


# Lista\_Varias\_questoes\_provas\_2025.pdf

## Semana 26 - 06/08/25

 Lista\_Varias\_questoes\_provas\_2025.pdf

# LEIA ↓

detalhe: **não** adicione **clear all** no início dos códigos, isso já é feito no menu. se precisar de limpar uma variável em específico, faça redeclarando-a ou limpando só ela.

ex.: preciso limpar o vetor/matriz X → **X=[];** ou **clear X;**

caso PRECISE de usar clear no código, coloque no final do código: “opcao = 1;”  
isso é pra o menu não quebrar, ele tá sempre usando a variável “opcao”

### menu.m

```
% Menu principal - Lista_Varias_questoes_provas_2025.pdf
```

```
% Semana 06/08
```

```
% Lucas Vasconcelos Resende - 205B
```

```
opcao = 1;
while (opcao != 0)
    clear all;
    opcao = input('digite o exercício ou 0 para sair: ');
    clc;
    close all
    switch(opcao)
        case 0
            printf('Saindo...\n');
            return
        case 1
            q1
        case 2
            q2
        case 3
            q3
        case 4
            q4
        case 5
            q5
        case 6
            q6
        case 7
```

q7  
case 8  
q8  
case 9  
q9  
case 10  
q10  
case 11  
q11  
case 12  
q12  
case 13  
q13  
case 14  
q14  
case 15  
q15  
case 16  
q16  
case 17  
q17  
case 18  
q18  
case 19  
q19  
case 20  
q20  
case 21  
q21  
case 22  
q22  
case 23  
q23  
case 24  
q24  
case 25  
q25  
case 26  
q26  
case 27  
q27  
case 28  
q28  
case 29  
q29  
case 30  
q30  
case 31

```
q31
case 32
q32
case 33
q33
case 34
q34
case 35
opt35 = input('Digite qual q35 deseja ver: ');
if opt35 == 1
    q35
elseif opt35 == 2
    q35pt2
endif
case 36
q36
case 37
q37
case 38
q38
case 39
q39
case 40
q40
case 41
q41
case 42
q42
case 43
q43
case 44
q44
case 45
q45
case 46
q46
case 47
q47
case 48
q48
case 49
q49
case 50
q50
case 51
q51
case 52
q52
```

```

case 53
    q53
case 54
    q54
case 55
    q55
case 56
    q56
case 57
    q57
case 58
    q58
case 59
    q59
case 60
    q60
otherwise
    printf("Digite uma opção válida!\n");
endswitch
endwhile

```

## q1.m

%lista varias questoes provas 2025 mateus leon 205

```
M = [0 1 0 0 1 0; ...
```

```
1 1 1 0 1 1; ...
```

```
0 0 0 1 0 1; ...
```

```
1 1 0 0 0 0; ...
```

```
0 0 0 1 1 1;]
```

```
onescount = 0;
```

```
zeroscount = 0;
```

```
for i = 1:size(M)(1)
```

```
    for k = 1:size(M)(2)
```

```
        if M(i, k) == 0
```

```
            zeroscount++;
```

```
        elseif M(i, k) == 1
```

```
            onescount++;
```

```
        endif
```

```
    endfor
```

```
endfor
```

```
printf("Número de zeros na matriz M: %d\n", zeroscount);
```

```
printf("Número de uns na matriz M: %d\n", onescount);
```

%gerando um vetor e matriz aleatorios

```
n = -1;
```

```
v = [];
```

```

vm = [];

while n <= 0
    n = input('Digite um tamanho para o vetor, maior que 0: ');
endwhile
v = randi(n, 1, n);

for i = 1:n
    vm(i, :) = v;
endfor

printf("matriz gerada:\n");
disp(vm);

```

## q2.m

```

% Questao 2 - Lista_Varias_questoes_provas_2025.pdf
% Semana 06/08
% Lucas Vasconcelos Resende - 205B

S = 0;
i = 1;
d = input('Digite d (1 se quiser que o somatório só funcione com impares, ou 2 se quiser que o somatório só funcione com pares): ');
a = input('Digite o valor de a: ');
b = input('Digite o valor de b: ');
figure(1, 'Name', 'Questao 2')
hold on
grid on
for k = a:b
    x(i) = k;
    if mod(i,2)==d
        S = S+i;
    endif
    y(i) = S;
    i=i+1;
endfor
printf("S = %d\n",S);
plot(x, y)

```

## q3.m

```

%1)
#y=3
#x=2
#y=-2x+3

```

```

xgraf= -6:6;
ygraf= -6:6;
y1 = ones(13); y1.*3;
x2 = ones(13); x2.*2;
y3 = -2.*xgraf+3;

figure
hold on
plot(xgraf, y1, 'r-');
plot(x2, ygraf, 'b-');
plot(xgraf, y3, 'g-');
title('Funções no gráfico')
grid on
axis([min(xgraf) max(xgraf) min(ygraf) max(ygraf)])
%2)
x = -10:10;
printf('Digite os parâmetros de duas retas:\n');
for i =1:2
    printf("Reta %d:\n", i);
    r(i, 1) = input('Digite A (inclinação): ');
    r(i, 2) = input('Digite B (constante): ');
endfor
y1 = r(1,1)*x + r(1,2);
y2 = r(2,1)*x + r(2,2);

figure
hold on
plot(x, y1, 'b');
plot(x, y2, 'r');
grid on
title('Gráfico das duas retas');

if r(1,1) == r(2,1) %retas paralelas
    if r(1,2) == r(2,2) %retas coincidentes
        printf("\nAs retas são coincidentes (infinitos pontos de interseção)\n");
    else
        printf("\nAs retas são paralelas e não se interceptam\n");
    endif
else %nem paralelas nem coincidentes
    %interseção:
    xi = (r(2,2) - r(1,2)) / (r(1,1) - r(2,1));
    yi = r(1,1)*xi + r(1,2);

    plot(xi, yi, 'ko', 'MarkerSize', 5, 'MarkerFaceColor', 'g');

    printf("\nAs retas se interceptam no ponto (%.2f, %.2f)\n", xi, yi);
endif
hold off;

```

## q4.m

```
%1)
#y=-3+x
#y=-2x-1
xgraf= -6:6;
y1 = -3+xgraf;
y2 = -2.*xgraf-1;

figure
subplot(2, 2, 1)
plot(xgraf, y1, 'r-'); hold on
plot(xgraf, y2, 'b-')
grid on
title('Funções no gráfico')
axis([min(xgraf) max(xgraf), min([min(y1),min(y2)]), max([max(y1),max(y2)])])

%2)
x = -10:10;
printf('Digite os parâmetros de duas retas:\n');
for i =1:2
    printf("Reta %d:\n", i);
    r(i, 1) = input('Digite A (inclinação): ');
    r(i, 2) = input('Digite B (constante): ');
endfor
y1 = r(1,1)*x + r(1,2);
y2 = r(2,1)*x + r(2,2);

subplot(2, 2, 2)
plot(x, y1, 'b')
grid on
axis equal
title('Função dada 1')
subplot(2, 2, 3)
plot(x, y2, 'r')
grid on
axis equal
title('Função dada 2')
```

## q5.m

```
%1)
#o loop executa se x>z, independente de qualquer outra coisa
#o loop executa se x>y ao mesmo tempo que z!=y, independente de qualquer outra coisa

%2)
#o loop while será executado 4 vezes, quando x=1, x=2, x=3 e x=4.
```

## q6.m

```
k=1;
n = input("numero de linhas: ");
for i=1:n
k=i;
for j=1:n
printf("%i \t",k);
k=k+i;
end
printf("\n");
end
```

## q7.m

%q7 meteus

```
n = -1;
while n < 1 || n > 8
    n = input('Digite n, sendo 0 > n < 9: ');
endwhile
m = [];
for linha = 1:n
    valor_inicial = 1 + sum(2:linha); % Calcula o primeiro valor da linha
    incremento = linha + 1;          % O incremento para esta linha

    for coluna = 1:n
        if linha == 1 && coluna == 1
            m(linha, coluna) = 1; % Posição (1,1) sempre 1
        else
            m(linha, coluna) = valor_inicial + (coluna - 1) * incremento;
        end
    end
end
m(n, n) = n^2 + n -1;

disp('Quadrado gerado:');
for linha = 1:n
    for coluna = 1:n
        printf('%3d ', m(linha, coluna));
    end
    printf("\n");
end
```

## q8.m

%q8 mateo



```

#a)
N = 1;
matriz = [];
while N <= 1 || N > 8
    N = input('Digite o valor de N (2 a 8): ');
endwhile
matriz = randi([0, 10], 3, N);

soma_impares = sum(matriz(mod(matriz, 2) == 1));

disp('Matriz gerada:');
disp(matriz);

printf('Soma dos elementos ímpares: %d\n', soma_impares);

```

```

#b)
n = 1;
matriz = [];
while n <= 2
    n = input('Digite o valor de n (n > 2): ');
endwhile
vetor = randi([1, n], n, 1);

%arredonda o numero de linhas (n/3) pra cima
num_linhas = ceil(n / 3);

matriz = zeros(num_linhas, 3);
matriz(1:n) = vetor;

disp('Vetor original (n×1):');
disp(vetor);

disp('Matriz reorganizada (3 colunas):');
disp(matriz);

```

## q9.m

```

%2, 9 e 32 estão repetidas
%mateus leon
S = 0; i=1;
d = input('digite d: ');
a = input('digite a: ');
b = input('digite b: ');
for k=a:b
    x(i) = k;
    if mod(i,2)==d
        S = S + i;
    end
end

