB;  %Add vector C in ans  vectorSum=vectorMult+vectorC;  %Add 32 in ans  vector32=vectorSum+32;  %Char A  char(vector32)  %Char B = Transpose char A  char(vector32') |



|  |
| --- |
| ans =  B  u  t                  y  o  u  r                h  o  m  e  w  o  r  k                i  s                          w  e  l  l                    d  o  n  e  ans =  But your homework is well done  >> |

EXERCISE “B”

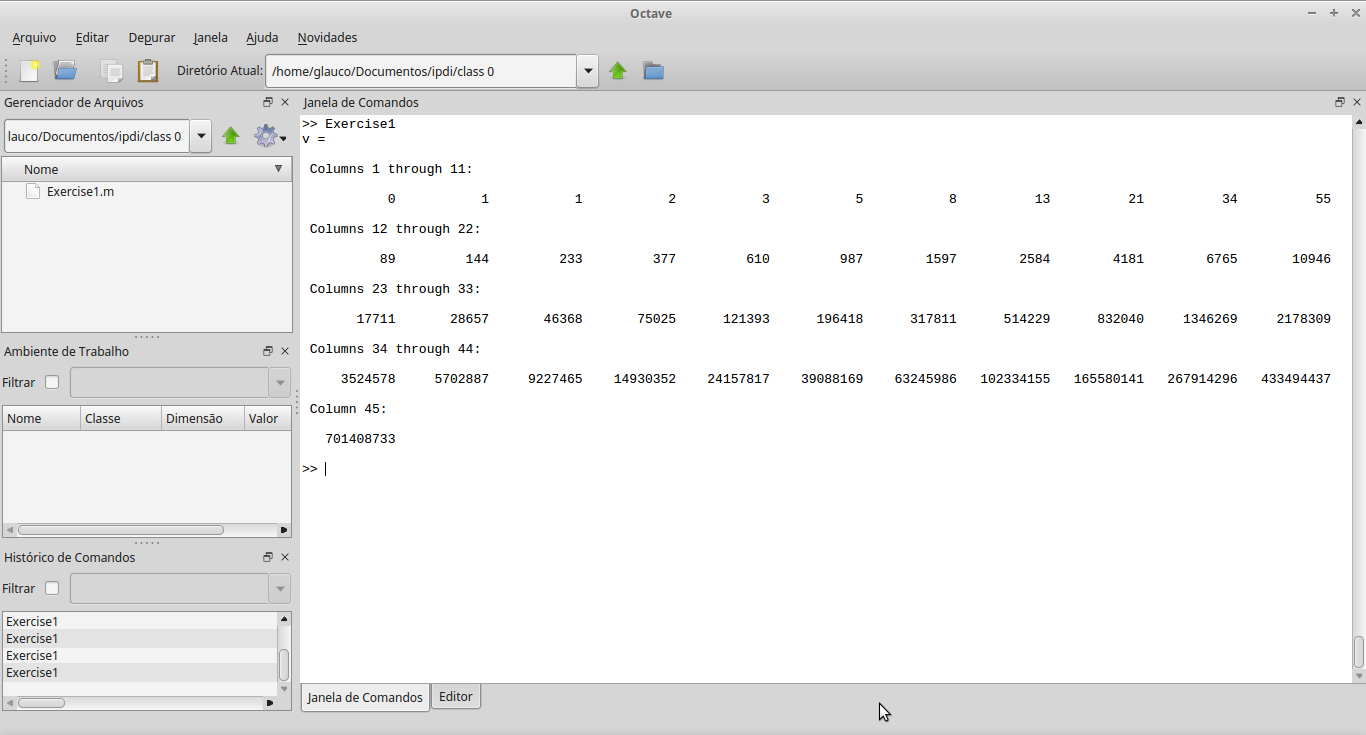
|  |
| --- |
| %Red Layer  r=rand(7,21)\*0;  %Green Layer  g =r;  %Blue Layer  b =r;    %Letter H  r(2:6,2) = [255,255,255,255,255];  r(2:6,4) = [255,255,255,255,255];  r(4,3) = 255;  %Letter E  g(2,6:8)=[255,255,255];  g(4,6:8)=[255,255,255];  g(6,6:8)=[255,255,255];  g(3,6)=255;  g(5,6)=255;  %Letter LB  b(2:6,10)=[255,255,255,255,255];  b(6,11:12)=[255,255];  %Letter LY  r(2:6,14)=[255,255,255,255,255];  r(6,15:16)=[255,255];  g(2:6,14)=[255,255,255,255,255];  g(6,15:16)=[255,255];  %Letter O  g(3:5,18)=[255,255,255];  g(3:5,20)=[255,255,255];  g(2,19)=255;  g(6,19)=255;  b(3:5,18)=[255,255,255];  b(3:5,20)=[255,255,255];  b(2,19)=255;  b(6,19)=255;    %Setting Matrix red values  hello(1:7,1:21,1)=r;  %Setting Matrix green values  hello(1:7,1:21,2)=g;  %Setting Matrix blue values  hello(1:7,1:21,3)=b;    %Showing Matrix  hello    %Turning matrix as HELLO  hello = uint8(hello)    %Showing Image ("HELLO")  image(hello) |



