###### Problem 1:

>> x=5; y=23;

>> z=x^3

**Q1. z = 125**

>> z=(2\*y)^2

**Q2. z = 2116**

>> z=5\*x^3 + 4\*y^2

**Q3. z = 2741**

>> z=16\*10^6 \*25\*10^(-3)

**Q4. z = 400000**

z=(sqrt(y^2 +10))\*(sqrt(x^2 -9))

**Q5. z = 92.8655**

###### Problem 2:

>> x=8;

>> x=x^3;

>> x=x+25;

>> x=sqrt(x);

>> x=x+16;

>> z5=x^(1/3)

**Q6. z5 = 3.3962**

###### Problem 3:

% Withdrawal amount

A = 20000;

% Input

r = input('Annual interest rate as a decimal (r): ');

N = input('Number of years (N): ');

% Calculate the initial deposit amount (P)

P = A \* ((1 + r)^N - 1) / (r \* (1 + r)^N);

% Display the result

fprintf('The initial amount (P) is $%.2f\n', P);

**Q7.** The initial amount to be deposited (P) is $**154434.70**

**Q8.** The initial amount to be deposited (P) is $**249244.21**

**Q9.** The initial amount to be deposited (P) is $**307449.02**

**Q10.** The initial amount to be deposited (P) is $**188538.29**

###### Problem 4:

% Tank height

H1=10; H2=18; H3=23;

% Tank radius

R1=5; R2=8; R3=12;

h = input('Height of the water (h): ');

% Calculate volume based on the height

if h >= 0 && h <= H1

V = pi() \* R1^2 \* h;

elseif h > H1 && h <= H2

V = pi() \* R2^2 \* (h - H1) + pi() \* R1^2 \* H1;

elseif h > H2 && h <= H3

V = pi() \* R3^2 \* (h - H2) + pi() \* R2^2 \* (H2 - H1) + pi() \* R1^2 \* H1;

else

disp('Invalid height input. Height must be between 0 and 23.');

return;

end

% Display the calculated volume

fprintf('Volume of water in the tank: %.2f cubic ft\n', V);

**Q11.** Volume of water in the tank: **628.32** cubic ft

**Q12.** Volume of water in the tank: **1790.71** cubic ft

**Q13.** Volume of water in the tank: **4203.45** cubic ft

###### Problem 5:

>> A=[15,8;7,3];

>> B=[14,17;23,25];

>> C=[18,-9;14,-10];

>> Z=A\*B\*C;

>> Z(2,1)

**Q14.** ans = **5722**

>> det(B)

**Q15.** ans =  **-41**

>> Z=(C)^(-1);

>> Z(1,2)

**Q16.** ans =  **-0.1667**

>> Z=(B\*C)^(-1)

>> Z(2,2)

**Q17.** ans = **0.2213**

###### Problem 6:

>> A=[2 6 3 8 9 1; 4 2 7 4 9 0]

>> B=[5 7; 3 7; 0 5; 1 4; 9 3; 3 8]

>> Z=A\*B;

>> Z(1,2)

**Q18.** ans = **138**

>> Z=B\*A;

>> Z(3,4)

**Q19.** ans = **20**

###### Problem 7:

a = 10; r = 0.8;

% Input

n = input('Number of terms (n): ');

% Initialize the sum

Sn = 0;

% Calculate the sum using a for loop

for i = 0:(n-1)

Sn = Sn + a \* r^i;

end

% Calculate the sum for an infinite number of terms

S\_infinite = a / (1 - r);

% Display the results

fprintf('Sum of the first %d terms: %.4f\n', n, Sn);

fprintf('Sum for an infinite number of terms: %.4f\n', S\_infinite);

**Q20.** Sum of the first 4 terms: **29.52**

**Q21.** Sum of the first 10 terms: **44.6313**

**Q22.** Sum of the first 100 terms: **50**

**Q23.** Sum for an infinite number of terms: **50**

###### Problem 8:

y0 = 10; % Initial height in m

v0 = 50; % Initial velocity in m/s

g = 9.81; % Gravity in m/s^2

% Time values from 0 to 10 seconds

t = 0:0.1:10; % Time values at 0.1-second intervals

% Calculate the position as a function of time (height)

y = y0 + v0 \* t - 0.5 \* g \* t.^2;

% Plot the curve

figure;

plot(t, y, 'b', 'LineWidth', 2);