```

```

i+=1;
end
printf("S=%i\n",S);
i = 1:length(x);
plot(i, x);
grid on
#a)S=12
#b)S=4

```

## q10.m

```

S = 0;
n = input('Entre com n (inteiro): ');
m = input('Entre com m (<n) (inteiro): ');
i = 1;
for k = n:-1:m
    x(i) = k;
    S = S + k;
    y(i) = 2^k;
    i = i + 1;
end
plot(x, y, '.b', 'markersize', 10);

```

%a) saída do programa:  
 #um gráfico com x de 5 a 10;  
 #y = 2^x (32, 64, 128, 256, 512, 1024)  
 %b) i=7  
 %c) k=5  
 %d) y(1) = 1024, y(2) = 512, y(3) = 256

## q11.m

%essa questão é a questão 5 repetida

## q12.m

```

%Q12 (SOLUÇÃO DO FANTINI)
S = 0;
n=input('entre com n (inteiro): ');
m=input('entre com m (>n) (inteiro): ');
i=m-n+1;
for k=n:1:m
    x2(i) = k;
    S = S + k;
    y2(i) = k^2;
    i=i-1;
end
plot(y2,'.b','markersize',10)

```

#b) i=0 no final  
#c) k=12 no final  
#d)  $y_2(1) = 144$  |  $y_2(2) = 121$  |  $y_2(3) = 100$

## q13.m

%Q13 (SOLUÇÃO DO FANTINI)

```
x=-1;
while (x < 5)
    printf(" x = %i ",x);
    for i=1:10
        disp(i);
    end
    printf("\n");
    x=x+1;
endwhile
```

## q14.m

%14.1 (SOLUÇÃO DO FANTINI)

```
while (a > b && b > c && a != b)
    printf("%i %i %i\n",x,y,z);
    :
    :
endwhile
```

# Para que o while NÃO seja executado, o resultado da operação lógica tem que ser FALSO.

# Um conjunto de valores para que isso ocorra é: x = 10; y = 9; z = 10.

# Observe que apenas uma operação sendo falsa fará o resultado ser falso,  
# pois temos duas operações AND na expressão.

%14.2

```
x=-1;
while (x < 5)
    printf(" x = %i ",x);
    for i=1:10
        disp(i);
    end
    printf("\n");
    x=x+1;
endwhile
```

#Ela será executada seis vezes. Desde x=1 até x=4. Ou seja, para x=-1, 0,1,2,3,4. Quando x for 5 o loop #termina...

## q15.m

```
%q15 mateus leon
M = load("100m.dat");
M2 = M(7:end, :);
x = M2(:, 1);
y = M2(:, 2);

p2 = polyfit(x, y, 2);
yp2 = polyval(p2, x);
p5 = polyfit(x, y, 5);

figure
plot(x, yp2, 'b-'); hold on
plot(x, y, 'ro', 'MarkerSize', 1);
grid on

t0 = 0.06;
t = [2-t0, 5-t0, 9.7-t0, t0];
yt = polyval(p5, t)
mpors = yt./t
kmporhr = 3.6.*mpors
acelera = mpors./t
figure
plot(t, yt, 'ko');
grid on
```

## q16.m

```
%q16 leoncioBolas
printf("Dado o sistema de equações lineares:\n")
printf("ax + by = u\n");
printf("cx + dy = v\n\n");

printf("Dê os coeficientes (a,b,c,d) e os resultados (u,v):\n");
a = input('Digite a: ');
b = input('Digite b: ');
c = input('Digite c: ');
d = input('Digite d: ');
u = input('Digite u: ');
v = input('Digite v: ');

x = d/(a*d-b*c)*u - b/(a*d-b*c)*v;
y = -c/(a*d-b*c)*u + a/(a*d-b*c)*v;

printf("Os valores calculados são:\nx=%d e y=%d.\n\n", x, y);
printf("Agora, para plotar as equações:\n");
x0 = input("Digite um valor de início para x: ");
```

```

xf = input("Digite um valor final para x: ");
n = input("Digite o número de intervalos no vetor x: ");

x = linspace(x0, xf, n);
y1 = a.*x + b.*y;
y2 = c.*x + d.*y;

figure
hold on
plot(x, y1, 'r-', 'DisplayName', '1a função');
plot(x, y2, 'b-', 'DisplayName', '2a função');
legend();
title('Gráfico com as duas funções')
grid on

```

## q17.m

```

% Questão 17 - Lista_Varias_questoes_provas_2025.pdf
% Semana 06/08
% Lucas Vasconcelos Resende - 205B

m = input('Digite o valor de M, entre 2 e 5: ');
while m < 2 || m > 5
    m = input('VALOR INVALIDO, digite o valor de M, entre 2 e 5: ');
endwhile
matriz = zeros(5, m);
dados = load('Numeros.txt');
indice = 1;
for i = 1:5
    for j = 1:m
        matriz(i,j) = dados(indice);
        indice += 1;
    endfor
endfor
disp(matriz)
printf("\n");
asteriscos = 0;
for i = 1:5
    asteriscos = sum(matriz(i, :));
    for k = 1:asteriscos
        printf("*");
    endfor
    printf("\n");
endfor

```

## q18.m

```
%q18mateus
opcao18 = 1;
while opcao18 != 4
clear;
printf("=====\nMENU PRINCIPAL - escolha uma opcao (1 a 4)\n=====\n");
printf("(1) Poligonos\n(2) Vetores\n(3) Matrices\n(4) Sair\n");
opcao18 = input('opção escolhida: ');
clc;
switch opcao18
case 1
q18pt1
case 2
q18pt2
case 3
q18pt3
case 4
printf("Saindo da q18...\n");
opcao = 1;
return
endswitch
endwhile
```

### *q18pt1.m*

```
%q18.1
n = input('Digite o número de lados, maior que 2: ');
while n <= 2
n = input('Digite outro número de lados, maior que 2: ');
endwhile
l = input('Digite o tamanho do lado do polígono: ');
%b) perimetro
perimetro = l*n;

%c) area
raio = l/2*sin(pi/n);
apotema = raio*cos(pi/n);
area = perimetro*apotema/2;

%d) diametro
diametro = 2*raio;

%respostas:
printf("(b) perímetro: %d\nc) área: %d\nd) diâmetro do círculo circunscrito: %d\n", perimetro,
area, diametro);

%extras pq sou fodao
centro = zeros(1,2);
```

```
theta = linspace(0, 2*pi, 100);
xc = centro(1) + raio*cos(theta);
yc = centro(2) + raio*sin(theta);
```

```
figure
plot(xc, yc, 'r-');
hold on
```

```
poligonoang = linspace(0, 2*pi, n+1);
xp = centro(1) + raio * cos(poligonoang);
yp = centro(2) + raio * sin(poligonoang);
```

```
xp(length(xp)+1) = xp(1);
yp(length(yp)+1) = yp(1);
```

```
plot(xp, yp, '-o');
axis equal
grid minor on
```

*q18pt2.m*

```
N = 1;
while N < 2 || N > 20
    N = input("Digite a dimensão N dos vetores (1 < N < 21): ");
endwhile
```

```
printf("Digite as coordenadas do vetor A:\n");
A = zeros(1, N);
for i = 1:N
    A(i) = input(sprintf("A(%d) = ", i));
endfor
```

```
printf("Digite as coordenadas do vetor B:\n");
B = zeros(1, N);
for i = 1:N
    B(i) = input(sprintf("B(%d) = ", i));
endfor
```

```
% (b) Soma dos vetores
Soma = A + B;
printf("\nb) Soma dos vetores A + B:\n");
printf(["A = ", num2str(A), "\n"]);
printf(["B = ", num2str(B), "\n"]);
printf(["A + B = ", num2str(Soma), "\n"]);
```

```
% (c) Produto escalar
ProdEscalar = dot(A, B);
printf("c) Produto escalar A*B:\n");
```

```
printf(["A = ", num2str(A), "\n"]);
printf(["B = ", num2str(B), "\n"]);
printf(["A * B = ", num2str(ProdEscalar), "\n"]);
```

% (d) Cálculo do módulo (norma) de cada vetor

```
ModuloA = norm(A);
ModuloB = norm(B);
printf("d) Módulos:\n");
printf(["|A| = ", num2str(ModuloA), "\n"]);
printf(["|B| = ", num2str(ModuloB), "\n"]);
```

% (e) Maior elemento de cada vetor

```
MaxA = max(A);
MaxB = max(B);
printf("e) Maiores elementos:\n");
printf(["Vetor A: ", num2str(MaxA), "\n"]);
printf(["Vetor B: ", num2str(MaxB), "\n"]);
```