|  |
| --- |
| hello(:,:,1) =  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 255 0 255 0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0  0 255 0 255 0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0  0 255 255 255 0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0  0 255 0 255 0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0  0 255 0 255 0 0 0 0 0 0 0 0 0 255 255 255 0 0 0 0 0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  hello(:,:,2) =  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 0 0 0 0 255 255 255 0 0 0 0 0 255 0 0 0 0 255 0 0  0 0 0 0 0 255 0 0 0 0 0 0 0 255 0 0 0 255 0 255 0  0 0 0 0 0 255 255 255 0 0 0 0 0 255 0 0 0 255 0 255 0  0 0 0 0 0 255 0 0 0 0 0 0 0 255 0 0 0 255 0 255 0  0 0 0 0 0 255 255 255 0 0 0 0 0 255 255 255 0 0 255 0 0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  hello(:,:,3) =  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0 0 255 0 0  0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0 255 0 255 0  0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0 255 0 255 0  0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0 255 0 255 0  0 0 0 0 0 0 0 0 0 255 255 255 0 0 0 0 0 0 255 0 0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  hello(:,:,1) =  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 255 0 255 0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0  0 255 0 255 0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0  0 255 255 255 0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0  0 255 0 255 0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0  0 255 0 255 0 0 0 0 0 0 0 0 0 255 255 255 0 0 0 0 0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  hello(:,:,2) =  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 0 0 0 0 255 255 255 0 0 0 0 0 255 0 0 0 0 255 0 0  0 0 0 0 0 255 0 0 0 0 0 0 0 255 0 0 0 255 0 255 0  0 0 0 0 0 255 255 255 0 0 0 0 0 255 0 0 0 255 0 255 0  0 0 0 0 0 255 0 0 0 0 0 0 0 255 0 0 0 255 0 255 0  0 0 0 0 0 255 255 255 0 0 0 0 0 255 255 255 0 0 255 0 0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  hello(:,:,3) =  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0 0 255 0 0  0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0 255 0 255 0  0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0 255 0 255 0  0 0 0 0 0 0 0 0 0 255 0 0 0 0 0 0 0 255 0 255 0  0 0 0 0 0 0 0 0 0 255 255 255 0 0 0 0 0 0 255 0 0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |

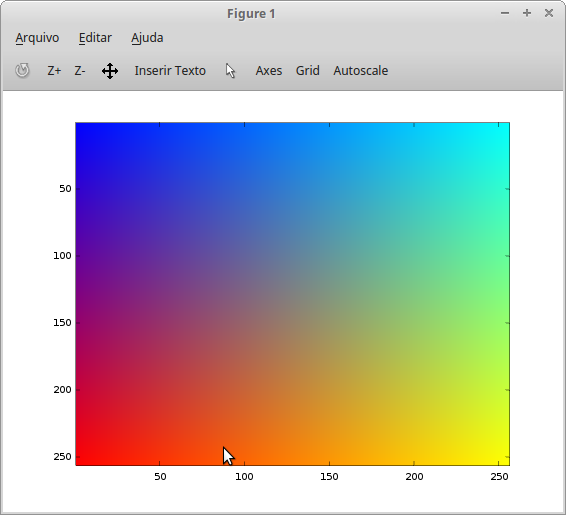
Exercise 1 - Single Loop: Fibonacci Numbers

|  |
| --- |
| function Exercise1()  v = rand(1,45)\*0;  v(1,2) = [1];  i=3;  while(i<=45)  v(1,i) = v(1,i-2) + v(1,i-1);  i=i+1;  end  v |



Exercise 2: Nested Loops: 2D gradient

|  |
| --- |
| function Exercise2()  %matriz gradiente vazia  %grad=rand(256,256)\*0;  grad=[]; %Slide 84  %Enunciado  % AZUL --- CYAN  % MAGENTA --- ????  % RED --- YELLOW  % 0~255 intensidade  % 1~256 dimensao da matriz  for i=1:256 %Slide 88  for j=1:256 %Slide 88  grad(i,j,1)=i; %vermelho aparece primeiro nas linhas depois nas colunas  grad(i,j,2)=j; %verde aparece primeiro nas colunas depois linhas  grad(i,j,3)=255 - (i-1); %Slide 87 decrementando o valor do azul, logo começa com o azul  end  end  image(uint8(grad));  end |



Exercise 3 - Nested Loops: Snake (not a real one)

NÃO USANDO IF-ELSE

|  |
| --- |
| function Exercise3()  %Criando a Matriz 8x8  n = rand(8,8)\*0;  %Valor escalar  %Cresce da esquerda para a direita  s = 0;  %Línhas impares  for i=1:i+2:8  for j=1:8  n(i,j) = s;  s = s + 1;  end  s = s + 8;  end  %pre visualizaçao  n  %Agora cresce da direita para a esquerda  s = 8;  for i=2:i+2:8  for j=8:-1:1  n(i,j) = s;  s = s + 1;  end  s = s + 8;  end  n  %mostrando a imagem  %image(uint8(n));  %usando a funçao dada  image(n), colormap(jet(64));  end |

USANDO IF-ELSE

|  |
| --- |
| function Exercise3()  %no exercicio colocou image(n) logo chamei a matriz de n  n = [ ];  s = 0;  %ideia eh criar linha com 8 valores  %trabalhar de forma mutua pra evitar trabalho  %linha impar e linha par  %na internet todo mundo usa mod ou rem pra verificar se eh impar ou par  for linha=1:8  if (rem(linha,2) == 1) %como x e y s~ao positivos posso usar rem no lugar de mod, mas o melhor eh usar mod  %impar  %se for impar eu incremento valor da esquerda pra direita (numero min=1 : numero max=8)  coluna = 1:8;  else  %par  %se for par eu incremento os valores da direita pra esquerda (numero maximo=8: -1 (para decrementar) : numero limite=1)  coluna = 8:-1:1;  %usa-se x:y:z quando se quer que o x=posição inicial; y o numero a somar a posição x; z = posição final  end  for j=coluna  n(linha,j) = s;  %como a matriz sempre incrementa de 1 em 1, logo, s++  s = s + 1;  end  end  n  image(n), colormap(jet(64)); %64=linha\*coluna=8\*8  end |