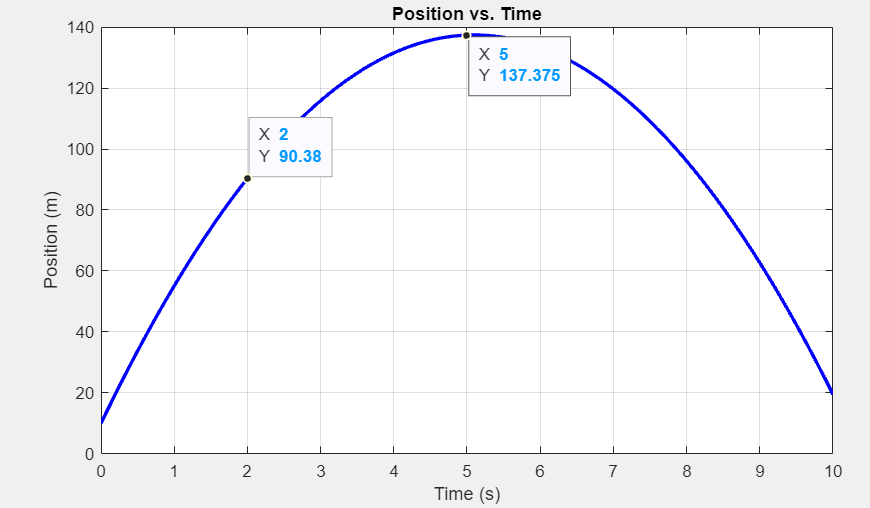
grid on;

% Labeling and title

xlabel('Time (s)');

ylabel('Position (m)');

title('Position vs. Time');



**Q24. 90.38**

**Q25. 137.375**

###### Problem 9:

% Define the range of x values

x = logspace(log10(0.001), log10(1000), 1000);

% Formula

y = x.^(1/6);

% Graph log-log plot

figure;

loglog(x, y, 'b');

xlabel('x (log scale)');

ylabel('y (log scale)');

title('Log-Log Plot of y = x^{1/6}');

% Given

x1 = 0.05; x2 = 500;

% Calculate y for the x values

y1 = x1^(1/6);

y2 = x2^(1/6);

% Calculate the Ln values

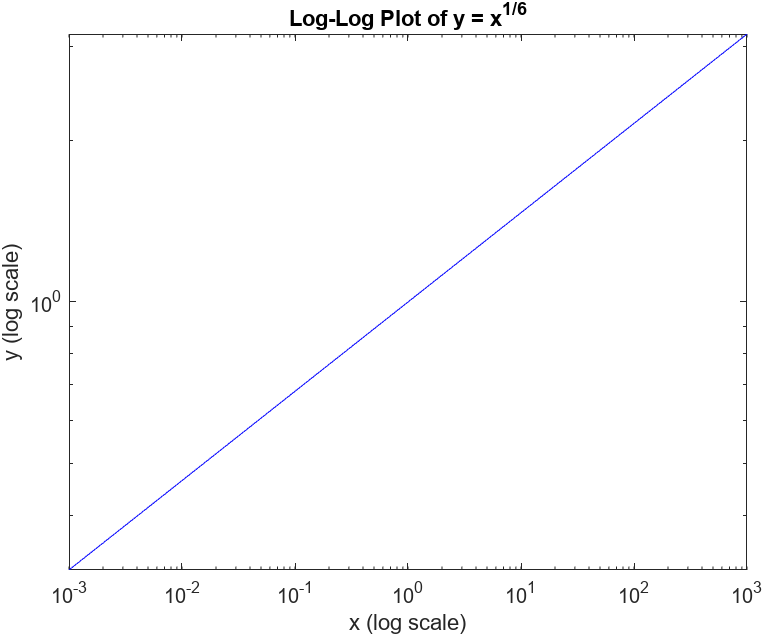
log\_y1 = log(y1);

log\_y2 = log(y2);

% Display

fprintf('For x = 0.05, log(y) = %.4f\n', log\_y1);

fprintf('For x = 500, log(y) = %.4f\n', log\_y2);



**Q26.** For x = 0.05, log(y) = **-0.4993**

**Q27.** For x = 500, log(y) = **1.0358**

###### Problem 10:

% Define the Coefficients

coefficients = [1, 3.2, -5.4, 12.3, 5.4, -3.2, 1];

% Find Roots

roots\_list = roots(coefficients);

% Find Real Roots

real\_roots = sum(isreal(roots\_list));

% Find pairs of complex-conjugate roots

complex\_conjugate\_roots = sum(~isreal(roots\_list)) / 2;

% Find the biggest negative real root

negative\_real\_roots = roots\_list(real\_roots & roots\_list < 0);

biggest\_negative\_real\_root = max(negative\_real\_roots);

% Find the real part of the complex conjugate root

complex\_roots = roots\_list(~isreal(roots\_list));

imaginary\_parts = imag(complex\_roots);

real\_part\_complex\_root = real(complex\_roots(imaginary\_parts == max(imaginary\_parts)));

% Display the results

fprintf('Number of real roots: %d\n', real\_roots);

fprintf('Number of pairs of complex-conjugate roots: %d\n', complex\_conjugate\_roots);

fprintf('Value of the biggest negative real root: %.4f\n', biggest\_negative\_real\_root);

fprintf('Real part of the complex conjugate root with the larger imaginary part: %.4f\n', real\_part\_complex\_root);

**Q28. 2**

**Q29. 4**

**Q30. -4.8025**

**Q31. 0.94082**

###### Problem 11:

% Define the range of x values

x = linspace(-3, 3, 200);