*q18pt3.m*

```
m = 1; n = 1;
while m < 3 || m > 12
    m = input("Digite o número de linhas m (2 < m < 13): ");
endwhile
while n < 3 || n > 12
    n = input("Digite o número de colunas n (2 < n < 13): ");
endwhile
```

```
M1 = randi([0, 9], m, n);
M2 = randi([0, 9], m, n);
```

```
printf("\nMatriz M1:\n");
disp(M1);
printf("Matriz M2:\n");
disp(M2);
```

```
%b
printf("\nb) M1 + M2:\n");
soma = M1 + M2;
disp(soma);
```

```
%c
printf("\nc) M1 - M2:\n");
subtracao = M1 - M2;
disp(subtracao);
```

```
%d
printf("\nd) Transpostas:\n");
printf("M1':\n");
```



```

disp(M1');
printf("M2':\n");
disp(M2');

%e
printf("\ne) multiplicação de matrizes:\n");
if n == m
    %pode as 2
    printf("M1 * M2:\n");
    mult1 = M1 * M2;
    disp(mult1);

    printf("M2 * M1:\n");
    mult2 = M2 * M1;
    disp(mult2);
else
    %verifica qual multiplicacao e possivel
    [m1, n1] = size(M1);
    [m2, n2] = size(M2);

    if n1 == m2
        printf("M1 * M2 é possível:\n");
        mult = M1 * M2;
        disp(mult);
    else
        printf("M1 * M2 não é possível.\n");
    endif

    if n2 == m1
        printf("M2 * M1 é possível:\n");
        mult = M2 * M1;
        disp(mult);
    else
        printf("M2 * M1 não é possível.\n");
    endif
endif
endif

```

## q19.m

```

%q19 mateus leon
ordem = 3;
p1 = [];
p2 = [];
printf("1o polinômio:\n")
for i = 1:ordem+1
    p1(i) = input(sprintf("Digite o %do coeficiente: ", i));
endfor
printf("\n2o polinômio:\n")

```

```

for i = 1:ordem+1
    p2(i) = input(sprintf("Digite o %do coeficiente: ", i));
endfor

p3 = p1+p2;
#raizes = roots(p3);
raizes = roots(p3);
printf("\nraízes do polinômio p3 (p1+p2):\n");
disp(raizes);

multp = p1.*p2;
printf("\npolinômio resultante de (p1*p2):\n");
for i = 1:ordem
    if multp(i) == 1
        printf("+x^%d ", ordem+1-i)
    else
        printf("%+dx^%d ", multp(i), ordem+1-i)
    endif
endfor
printf("%+d\n", multp(ordem+1));

```

## q20.m

```

% Questao 20 - Lista_Varias_questoes_provas_2025.pdf
% Semana 06/08
% Lucas Vasconcelos Resende - 205B

```

```

x1 = input('Digite a coordenada x: ');
y1 = input('Digite a coordenada y: ');
p = cell(4); %não precisa usar cell, usei pq quis
p{1} = [x1,y1];
p{2} = [-x1,y1];
p{3} = [x1,-y1];
p{4} = [-x1,-y1];
figure(1, 'Name', 'Questao 20');
hold on
grid on
plot(p{1}(1),p{1}(2), 'b', 'markersize', 12)
plot(p{2}(1),p{2}(2), 'b', 'markersize', 12)
plot(p{3}(1),p{3}(2), 'b', 'markersize', 12)
plot(p{4}(1),p{4}(2), 'b', 'markersize', 12)
plot(0,0, 'ok', 'markersize', 12)
plot([p{1}(1),p{2}(1)], [p{1}(2),p{2}(2)], 'g')
plot([p{3}(1),p{4}(1)], [p{3}(2),p{4}(2)], 'g')
plot([p{1}(1),p{4}(1)], [p{1}(2),p{4}(2)], 'r')
plot([p{2}(1),p{3}(1)], [p{2}(2),p{3}(2)], 'r')
axis([-x1-1, x1+1, -y1-1, y1+1])

```

## q21.m

% Questao 21 - Lista\_Varias\_questoes\_provas\_2025.pdf

% Semana 06/08

% Lucas Vasconcelos Resende - 205B

```
x1 = input('Digite a coordenada x: ');
y1 = input('Digite a coordenada y: ');
p = cell(4); %não precisa usar cell, usei pq quis
p{1} = [-x1,y1];
p{2} = [-x1,-y1];
p{3} = [x1,y1];
p{4} = [x1,-y1];
figure(1, 'Name', 'Questao 20');
hold on
grid on
title('Figura Questao 20: Pontos Simetricos');
xlabel('x');
ylabel('y');
plot(p{1}(1),p{1}(2), 'k', 'markersize', 12)
plot(p{2}(1),p{2}(2), 'k', 'markersize', 12)
plot(p{3}(1),p{3}(2), 'k', 'markersize', 12)
plot(p{4}(1),p{4}(2), 'k', 'markersize', 12)
plot([p{1}(1),p{4}(1)], [p{1}(2),p{4}(2)], 'b')
plot([p{2}(1),p{3}(1)], [p{2}(2),p{3}(2)], 'b')
axis([-10, 10, -10, 10])
```

## q22.m

% Questão 22 - Lista\_Varias\_questoes\_provas\_2025.pdf

% Semana 06/08

% Lucas Vasconcelos Resende - 205B

%inputs

```
A = input('Digite o valor de A: ');
lambda = input('Digite o valor de lambda: ');
B = input('Digite o valor de B: ');
f = input('Digite o valor de f: ');
t0 = input('Digite o valor de t0: ');
tf = input('Digite o valor de tf: ');
deltat = input('Digite o valor de deltat: ');
%fazer as funcoes
t = t0:deltat:tf;
g = A*exp(-lambda.*t);
h = B*sin(2*pi*f.*t);
```

```

z = g.*h;
%plots
figure(1, 'Name', 'Questão 22')
subplot(2,2,1)
plot(t, g)
grid on
title('g(t) = Ae-lambda*t');
xlabel('t');
ylabel('g');
subplot(2,2,2)
plot(t, h)
grid on
title('h(t) = Bsen(2*pi*f*t)');
xlabel('t');
ylabel('h');
subplot(2,2,3)
plot(t, z)
grid on
title('z(t) = g(t)*h(t)');
xlabel('t');
ylabel('z');

```

## q23.m

```

% Questão 23 - Lista_Varias_questoes_provas_2025.pdf
% Semana 06/08
% Lucas Vasconcelos Resende - 205B

theta = input('digite o angulo theta: ');
radiano = theta*pi/180;
real = cos(radiano);
p = input('digite a precisão: ');
aproximado = 0;
termo = 1;
k = 0;
while(abs(termo) >= p)
    termo = ((-1)^k) * (radiano^(2*k)) / factorial(2*k);;
    aproximado += termo;
    k += 1;
endwhile
%resultados
printf("Cosseno real: %f\n", real);
printf("Cosseno aproximado com a equação: %f\n", aproximado);
printf("Erro: %.10f\n", abs(real - aproximado));
printf("Número de termos usados: %d\n", k);

```

## q24.m

```
% Questão 24 - Lista_Varias_questoes_provas_2025.pdf
% Semana 06/08
% Lucas Vasconcelos Resende - 205B

n = input('Digite o valor de n, (entre 6 e 15): ');
while n < 6 || n > 15
    n = input('VALOR INVALIDO, digite o valor de n, (entre 6 e 15): ');
endwhile
for i = 1:n
    for j = 1:n
        if i == 1 || i == n
            printf(" * ");
        elseif j == 1 || j == n
            printf(" * ");
        elseif i + j == n + 1
            printf(" * ");
        else
            printf(" ");
        endif
    endfor
    printf("\n");
endfor
```

## q25.m

```
% Questão 25 - Lista_Varias_questoes_provas_2025.pdf
% Semana 06/08
% Lucas Vasconcelos Resende - 205B
```

```
M = load('provaA.dat');
x = M(:, 1);
y = M(:, 4);
n = input('Digite o grau do polinomio: ');
figure(1, 'Name', 'Questão 25')
hold on
plot(x, y, 'ok', 'markersize', 10);
p = polyfit(x, y, n);
y2 = polyval(p, x);
plot(x, y2)
grid on
```

## q26.m

% Questão 26 - Lista\_Varias\_questoes\_provas\_2025.pdf  
% Semana 06/08  
% Lucas Vasconcelos Resende - 205B

```
m = input('Digite o valor de m: ');
n = input('Digite o valor de n: ');
M = zeros(n, m);
for i = 1:n
    for j = 1:m
        if i ~= j
            M(i,j) = (i+j)^i;
        else
            M(i,j) = i;
        endif
    endfor
    printf("\n");
endfor
%(1)
printf("\nMatriz:\n");
disp(M);
printf("\nMatriz Transposta:\n");
disp(M');
%(2)
v = zeros(n, 1);
for i = 1:n
    for j = 1:m
        v(i) += M(i,j);
    endfor
endfor
printf("Vetor v: ");
disp(v);
%(3)
soma = 0;
for i = 1:n
    for j = 1:m
        soma += M(i,j);
    endfor
endfor
printf("Soma de todos os elementos do vetor: ");
disp(soma);
```

## q27.m

% Questão 27 - Lista\_Varias\_questoes\_provas\_2025.pdf  
% Semana 06/08  
% Lucas Vasconcelos Resende - 205B

```

x1 = input('Digite o valor para x do ponto1: ');
y1 = input('Digite o valor para y do ponto1: ');
p = cell(9);
p{1} = [x1, y1];
p{2} = [-x1, y1];
p{3} = [-x1, -y1];
p{4} = [x1, -y1];
p{5} = [0, 0];
figure(1, 'Name', 'Questão 27')
hold on
for i = 1:5
    plot(p{i}(1),p{i}(2), 'k', 'markersize', 12)
endfor
%linhas
for i = 1:5
    plot([p{i}(1),0],[p{i}(2),0], 'b');
endfor
%pontos medios
p{6} = [x1/2, y1/2];
p{7} = [-x1/2, y1/2];
p{8} = [-x1/2, -y1/2];
p{9} = [x1/2, -y1/2];
for i = 6:9
    plot(p{i}(1),p{i}(2), '+', 'markersize', 12)
endfor
axis([-x1-2, x1+2, -y1-2, y1+2]);

