```
clear  % This clears out the internal memory of Matlab - always a good idea
       % at the start of a new computation.

% Code for part (a) of Lagrange interpolation program

N=input('Enter order of polynomial being generated ');

for i = 1:N+1
   x(i) = input('Enter next x-value ');
   y(i) = input('Enter next y-value ');
end

xstar = input('Enter value of x at which you wish to interpolate ');

disp(' ')
disp('Part (a) completed')
disp('Press any key to continue')
disp(' ')
pause % pause calculation

% Part (a) finished.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Code for part (b) of Lagrange interpolation program

% Note that "term(1)" is built up term-by-term in a "loop"
% which is repeated N times:

term(1) = 1.0;  % Initialize "term(1)"
for i = 1:N+1   % Begin computational loop
   if i ~= 1    % "~=" mean "not equal to"
      term(1) = term(1)*(xstar - x(i))/(x(1)-x(i));  % Build up L0(x)
   end          % End conditional statement
end             % End computational loop

term(1) = term(1)*y(1)      % Factor in y-value
disp(' ')                   % Print a blank line on screen
disp('Part (b) completed')  % Print text
disp('Press any key to continue')

% Part (b) finished.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Code for part (c) of Lagrange interpolation program
% This is similar to the part (b) *except* that instead of generating
% the first term of the Lagrange polynomial, we generate the j-th term.


j = input('Specify which Lagrange polynomial term is being calculated ')

term(j) = 1.0;  % Initialize "term(j)"
for i = 1:N+1   % Begin computational loop
   if i ~= j
      term(j) = term(j)*(xstar - x(i))/(x(j)-x(i));   % Build up Lj(x)
   end
end

term(j) = term(j)*y(j)   % Factor in y-value
disp(' ')
disp('Part (c) completed')
disp('Press any key to continue')

% Part (c) finished.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Code for part (d) of Lagrangian interpolation program
% This is based directly on part (c) code which is used
% to build up whole polynomial.
% The complete Lagrange interpolation program will consist of
% Part (a) code + code below!

Pn = 0.0
for j = 1:N+1
  term(j) = 1.0;  % Initialize "term(j)"
  for i = 1:N+1   % Begin computational loop
     if i ~= j
        term(j) = term(j)*(xstar - x(i))/(x(j)-x(i));   % Build up Lj(x)
     end
  end
  term(j) = term(j)*y(j);   % Factor in y-value
  Pn = Pn + term(j);        % Build up polynomial
end
disp(' ')
disp('Interpolated y-value at xstar: '); Pn
```