% Convert the angle from degrees to radians

a\_deg = 40;

a\_rad = deg2rad(a\_deg);

% Calculate y values using the equation

y = 24 \* sin(3.5 \* x + a\_rad);

% Find the value of y for x = 1

x\_1 = 1;

y\_1 = 24 \* sin(3.5 \* x\_1 + a\_rad);

% Find the value of y for x = 2

x\_2 = 2;

y\_2 = 24 \* sin(3.5 \* x\_2 + a\_rad);

% Plot the curve

figure;

plot(x, y)

title('Plot of y = 24\*sin(3.5x + 40^\circ)')

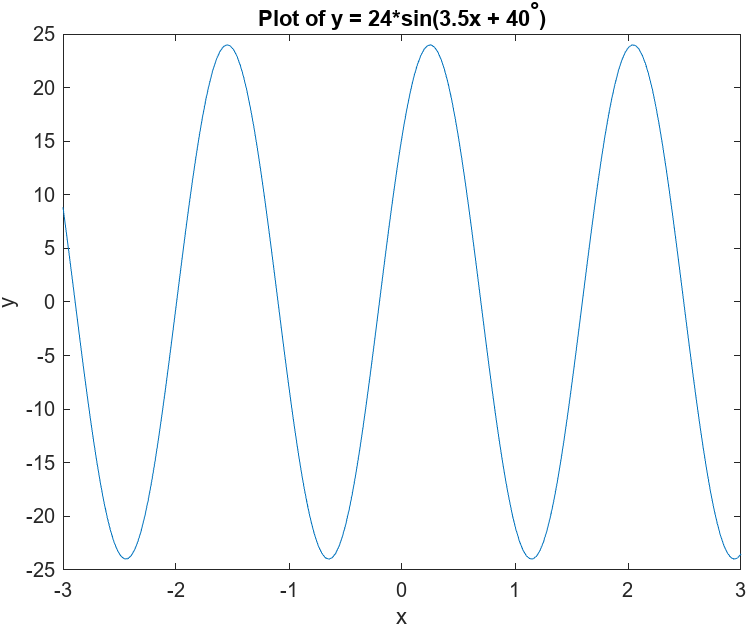
xlabel('x')

ylabel('y')

% Display the values

fprintf('Value of y for x = 2: %.2f\n', y\_2);

fprintf('Value of y for x = 1: %.2f\n', y\_1);



**Q32. 23.71**

**Q33. -20.90**

###### Problem 12:

% Define the range of t values

t = linspace(0, 10, 1000);

% Formulas

x = 10 \* cos(3 \* t) + 2 \* sin(20 \* t);

y = 10 \* t;

z = 10 \* sin(3 \* t) - 2 \* cos(20 \* t);

%Plot

figure;

plot(t, x, 'b', t, y, 'g', t, z, 'r');

xlabel('Time (t)');

ylabel('Value');

legend('x(t)', 'z(t)');

title('Plot of x(t) and z(t) vs. Time');

grid on;

% Restrict the vertical range to -20 to 20

ylim([-20, 20]);

t\_x = 1;

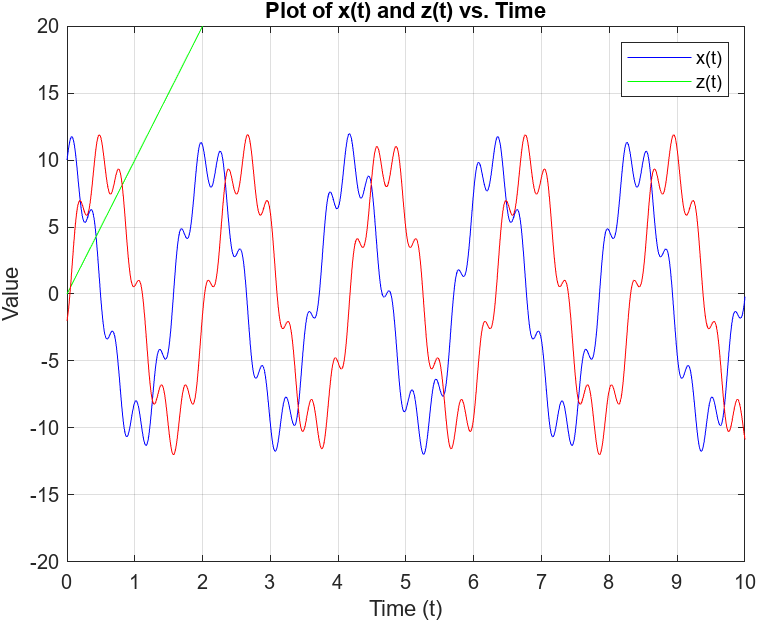
t\_z = 2;

x\_value = interp1(t, x, t\_x);

z\_value = interp1(t, z, t\_z);

fprintf('Value of x for t=1: %.2f\n', x\_value);

fprintf('Value of z for t=2: %.2f\n', z\_value);



**Q34. -8.08**

**Q35. -1.46**