```

## q28.m

```

% Questão 28 - Lista_Varias_questoes_provas_2025.pdf
% Semana 06/08
% Lucas Vasconcelos Resende - 205B

```

```

n = input('Digite o valor de n: ');
k = input('Digite o valor de k: ');
soma1 = 0;
for i = -n:n
    if i != 0
        termo1 = (2*i+1)/i;
        soma1 += termo1;
    end
endfor
soma2 = 0;
for i = -k:k
    if i != 0
        termo2 = (i+1)/(k*i);
        soma2 += termo2;
    end
endfor

```

```

    endif
endfor
soma = soma1 + soma2;
printf("O somatório é igual a %.2f", soma);

```

## q29.m

```

%q29 mateus leon
n = 1;
while n <= 5 || n >= 17
n = input('Digite n (5 < n < 17): ');
endwhile

iaux = n-1;
for i = 1:n
    for k = iaux:1:n-2
        printf("**")
    endfor
    printf(" ")
    for j = iaux:-1:1
        printf("**")
    endfor
    iaux--;
    printf("\n")
endfor

```

## q30.m

```

%q30 mateus lseons
x = [-6:6];
y1 = ones(1, 13).*-2;
y2 = 2.*x-1;

n = 0;
while n < 1 || n > 10
n = input("Digite quantas funções afim deseja plotar (mín 1 máx 10): ");
endwhile
for i = 1:n
    printf("\n- Função afim %d -\n", i)
    for j = 1:2
        f(i, j) = input(sprintf('Digite o %do coeficiente: ', j));
    endfor
    y(i, :) = polyval(f(i, :), x);
endfor

figure('Name', 'Questão 30')
hold on
for i = 1:n

```



```

    plot(x, y(i, :), 'DisplayName', sprintf("F%d (input)", i));
endfor
plot(x, y1, 'r-', 'DisplayName', 'F1 (original)');
plot(x, y2, 'b-', 'DisplayName', 'F2 (original)');
legend();
grid on
title('Gráfico com 2 funções originais e todas as dadas no input')
axis([min(x) max(x) min(x) max(x)])

```

## q31.m

```

% Questão 31 - Lista_Varias_questoes_provas_2025.pdf
% Semana 06/08
% Lucas Vasconcelos Resende - 205B

clear; clc;
n = input('Digite o valor de n,(entre 0 e 9): ');
while n < 0 || n > 9
    n = input('VALOR INVALIDO, digite o valor de n,(entre 0 e 9): ');
endwhile
for i = 1:n
    for j = 1:n
        if j == 1
            printf("%d ",i);
        else
            printf("%d ", i+n*(j-1));
        endif
    endfor
    printf("\n");
endfor

```

## q32.m

#a q2, q9 e q32 estão repetidas - na 2 tem o código do Vasco e na 9 tem o código do Leon

## q33.m

%q33 mateus leon

```

n = 1;
while n < 1 || n > 12 || mod(n, 2) != 0
    n = input('Digite n (n deve ser um inteiro par de 1 a 12):');
endwhile

jaux = 1;
for i = 1:n
    %espaço antes (formatação)

```

```

for k = 1:i
    printf(" ")
endfor

%se o número tiver 2 dígitos precisa de cada vez mais espaços
if i > 10
    for k = 1:i-10
        printf(" ");
    endfor
endif

%print e calculo da soma
soma = 0;
for j = 1:n
    printf("%d ", j);
    soma += j;
endfor
printf(" %d\n", soma);
jaux++;
endfor

```

## q34.m

%q34 noel suetam

```

p = [];
n = input('Digite o número de linhas da matriz: ');
m = input('Digite o número de colunas da matriz: ');

soma = 0;
for i = 1:n
    for j = 1:m
        if i == j
            %n aleatorio entre 0 e 20 (0<n<20)
            p(i, j) = randi(19);
        else
            p(i, j) = i + j;
        endif
        soma += p(i, j);
    endfor
endfor

printf("\nSoma dos elementos: %d\n", soma);
printf("\nMatriz:\n");
disp(p);
printf("\nMatriz transposta:\n")
disp(p')
if n == m

```

```

if det(p) != 0
    printf("\nA matriz quadrada dada é invertível. Matriz inversa:\n")
    disp(inv(p))
else
    printf("\nA matriz quadrada dada não é invertível.\n")
endif
else
    printf("\nEssa matriz não é quadrada, portanto, não tem inversa.\nContudo, ela pode ter uma pseudoinversa. ");
    check = input("Deseja checar? (s/n): ", "s");
    if check == "s"
        try
            pseudoinv = pinv(p);
            printf("\nA matriz retangular possui pseudoinversa:\n");
            disp(pseudoinv);
        catch
            printf("\nA matriz retangular não possui pseudoinversa.\n")
        end_try_catch
    endif
endif
endif

```

## q35.m

%q35 leon

```

angd = input('Digite um ângulo em graus: ');
n=1;
while n < 2
    n = input('Digite o número de iterações (>1): ');
endwhile
ang = deg2rad(angd);
seno = 0;
for i = 0:n-1
    seno += ((-1)^i*ang^(2*i+1)) / factorial(2*i+1);
endfor
senoreal = sin(ang);

printf("seno aproximado: %d\nseno real: %d\nerro: %d\n", seno, senoreal, senoreal-seno)

```

$$\sin(\theta) = \sum_{n=0}^{\infty} (-1)^n \frac{\theta^{2n+1}}{(2n+1)!}$$

## q35-2.m

%questao 35

%March Antony Paul Leon

```

clc;

x=input('Digite a coordenada x: ');
y=input('Digite a coordenada y: ');
P1=[-x,0];
P2=[x,0];
P3=[0,y];
P4=[0,-y];

figure(1);
hold on;
plot(P1(1),P1(2),'.k','markersize',15);
plot(P2(1),P2(2),'.k','markersize',15);
plot(P3(1),P3(2),'.k','markersize',15);
plot(P4(1),P4(2),'.k','markersize',15);

plot([P1(1),P3(1)],[P1(2),P3(2)],'-b','linewidth',2);
plot([P1(1),P4(1)],[P1(2),P4(2)],'-b','linewidth',2);
plot([P2(1),P3(1)],[P2(2),P3(2)],'-b','linewidth',2);
plot([P2(1),P4(1)],[P2(2),P4(2)],'-b','linewidth',2);
hold off;
title('losango');
xlabel('x');
ylabel('y');
grid on;
axis equal;

```

## q36.m

%questao 36

%March Antony Paul Leon

```

clc;

p = 1;
R=0;
S=0;
termo=0;
while (p != 0)
    printf("=====MENU=====\\n");
    printf("Escolha sua acao:\\n(0) Para encerrar\\n(1) Para entrar com valor de N\\n(2) Para  

    calcular o valor do somat'orio para N termos.\\n");
    p = input('Insira a acao desejada: ');
    close all
    switch(p)
        case 1
            n=input('Digite o valor de n: ');

```

```

case 2
for i=1:n
    termo=(2*n-1)*(sin(pi/(i+1)))/(i*pi);
    if(mod(i,2)==0)
        R=termo;
    else
        R=-termo;
    endif
    S+=R;
endfor
printf("Valor do somatorio: %d\n", S);
endswitch
endwhile

```

## q37.m

%questao 37

%March Antony Paul Leon

```

clc;

```

```

p = 1;
S=0;
termo=0;
while (p != 0)
    printf("=====MENU=====\\n");
    printf("Escolha sua acao:\\n(0) Para encerrar\\n(1) Para entrar com valor de N0 e Nf\\n(2) Para calcular o valor do somatorio para N termos.\\n");
    p = input('Insira a acao desejada: ');
    close all
    switch(p)
        case 1
            N0=input('Digite o valor de N0: ');
            Nf=input('Digite o valor de Nf: ');
            while(Nf<N0)
                printf("Valor invalido!\\n");
                N0=input('Digite o valor de N0: ');
                Nf=input('Digite o valor de Nf: ');
            end
        case 2
            for i=N0:Nf
                termo=(i-1)/(2+i);
                S+=termo;
            endfor
            printf("Valor do somatorio: %d\\n", S);
        endswitch
    endwhile

```

## q38.m

%questao 38

%March Antony Paul Leon

clc;

p = 1;

S=0;

termo1=0;

termo2=0;

while (p != 0)

    printf("=====MENU=====\\n");

    printf("Escolha sua acao:\\n(0) Para encerrar\\n(1) Para entrar com valor de N0 e Nf\\n(2)

Para calcular o valor do somatorio para N termos.\\n");

    p = input('Insira a acao desejada: ');

    close all

    switch(p)

        case 1

            N0=input('Digite o valor de N0: ');

            Nf=input('Digite o valor de Nf: ');

            while(Nf<N0)

                printf("Valor invalido!\\n");

                N0=input('Digite o valor de N0: ');

                Nf=input('Digite o valor de Nf: ');

            end

        case 2

            for i=N0:Nf

                if i ~= 1 && i ~= -2

                    termo1=(2\*i+1)/(i-1);

                    termo2=(2-i)/(i+2);

