2.1.1 Greatest of 3 Numbers

```
a = input('Enter first number: ');
b = input('Enter second number: ');
c = input('Enter third number: ');

max_num = max([a, b, c]);
disp(['The maximum number is: ', num2str(max_num)]);
```

2.1.2 Greatest of 3 Numbers using Conditionals (if ma'am asks)

```
a = input('Enter first number: ');
b = input('Enter second number: ');
c = input('Enter third number: ');

if a >= b && a >= c
   max_num = a;
elseif b >= a && b >= c
   max_num = b;
else
   max_num = c;
end

disp(['The maximum number is: ', num2str(max_num)]);
```

2.2 Program for Calculator Using Switch Case

```
num1 = input('Enter first number: ');
num2 = input('Enter second number: ');
op = input('Enter operator (+, -, *, /): ', 's');

switch op
```

```
    case '+'
        result = num1 + num2;
    case '-'
        result = num1 - num2;
    case '*'
        result = num1 * num2;
    case '/'
        result = num1 / num2;
    otherwise
        disp('Invalid operator');
        return;
end

disp(['Result: ', num2str(result)]);
```

3.1 Armstrong Number

```
num = input('Enter a number: ');

digits = num2str(num) - '0';
num_digits = length(digits);

armstrong = sum(digits .^ num_digits) == num;

if armstrong
    disp('It is an Armstrong number');
else
    disp('It is not an Armstrong number');
end
```

3.2 Prime Number

```
num = input('Enter a number: ');

prime = true;

if num <= 1
   prime = false;
else
   for i = 2:sqrt(num)
      if rem(num, i) == 0
         prime = false;
         break;
      end
   end
end

if prime
   disp('It is a prime number');
else
   disp('It is not a prime number');
end
```

4. Bubble Sort

```
arr = input('Enter numbers to sort (in square brackets): ');

n = length(arr);

for i = 1:n-1
   for j = 1:n-i
      if arr(j) > arr(j+1)
```

```
            temp = arr(j);
            arr(j) = arr(j+1);
            arr(j+1) = temp;
        end
    end
end

disp('Sorted array:');
disp(arr);
```

5. Program to perform + * / inverse and transpose of matrices

```
mat1 = input('Enter the first matrix: ');
mat2 = input('Enter the second matrix: ');

addition = mat1 + mat2;
disp('Addition:');
disp(addition);

subtraction = mat1 - mat2;
disp('Subtraction:');
disp(subtraction);

multiplication = mat1 * mat2;
disp('Multiplication:');
disp(multiplication);

division = mat1 ./ mat2;
disp('Division:');
```

```
disp(division);

inv_mat1 = inv(mat1);
disp('Inverse of the first matrix:');
disp(inv_mat1);

transpose_mat1 = transpose(mat1);
disp('Transpose of the first matrix:');
disp(transpose_mat1);
```

6. Program to solve matrices using linsolve and solve

```
% Solving a system of linear equations using linsolve
A = input('Enter the coefficients matrix A: ');
B = input('Enter the constants vector B: ');

x = linsolve(A, B);
disp('Solution using linsolve:');
disp(x);

% Solving equations symbolically using solve
syms x1 x2 x3; % Define symbolic variables if the system has 3 unknowns, adjust as needed
equations = A * [x1; x2; x3] == B;
solution = solve(equations, [x1; x2; x3]);
disp('Solution using solve:');
disp(solution);
```

7. Plotting graph

```
x = -5:0.1:5;
y = x.^2;
```

```
plot(x, y, 'LineWidth', 2, 'Marker', 'o', 'MarkerSize', 5, 'MarkerFaceColor', 'red', 'Color', 'blue', 'LineStyle', '--');
title('Plot of y = x^2');
xlabel('x');
ylabel('y');
grid on;
```

8. Quick Sort

```
function sortedArr = quicksort(arr)
    if numel(arr) <= 1
        sortedArr = arr;
        return;
    end

    pivot = arr(ceil(end/2));
    less = arr(arr < pivot);
    equal = arr(arr == pivot);
    greater = arr(arr > pivot);

    sortedArr = [quicksort(less), equal, quicksort(greater)];
end

% Example usage:
arr = input('Enter numbers to sort (in square brackets): ');
sorted = quicksort(arr);
disp('Sorted array:');
disp(sorted);
```