                    S+=termo1+termo2;

                endif

            endfor

            printf("Valor do somatorio: %d\\n", S);

        endswitch

    endwhile

## q39.m

% Questão 39 - Lista\_Varias\_questoes\_provas\_2025.pdf

% Semana 06/08

% Lucas Vasconcelos Resende - 205B

```

opcao = 1;
while (opcao != 0)
    printf("(1) - Numeros Pares\n(2) - Soma N termos\n(3) - Loop com x e x^2\n(4) - Voltar ao
menu principal.\n");
    opcao = input('digite o exercício ou 4 para sair: ');
    close all
    switch(opcao)
        case 1
            %(1) - Numeros Pares
            n = input('Digite um numero par: ');
            while mod(n, 2) != 0
                n = input('VALOR INVALIDO, digite um numero par: ');
            endwhile
            for i = 0:n
                if i == n
                    printf("%d\n\n", i);
                elseif mod(i, 2) == 0
                    printf("%d, ", i);
                endif
            endfor
        case 2
            %(2) - Soma N termos
            n = input('Digite um numero: ');
            soma = 0;
            for i = 1:n
                if mod(i,2) == 1
                    termo = i*cos(pi/i);
                else
                    termo = i*sin(pi/i);
                endif
                soma += termo;
            endfor
            printf("Somatório = %f\n\n", soma);
        case 3
            %(3) - Loop com x e x^2
            do
                x = input('Digite o valor de x: ');
                printf("x = %.2f\nx^2 = %.2f\n", x, x^2);
            until x < 0
        case 4
            %(4) - Voltar ao menu principal
            break;
    endswitch
endwhile

```

**q40.m**

% Questão 40 - Lista\_Varias\_questoes\_provas\_2025.pdf  
% Semana 06/08  
% Lucas Vasconcelos Resende - 205B

```
opcao40 = 1;
while (opcao40 != 5)
    printf("(1)somatório S1 (pedindo n0 e nf)\n(2)somatório S2 (pedindo i0 e if)\n(3)soma S1 + S2\n(4)subtração S1 - S2\n");
    opcao40 = input('digite o exercício ou 5 para sair: ');
    switch(opcao40)
        case 1
            printf("-----\n");
            printf("somatório S1 (pedindo n0 e nf)\n");
            printf("-----\n");
            n0 = input('Digite o valor de n0: ');
            nf = input('Digite o valor de nf: ');
            soma1 = 0;
            for n = n0:nf
                termo = 1/(n)^3;
                soma1 += termo;
            endfor
            printf("Somatório S1, com n0 = %f, e nf = %f, é igual a:\n%f\n\n", n0, nf, soma1);
        case 2
            printf("-----\n");
            printf("somatório S2 (pedindo i0 e if)\n");
            printf("-----\n");
            i0 = input('Digite o valor de i0: ');
            ifs = input('Digite o valor de if: ');
            soma2 = 0;
            for i = i0:ifs
                termo = (i+1)/(i-2);
                soma2 += termo;
            endfor
            printf("Somatório S2, com i0 = %f, e if = %f, é igual a:\n%f\n\n", i0, ifs, soma2);
        case 3
            printf("-----\n");
            printf("      soma S1 + S2      \n");
            printf("-----\n");
            n0 = input('Digite o valor de n0: ');
            nf = input('Digite o valor de nf: ');
            soma1 = 0;
            for n = n0:nf
                termo = 1/(n)^3;
                soma1 += termo;
            endfor
            i0 = input('Digite o valor de i0: ');
            ifs = input('Digite o valor de if: ');
            soma2 = 0;
```



```

    for i = i0:ifs
        termo = (i+1)/(i-2);
        soma2 += termo;
    endfor
    soma = soma1 + soma2;
    printf("A soma entre o S1 e S2, com n0 = %f, nf = %f, i0 = %f, if = %f, é:\n%f\n\n", n0, nf,
i0, ifs, soma);
    case 4
        printf("-----\n");
        printf("    subtração S1 - S2    \n");
        printf("-----\n");
        n0 = input('Digite o valor de n0: ');
        nf = input('Digite o valor de nf: ');
        soma1 = 0;
        for n = n0:nf
            termo = 1/(n)^3;
            soma1 += termo;
        endfor
        i0 = input('Digite o valor de i0: ');
        ifs = input('Digite o valor de if: ');
        soma2 = 0;
        for i = i0:ifs
            termo = (i+1)/(i-2);
            soma2 += termo;
        endfor
        subtracao = soma1 - soma2;
        printf("A subtracao entre o S1 e S2, com n0 = %f, nf = %f, i0 = %f, if = %f, é:\n%f\n\n",
n0, nf, i0, ifs, subtracao);
    endswitch
endwhile

```

## q41.m

% Questão 41 - Lista\_Varias\_questoes\_provas\_2025.pdf

% Semana 06/08

% Lucas Vasconcelos Resende - 205B

```

opcao41 = 1;
while (opcao41 != 4)
    printf("(1)soma S1 (limites dados pelo usuário)\n");
    printf("(2)soma S2 (limites dados pelo usuário)\n");
    printf("(3)R = S1 + S2\n");
    opcao41 = input('digite o exercício ou 4 para sair: ');
    switch(opcao41)
        case 1
            printf("-----\n");
            printf("soma S1 (limites dados pelo usuário)\n");

```

```

printf("-----\n");
n0 = input('Digite o valor de n0: ');
nf = input('Digite o valor de nf: ');
while nf < n0
    n0 = input('VALOR INVALIDO, digite o valor de n0: ');
    nf = input('VALOR INVALIDO, digite o valor de nf: ');
endwhile
soma1 = 0;
for n = n0:nf
    termo = (n+1)*(sin((n*pi)/(n+1))/(n*pi));
    soma1 += termo;
endfor
printf("Somatório S1, com n0 = %f, e nf = %f, é igual a:\n%f\n\n", n0, nf, soma1);
case 2
printf("-----\n");
printf("soma S2 (limites dados pelo usuário)\n");
printf("-----\n");
i0 = input('Digite o valor de i0: ');
ifs = input('Digite o valor de if: ');
while ifs < i0
    i0 = input('VALOR INVALIDO, digite o valor de i0: ');
    ifs = input('VALOR INVALIDO, digite o valor de if: ');
endwhile
soma2 = 0;
for i = i0:ifs
    termo1 = (2*i-1)/(2*i+1);
    termo2 = (i+1)/(i-1);
    soma2 += termo1 + termo2;
endfor
printf("Somatório S2, com i0 = %f, e if = %f, é igual a:\n%f\n\n", i0, ifs, soma2);
case 3
printf("-----\n");
printf("      R = S1 + S2      \n");
printf("-----\n");
n0 = input('Digite o valor de n0: ');
nf = input('Digite o valor de nf: ');
while nf < n0
    n0 = input('VALOR INVALIDO, digite o valor de n0: ');
    nf = input('VALOR INVALIDO, digite o valor de nf: ');
endwhile
soma1 = 0;
for n = n0:nf
    termo = (n+1)*(sin((n*pi)/(n+1))/(n*pi));
    soma1 += termo;
endfor
i0 = input('Digite o valor de i0: ');
ifs = input('Digite o valor de if: ');
while ifs < i0

```

```

        i0 = input('VALOR INVALIDO, digite o valor de i0: ');
        ifs = input('VALOR INVALIDO, digite o valor de if: ');
    endwhile
    soma2 = 0;
    for i = i0:ifs
        termo1 = (2*i-1)/(2*i+1);
        termo2 = (i+1)/(i-1);
        soma2 += termo1 + termo2;
    endfor
    R = soma1 + soma2;
    printf("A soma entre o S1 e S2, com n0 = %f, nf = %f, i0 = %f, if = %f, é:\n%f\n\n", n0,
nf, i0, ifs, R);
endswitch
endwhile

```

## q42.m

```

% Questão 42 - Lista_Varias_questoes_provas_2025.pdf
% Semana 06/08
% Lucas Vasconcelos Resende - 205B

```

```

opcao42 = 1;
function[resultado, vetor] = raizes(a, b, c)
    vetor = zeros(1, 2);
    delta = b^2 - 4*(a)*(c);
    if delta < 0
        resultado = 0;
    elseif delta == 0
        printf("A funcao tem uma raiz real\n");
        x1 = (-(b)+sqrt(delta))/(2*a);
        vetor(1) = x1;
        vetor(2) = x1;
        resultado = 1;
    else
        printf("A funcao tem duas raizes reais\n");
        x1 = (-(b)+sqrt(delta))/(2*a);
        x2 = (-(b)-sqrt(delta))/(2*a);
        vetor(1) = x1;
        vetor(2) = x2;
        resultado = 1;
    endif
endfunction
function soma = somatorio(n0, nf, n)
    soma = 0;
    for i = n0:nf
        termo = (n^i)/factorial(n - 1);
        soma += termo;
    endfor

```

```

endfunction
while (opcao42 != 4)
    printf("(1)chama a funcao para calcular as raizes reais de uma equacao de segundo grau\n");
    printf("(2)chama a funcao para calcular o somatorio\n");
    opcao42 = input('digite o exercício ou 4 para sair: ');
    switch(opcao42)
        case 1
            printf("-----\n");
            printf("    equacao de segundo grau  \n");
            printf("-----\n");
            vetor = zeros(1,2);
            a = input('Digite o valor de A: ');
            while a == 0
                a = input('Se A for 0, não é uma funcao de segundo grau, digite o valor de A: ');
            endwhile
            b = input('Digite o valor de B: ');
            c = input('Digite o valor de C: ');
            [resultado, vetor] = raizes(a, b, c);
            if resultado == 1
                printf("O valor de x1 e x2 é, respectivamente:\n");
                disp(vetor);
            else
                printf("não tem raizes reais\n");
            endif
        case 2
            printf("-----\n");
            printf("    somatorio      \n");
            printf("-----\n");
            n0 = input('Digite o valor de n0: ');
            nf = input('Digite o valor de nf: ');
            n = input('Digite o valor de n: ');
            soma = somatorio(n0, nf, n);
            printf("Somatório S1, com n0 = %f, e nf = %f, e n = %f, é igual a:\n%f\n\n", n0, nf, n,
soma);
        endswitch
    endwhile
endwhile

```

## q43.m

```

%q43 deepseek fez
% Script de Operações Matemáticas
while true
    % Menu Principal
    disp('-----');
    disp('MENU PRINCIPAL - escolha uma opção (1 a 4)');
    disp('-----');
    disp('(1) Figuras');

```

```

disp('(2) Polinômios');
disp('(3) Matrizes');
disp('(4) Sair');
opcao = input('Escolha uma opção: ');

switch opcao
case 1 % Figuras
    while true
        disp('-----');
        disp('MENU DE FIGURAS - escolha uma opção (1 a 4)');
        disp('-----');
        disp('(1) Retângulos');
        disp('(2) Losangos');
        disp('(3) Triângulos');
        disp('(4) Voltar');
        opcao_fig = input('Escolha uma opção: ');

        switch opcao_fig
        case 1 % Retângulo
            n = input('Digite o tamanho do lado do retângulo (n >= 2): ');
            if n < 2
                disp('Tamanho inválido! Deve ser >= 2');
                continue;
            end
            disp('Retângulo:');
            for i = 1:n
                for j = 1:n
                    if i == 1 || i == n || j == 1 || j == n
                        fprintf('*');
                    else
                        fprintf(' ');
                    end
                end
                fprintf('\n');
            end

        case 2 % Losango
            while true
                n = input('Digite a altura do losango (número ímpar >= 3): ');
                if mod(n, 2) == 1 && n >= 3
                    break;
                else
                    disp('Altura inválida! Deve ser ímpar e >= 3');
                end
            end
            disp('Losango:');
            meio = ceil(n/2);
            for i = 1:n

```

```

        espacos = abs(meio - i);
        asteriscos = n - 2*espacos;
        fprintf('%s', repmat(' ', 1, espacos));
        fprintf('%s', repmat('*', 1, asteriscos));
        fprintf('\n');
    end

case 3 % Triângulo
    n = input('Digite a altura do triângulo (n >= 1): ');
    if n < 1
        disp('Altura inválida! Deve ser >= 1');
        continue;
    end
    disp('Triângulo:');
    for i = 1:n
        espacos = n - i;
        asteriscos = 2*i - 1;
        fprintf('%s', repmat(' ', 1, espacos));
        fprintf('%s', repmat('*', 1, asteriscos));
        fprintf('\n');
    end

case 4 % Voltar
    break;

otherwise
    disp('Opção inválida!');
end
end

case 2 % Polinômios
    coeficientes = [];
    grau = 0;

    while true
        disp('-----');
        disp('MENU DE POLINÔMIOS - escolha uma opção (a-e)');
        disp('-----');
        disp('a) Ler os dados do polinômio');
        disp('b) Mostrar p(x)');
        disp('c) Coeficiente do maior grau');
        disp('d) Soma das raízes');
        disp('e) Produto das raízes');
        disp('f) Voltar');
        opcao_poli = input('Escolha uma opção: ', 's');

        switch lower(opcao_poli)
            case 'a' % Ler polinômio

```

```

    grau = input('Digite o grau do polinômio (n >= 0): ');
    if grau < 0
        disp('Grau inválido!');
        continue;
    end
    disp('Digite os coeficientes do polinômio (do a0 ao an):');
    coeficientes = zeros(1, grau+1);
    for i = 1:grau+1
        coeficientes(i) = input(sprintf('a%d = ', i-1));
    end

case 'b' % Mostrar p(x)
    if isempty(coeficientes)
        disp('Polinômio não definido!');
        continue;
    end
    fprintf('p(x) = ');
    for i = 1:length(coeficientes)
        if coeficientes(i) ~= 0
            if i == 1
                fprintf('%g', coeficientes(i));
            else
                if coeficientes(i) > 0
                    fprintf(' + %gx^%d', coeficientes(i), i-1);
                else
                    fprintf(' - %gx^%d', abs(coeficientes(i)), i-1);
                end
            end
        end
    end
    fprintf('\n');

case 'c' % Coeficiente maior grau
    if isempty(coeficientes)
        disp('Polinômio não definido!');
        continue;
    end
    fprintf('Coeficiente do maior grau: %g\n', coeficientes(end));

case 'd' % Soma das raízes
    if isempty(coeficientes) || grau == 0
        disp('Polinômio não definido ou grau zero!');
        continue;
    end
    if grau < 2
        disp('Polinômio de grau 1 não tem soma de raízes definida');
    else
        soma = -coeficientes(end-1)/coeficientes(end);

```

```

        fprintf('Soma das raízes: %g\n', soma);
    end

    case 'e' % Produto das raízes
        if isempty(coeficientes) || grau == 0
            disp('Polinômio não definido ou grau zero!');
            continue;
        end
        produto = ((-1)^grau)*coeficientes(1)/coeficientes(end);
        fprintf('Produto das raízes: %g\n', produto);

    case 'f' % Voltar
        break;

    otherwise
        disp('Opção inválida!');
    end
end

case 3 % Matrizes
    disp('Questão das matrizes não existe');

case 4 % Sair
    disp('Encerrando o programa...');
    break;

otherwise
    disp('Opção inválida!');
end
end

```

## q44.m

```

%q44 leblon
opt44 = -1;
function q1(n)
    for i = 1:n
        for j = 1:i
            printf("*");
        endfor
        printf("\n")
    endfor
    for i = n-1:-1:1
        for j = 1:i
            printf("*");
        endfor
        printf("\n")
    endfor
endfunction

```



```
function somatorio = q2(n0, nf)
    somatorio = 0;
    for i = n0:nf
        somatorio += (i+1)/(2*i+5);
    endfor
endfunction
```

```
while opt44 != 0
    opt44 = input('Digite qual das 2 funções deseja chamar (1, 2 ou 0 para sair): ');
    switch opt44
        case 1
            n = -1;
            while n < 1
                n = input('Digite n > 0: ');
            endwhile
            q1(n);
        case 2
            n0 = 2;
            nf = 1;
            while n0 > nf
                printf("Digite os limites do somatório:\n")
                n0 = input('n0 (n inicial): ');
                nf = input('nf (n final, maior ou igual a n0): ');
                if n0 > nf
                    printf("n0 deve ser menor ou igual a nf!\n");
                endif
            endwhile
            somat = q2(n0, nf);
            printf("resultado: %d\n", somat);
        case 0
            printf("saindo...\n")
        otherwise
            printf("Digite uma opção válida!\n")
    endswitch
endwhile
```

## q45.m

```
%q45 lebron
n = 2;
while mod(n, 2) != 1
    n = input('Digite n (ímpar): ');
endwhile

for i = 1:2:n
    for k = n:-1:i
        printf(" ");
```

```

endfor
for j = 1:i
    printf("%d ", j);
endfor
printf("\n");
endfor

```

## q46.m

%q46 The Bronze Age (Lebron James)

```

n = 2;
while mod(n, 2) != 1
    n = input('Digite n (ímpar): ');
endwhile

```

```

for i = 1:2:n

```

```

    if n < 10 %formatação 1 dig
        for k = n:-1:i
            printf(" ");
        endfor
        for j = 1:i
            printf("%d ", n);
        endfor
    else %formatação 2 dig
        for k = n:-1:i
            printf(" ");
        endfor
        for j = 1:i
            printf("%d ", n);
        endfor
    endif
    printf("\n");
endfor

```

```

for i = n-2:-2:1

```

```

    if n < 10 %formatação 1 dig
        for k = n:-1:i
            printf(" ");
        endfor
        for j = 1:i
            printf("%d ", n);
        endfor
    else %formatação 2 dig
        for k = n:-1:i
            printf(" ");
        endfor
        for j = 1:i
            printf("%d ", n);
        endfor
    endif
endfor

```

```

        endfor
    endif
    printf("\n");
endfor

```

## q47.m

```

%q47 leon uwu
opt47 = -1;
while opt47 != 0
    opt47 = input('Digite qual dos 4 desenhos deseja ver (1-4 ou 0 para sair): ');
    switch opt47
        case 1
            n = input('Digite n: ');
            for i = 1:n
                for j = n+1-i:-1:1
                    printf("%d ", j)
                endfor
                printf("\n")
            endfor
        case 2
            n = input('Digite n: ');
            for i = 1:n
                k = n;
                for j = n+1-i:-1:1
                    printf("%d ", k);
                    k--;
                endfor
                printf("\n")
            endfor
        case 3
            n = input('Digite n: ');
            for i = 1:n
                k = n+1;
                for j = n+1-i:-1:1
                    printf("%d ", k-j);
                endfor
                printf("\n")
            endfor
        case 4
            n = input('Digite n: ');
            for i = 1:n
                k = 1;
                for j = n+1-i:-1:1
                    printf("%d ", k);
                    k++;
                endfor
                printf("\n")
            endfor
        default
            break;
    endswitch
endwhile

```

```

        endfor
    case 0
        printf("saindo...\n")
    otherwise
        printf("Digite uma opção válida!\n")
    endswitch
endwhile

```

## q48.m

```

% questao 48
% March Antony Paul Leon

```

```

clc;

n = input('Digite o valor de N: ');

total_linhas = 2*n - 1;

for i = 1:total_linhas
    if i < n
        for j = 1:i
            printf("*");
        endfor
    elseif i == n
        for j = 1:n
            printf("-");
        endfor
    else
        for j = 1:(total_linhas - i + 1)
            printf("*");
        endfor
    endif
    printf("\n");
endfor

```

## q49.m

```

%questao 49
%Marco Antonio P Leao

```

```

clc;

escolha = 1;
while (escolha != 0)
    printf("===== \n");
    printf("1. A\n2. B\n3. C\n4. D\n");

```

```

escolha = input('digite o exercício ou 0 para sair: ');
switch(escolha)
case 1
    for i = 1:10
        for j = 1:i
            printf("* ");
        endfor
        printf("\n");
    endfor
case 2
    for i = 1:10
        for j = 1: 10 - i + 1
            printf("* ");
        endfor
        printf("\n");
    endfor
case 3
    for i = 1:10
        for j = 1:i -1
            printf(" ");
        endfor
        for k = 1: 10 - i + 1
            printf(" *");
        endfor
        printf("\n");
    endfor
case 4
    for i = 1:10
        for j = 1: 10 - i
            printf(" ");
        endfor
        for k = 1:i
            printf(" *");
        endfor
        printf("\n");
    endfor
endswitch
endwhile

```

## q50.m

```

% Questão 50 - Lista_Varias_questoes_provas_2025.pdf
% Semana 06/08
% Lucas Vasconcelos Resende - 205B

```

```

m = input('Digite o valor de M, entre 2 e 5: ');

```

```

while m < 2 || m > 5
    m = input('VALOR INVALIDO, digite o valor de M, entre 2 e 5: ');
endwhile
matriz = zeros(5, m);
dados = load('Numeros.txt');
indice = 1;
for i = 1:5
    for j = 1:m
        matriz(i,j) = dados(indice);
        indice += 1;
    endfor
endfor
disp(matriz)
printf("\n");
asteriscos = 0;
for i = 1:5
    asteriscos = sum(matriz(i, :));
    printf("%d  ", asteriscos);
    for k = 1:asteriscos
        printf("*");
    endfor
    printf("\n");
endfor

```

## q51.m

```

%questao 51
%Marco Antonio P Leao

```

```

clc;

n=input('Digite o valor de n (n<=15): ');
if(n<=15)
    for i = 1:n
        printf('termo = %d\t',i);
        for j = 1:i
            printf("* ");
        endfor
        printf("\n");
    endfor
else
    disp('valor invalido para n!');
endif

```

## q52.m

```

%questao 52

```

```
%Marco Antonio P Leao
```

```
clc;
```

```
n = input("digite o valor (entre 3 e 9) do lado do triangulo ao contrario de asteriscos: ");
if((n >= 3)&&(n <= 9))
    for i = 1:n
        for j = 1:n
            if(i == 1 || i == n || j == 1 || j == n)
                printf("%d ",n);
            else
                printf(" ");
            end
        endfor
        printf("\n");
    endfor
else
    disp("Erro, o valor deve ser entre 3 e 9");
endif
```

## q53.m

```
% Questão 53 - Lista_Varias_questoes_provas_2025.pdf
% Semana 06/08
% Lucas Vasconcelos Resende - 205B
```

```
n = input('Digite o valor de N: ');
while mod(n,2) == 0
    n = input('VALOR INVALIDO, digite o valor de N: ');
endwhile
metade = (n-1)/2; %quantidade de linhas até o centro
disp(n);
for i = 0:n-1
    if(i == 0 || i == n-1)
        for k = 1:n-1
            printf("o");
        endfor
        printf(" ");
        for k = 1:n-1
            printf("o");
        endfor
    else
        if i <= metade
            quantidade = (n-1) - i;
            espaco = 2*i+1;
        else
            quantidade = i;
            espaco = 2*(n-i)-1;
        end
        for k = 1:quantidade
            printf(" ");
        endfor
        for k = 1:quantidade
            printf("o");
        endfor
        for k = 1:espaco
            printf(" ");
        endfor
        for k = 1:quantidade
            printf("o");
        endfor
    end
endfor
```

```

        quantidade = i;
        espaco = 2*(n-1-i)+1;
    endif
    for k = 1:quantidade
        printf("o");
    endfor
    for k = 1:espaco
        printf(" ");
    endfor
    for k = 1:quantidade
        printf("o");
    endfor
    endif
    printf("\n");
endfor

```

## q54.m

```

%questao 54
%Marco Antonio P Leao
clc;

```

```

er = input('Digite o valor da precisão desejada: ');
g=input('Digite o valor do grau: ');

```

```

x=g*pi/180;
erro = 1;
k = 0;
sin_aproximado = 0;
sin_real = sin(x);

```

```

while erro > er
    termo = (-1)^k * (x^(2*k+1))/(factorial(2*k+1));
    sin_aproximado += termo;

```

```

    erro = abs(sin_real - sin_aproximado);

```

```

    k++;
end

```

```

sin_aproximado

```

## q55.m

```

%Questao 55
%Marco Antonio P Leao

```



```

clc;

v=(10:10:80)';
F=[25; 70; 380; 550; 610; 1220; 830; 1450];

X = [ones(length(v),1), v];
A = (X' * X) \ (X' * F);
a0 = A(1);
a1 = A(2);
Fhat = X * A;
SS_res = sum((F - Fhat).^2);
SS_tot = sum((F - mean(F)).^2);
R2 = 1 - SS_res / SS_tot;

erro=Fhat-F;

figure(1);
hold on;
plot(v,F,'r','markersize',9);
plot(v,F,'-r');
v_lin = linspace(min(v), max(v), 200)';
plot(v_lin, a0 + a1 * v_lin, '-k');
hold off;
title('experimento do tunel de vento');
xlabel('velocidade do vento (m/s)');
ylabel('forca do vento (N)');
grid on;

figure(2);
plot(v, erro, 'o-b','MarkerSize', 9);
title('erro');
xlabel('velocidade do vento (m/s)');
ylabel('erro');
grid on;

printf('Funcao encontrada: F(v) = %d + %d * v\n', a0, a1);

```

## q56.m

```

%q56 DeepSeek
% Dados do termistor NTC
B0 = -5.380125;
B1 = 4777.517;
B2 = -1.20014e5;
B3 = -2.1239e6;

% Faixa de temperatura em Celsius

```

```

T_C = linspace(0, 70, 100);
T_K = T_C + 273.15; % Converter para Kelvin

% 1. Calcular R usando a equação de Steinhart-Hart
lnR = B0 + B1./T_K + B2./T_K.^2 + B3./T_K.^3;
R = exp(lnR); % Resistência em Ohm
R_kOhm = R / 1000; % Resistência em kOhm

% 2. Plotar Figura 1 - Relação R=f(T)
figure(1);
plot(T_C, R_kOhm, 'b', 'LineWidth', 2);
xlabel('Temperatura (°C)');
ylabel('Resistência (kohms)');
title('Relação R=f(T) para termistor NTC');
grid on;
hold on;

% 3. Obter polinômio de grau 2 que melhor representa R=f(T)
p2 = polyfit(T_C, R, 2);
R_p2 = polyval(p2, T_C);
R_p2_kOhm = R_p2 / 1000;
plot(T_C, R_p2_kOhm, 'r--', 'LineWidth', 1.5);

% 4. Calcular erro para polinômio de grau 2
e2 = abs(R - R_p2);

% 5. Obter polinômio de grau 3 que melhor representa R=f(T)
p3 = polyfit(T_C, R, 3);
R_p3 = polyval(p3, T_C);
R_p3_kOhm = R_p3 / 1000;
plot(T_C, R_p3_kOhm, 'g-', 'LineWidth', 1.5);
legend('Equação Steinhart-Hart', 'Polinômio grau 2', 'Polinômio grau 3');
hold off;

% 6. Calcular erro para polinômio de grau 3
e3 = abs(R - R_p3);

% 7. Plotar Figura 2 - Erros
figure(2);
subplot(2,1,1);
plot(T_C, e2, 'r');
title('Erro do polinômio de grau 2');
xlabel('Temperatura (°C)');
ylabel('Erro (ohms)');
grid on;

subplot(2,1,2);
plot(T_C, e3, 'g');

```

```

title('Erro do polinômio de grau 3');
xlabel('Temperatura (°C)');
ylabel('Erro (ohms)');
grid on;

% 8. Mostrar valores médios dos erros
mean_e2 = mean(e2);
mean_e3 = mean(e3);
disp(['Erro médio polinômio grau 2: ', num2str(mean_e2), ' ohms']);
disp(['Erro médio polinômio grau 3: ', num2str(mean_e3), ' ohms']);

% 9. Encontrar polinômio para T=f(R) usando regressão
% Primeiro, vamos criar mais pontos para melhor ajuste
T_C_fine = linspace(0, 70, 500);
T_K_fine = T_C_fine + 273.15;
lnR_fine = B0 + B1./T_K_fine + B2./T_K_fine.^2 + B3./T_K_fine.^3;
R_fine = exp(lnR_fine);

% Ajustar polinômio (usando grau 3 para T=f(R))
p_T = polyfit(R_fine, T_C_fine, 3);

% 10. Plotar Figura 3 - Relação T=f(R)
R_plot = linspace(1, 40000, 500); % de 1 a 40k Ohm
T_steinhart = 1./ ( (log(R_plot) - B0 - B2./(log(R_plot).^2) - B3./(log(R_plot).^3) ) / B1 ) -
273.15;

figure(3);
plot(R_plot/1000, T_steinhart, 'b', 'LineWidth', 2);
xlabel('Resistência (kohms)');
ylabel('Temperatura (°C)');
title('Relação T=f(R) para termistor NTC');
grid on;
hold on;

% Plotar polinômio encontrado
T_poly = polyval(p_T, R_plot);
plot(R_plot/1000, T_poly, 'r--', 'LineWidth', 1.5);
legend('Equação Steinhart-Hart', 'Polinômio ajustado');
hold off;

% 11. Calcular valores para T=30°C
T_target = 30;
T_K_target = T_target + 273.15;

% (a) Usando equação original
lnR_a = B0 + B1/T_K_target + B2/T_K_target^2 + B3/T_K_target^3;
R_a = exp(lnR_a);

```

```

% (b) Usando polinômio grau 2
R_b = polyval(p2, T_target);

% (c) Usando polinômio grau 3
R_c = polyval(p3, T_target);

disp('Valores de R para T=30°C:');
disp(['(a) Steinhart-Hart: ', num2str(R_a), ' ohms']);
disp(['(b) Polinômio grau 2: ', num2str(R_b), ' ohms']);
disp(['(c) Polinômio grau 3: ', num2str(R_c), ' ohms']);

% 12. Calcular valores de T para os R encontrados
% (a) Para R_a
T_a = polyval(p_T, R_a);

% (b) Para R_b
T_b = polyval(p_T, R_b);

% (c) Para R_c
T_c = polyval(p_T, R_c);

disp('Valores de T para os R encontrados:');
disp(['Para R(a): ', num2str(T_a), ' °C']);
disp(['Para R(b): ', num2str(T_b), ' °C']);
disp(['Para R(c): ', num2str(T_c), ' °C']);

```

## q57.m

% Questao 57

% Lucas Vasconcelos Resende

```

p = input('digite a precisão: ');
real = pi;
aprox1 = 0;
indices1 = [];
erro1 = [];
k = 0;
termo1 = 1;
while(termo1 >= p) || (termo1 <= -p)
    termo1 = ((-3)^(-k)) / (2*k + 1);
    aprox1 += termo1;
    piaprox1 = sqrt(12)*aprox1;
    erro1(end+1) = abs(pi - piaprox1);
    indices1(end+1) = k+1;
    k += 1;
endwhile
aprox2 = 0;
indices2 = [];

```

```

erro2 = [];
k = 0;
termo2 = 1;
while(termo2 >= p) || (termo2 <= -p)
    termo2 = (2 * sqrt(2) / 9801) * (factorial(4*k) * (1103 + 26390*k) / ((factorial(k))^4 *
396^(4*k)));
    aprox2 += termo2;
    piaprox2 = inv(aprox2);
    erro2(end+1) = abs(pi - piaprox2);
    indices2(end+1) = k+1;
    k += 1;
endwhile
%resultados
printf("pi real: %f\n", real);
printf("Pi aproximado com a PRIMEIRA equação: %f\n", piaprox1);
printf("Pi aproximado com a SEGUNDA equação: %f\n", piaprox2);
%plot dos erros
figure(1)
subplot(2,1,1)
plot(indices1, erro1)
grid on
subplot(2,1,2)
plot(indices2, erro2)
grid on

```

## q58.m

%Questao 58

%Marco Antonio P Leao

```

x=1:10;
y=[0.3808,4.5265,5.0574,8.8480,9.2784,10.1501,11.2439,13.3886,12.1952,14.5999];

```

```

figure(1)
plot(x,y,'ob','markersize',9)
hold on
X=log(x);
a=polyfit(X,y,1);
b=polyval(a,X);
plot(x,b,'-r',9)
xlabel('x');
ylabel('y');
grid
axis('equal');

```

```

erro=y-b;
figure(2)
plot(x,erro,'-r', 9)

```

```

xlabel('x');
ylabel('y');
grid

fprintf("f(x) = %d * ln(x) + %d\n", a);

```

## q59.m

```

%q59 DeepSeek
% Constantes da equação de Steinhart-Hart para termistor 10kohms (da tabela 10k-2)
A0 = 1.0295e-3;
A1 = 2.391e-4;
A3 = 1.568e-7;

% Faixa de resistência em Ohm
R_Ohm = linspace(1000, 30000, 500); % de 1kohms a 30kohms
R_kOhm = R_Ohm / 1000; % em kohms

% 1. Calcular T usando a equação de Steinhart-Hart
lnR = log(R_Ohm);
T_K = 1 ./ (A0 + A1.*lnR + A3.*(lnR).^3);
T_C = T_K - 273.15; % Converter para Celsius

% 2. Plotar Figura 1 - Relação T=f(R)
figure(1);
plot(R_kOhm, T_C, 'b', 'LineWidth', 2);
xlabel('Resistência (kohms)');
ylabel('Temperatura (°C)');
title('Relação T=f(R) para termistor NTC 10kohms');
grid on;
hold on;

% 3. Obter polinômio de grau 2 que melhor representa T=f(R)
p2 = polyfit(R_Ohm, T_C, 2);
T_p2 = polyval(p2, R_Ohm);
plot(R_kOhm, T_p2, 'r--', 'LineWidth', 1.5);

% 4. Calcular erro para polinômio de grau 2
e2 = abs(T_C - T_p2);

% 5. Obter polinômio de grau 3 que melhor representa T=f(R)
p3 = polyfit(R_Ohm, T_C, 3);
T_p3 = polyval(p3, R_Ohm);
plot(R_kOhm, T_p3, 'g-.', 'LineWidth', 1.5);
legend('Equação Steinhart-Hart', 'Polinômio grau 2', 'Polinômio grau 3');
hold off;

% 6. Calcular erro para polinômio de grau 3

```

```

e3 = abs(T_C - T_p3);

% 7. Plotar Figura 2 - Erros
figure(2);
subplot(2,1,1);
plot(R_kOhm, e2, 'r');
title('Erro do polinômio de grau 2');
xlabel('Resistência (kohms)');
ylabel('Erro (°C)');
grid on;

subplot(2,1,2);
plot(R_kOhm, e3, 'g');
title('Erro do polinômio de grau 3');
xlabel('Resistência (kohms)');
ylabel('Erro (°C)');
grid on;

% 8. Mostrar valores médios dos erros
mean_e2 = mean(e2);
mean_e3 = mean(e3);
disp(['Erro médio polinômio grau 2: ', num2str(mean_e2), ' °C']);
disp(['Erro médio polinômio grau 3: ', num2str(mean_e3), ' °C']);

% 9. Encontrar polinômio para R=f(T) usando regressão
% Criar pontos para ajuste
T_C_fine = linspace(0.1, 100, 500); % de 0.1°C a 100°C
T_K_fine = T_C_fine + 273.15;
R_fine = zeros(size(T_C_fine));

% Resolver a equação de Steinhart-Hart para R (numericamente)
for i = 1:length(T_C_fine)
    f = @(R) 1./T_K_fine(i) - (A0 + A1*log(R) + A3*(log(R)).^3);
    R_fine(i) = fzero(f, 10000); % Valor inicial 10kohms
end

% Ajustar polinômio (usando grau 3 para R=f(T))
p_R = polyfit(T_C_fine, R_fine, 3);

% 10. Plotar Figura 3 - Relação R=f(T)
T_plot = linspace(0.1, 100, 500);
R_steinhart = zeros(size(T_plot));

for i = 1:length(T_plot)
    T_K_plot = T_plot(i) + 273.15;
    f = @(R) 1./T_K_plot - (A0 + A1*log(R) + A3*(log(R)).^3);
    R_steinhart(i) = fzero(f, 10000);
end

```

```

figure(3);
plot(T_plot, R_steinhart/1000, 'b', 'LineWidth', 2);
xlabel('Temperatura (°C)');
ylabel('Resistência (kohms)');
title('Relação R=f(T) para termistor NTC 10kohms');
grid on;
hold on;

% Plotar polinômio encontrado
R_poly = polyval(p_R, T_plot);
plot(T_plot, R_poly/1000, 'r--', 'LineWidth', 1.5);
legend('Equação Steinhart-Hart', 'Polinômio ajustado');
hold off;

% 11. Calcular valores para R=15kohms
R_target = 15000; % 15kohms

% (a) Usando equação original
lnR_a = log(R_target);
T_K_a = 1 / (A0 + A1*lnR_a + A3*(lnR_a)^3);
T_a = T_K_a - 273.15;

% (b) Usando polinômio grau 2
T_b = polyval(p2, R_target);

% (c) Usando polinômio grau 3
T_c = polyval(p3, R_target);

disp('Valores de T para R=15kohms:');
disp(['(a) Steinhart-Hart: ', num2str(T_a), ' °C']);
disp(['(b) Polinômio grau 2: ', num2str(T_b), ' °C']);
disp(['(c) Polinômio grau 3: ', num2str(T_c), ' °C']);

% 12. Calcular valores de R para os T encontrados
% (a) Para T_a
R_a = polyval(p_R, T_a);

% (b) Para T_b
R_b = polyval(p_R, T_b);

% (c) Para T_c
R_c = polyval(p_R, T_c);

disp('Valores de R para os T encontrados:');
disp(['Para T(a): ', num2str(R_a/1000), ' kohms']);
disp(['Para T(b): ', num2str(R_b/1000), ' kohms']);
disp(['Para T(c): ', num2str(R_c/1000), ' kohms']);

```



**q60.m**

questao repetida da 57