Lab 7 Resultant Force Calculator

Part A: Polar to Rectangular Function

Create a MATLAB function that will convert a force in polar form to rectangular form. The function should have two input arguments, the magnitude and phase (in degrees) of the force. The function should have two output arguments, the x-component and y-component of the force.

Test your function by completing the table below. Check results with your T.A.

Force	x-component	y-component
$F_1 = 15\angle 30^o\ N$	12.990381	7.500000
$F_2 = 20\angle 150^o\ N$	-17.320508	10.000000
$F_3 = 75\angle -140^o\ N$	-57.453333	-48.209071

PASTE THE COMMAND YOU USED IN THE COMMAND WINDOW TO RUN THE FUNCTION FOR ONE OF THE FORCES IN THE TABLE:

```
clear;clc;
n=input('How many forces do you want to entre?');
ResultantForce( n );
```

PASTE MATLAB CODE FOR POLAR TO RECTANGULAR FUNCTION HERE:

```
function [ magnitude , angle ] = ResultantForce( n )
%  The purpose of this function is to calculate the
resultant force and plot the individual forces and the
resultant force
%  input argument: n is the number if forces the user
wishes to enter
%  output arguments: magnitude is the the magnitude of the
resultant force
%                    angele is the angle of the resultant
force
%  usage: [ magnitude , angle ] = ResultantForce( f )
   X = zeros(1,n);
   Y = zeros(1,n);
   fx=0;
   fy=0;
   for i=1:n
       form = menu('What is the form of the force you wish
to enter?','Rectangular form','Polar form');
       if form == 1
          choose = 2;
```

```
            while choose == 2
                X(i)=input('The x component of the force:');
                Y(i)=input('The y component of the force:');
                fprintf('The x component is %f and y
component is %f',X(i),Y(i))
                choose = menu('Are these values
correct?','yes','no');
            end
        elseif form == 2
             choose = 2;
            while choose == 2
                m = input('The x component of the force:');
                a = input('The y component of the force:');
                X(i)=m*cosd(a);
                Y(i)=m*sind(a);
                fprintf('The values of magnitude is %f and
angle is %f',X(i),Y(i))
                choose = menu('Are these values
correct?','yes','no');
            end
        end
        fx = fx+X(i);
        fy = fy+Y(i);
        plot([0,X(i)],[0,Y(i)],'k-')
        hold on
    end
    title('Resultant Force')
    plot([0,fx],[0,fy],'--b')
    grid on
    magnitude = sqrt(fx.^2+fy.^2);
    angle = atan2d(fy,fx);
    fprintf('The magnitude is');
    disp(magnitude);
    fprintf('The angle is degree');
    disp(angle)
    fprintf('X-component is');
    disp(fx);
    fprintf('Y-component is');
    disp(fy);
end
```

Part B: Resultant Force Calculator

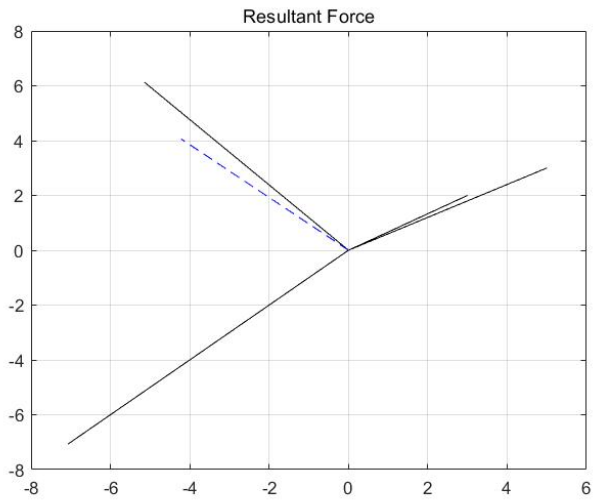
Create a MATLAB function that will compute the resultant force based on individual forces entered by the user.

- The function should have one input argument, the number of forces the user wishes to enter.
- The function should have four output arguments: the magnitude and phase of the resultant force and the x and y components of the resultant force.
- The function should be set up to accommodate any combination of forms for the user's forces; in other words, if the user has 5 forces to enter, three of them could be in polar form and two could be in rectangular form.
- When the user enters a force, the values the user entered should be displayed in the command window. The function should then include code to ask the user if the values were entered correctly, and if not, allow the user to re-enter the force values. Use a while loop for this.
- If a force is entered in polar form, the function should call the function written in Part A to convert it to rectangular form.
- The function should also produce a single plot showing the individual forces in one color (your choice) and the resultant force in another color (again, your choice).
  Hint: `plot([x1,x2],[y1,y2],'k-')` will draw a black line from the point (x_1,y_1) to the point (x_2,y_2).

Test your function for the set of forces shown in the table below. Check results with the T.A. Then enter the output produced in the command window in the table below and paste the plot where indicated.

Function	Magnitude	Angle	X-component	Y-Component
F1			5	3
F2	8	130°		
F3	10	225°		
F4			3	2
Resultant Force	5.8493	136.0811	-4.2134	4.0573

PLOT:



PASTE MATLAB CODE FOR RESULTANT FORCE CALCULATOR FUNCTION HERE:

```
function [ magnitude , angle ] = ResultantForce( n )
%  The purpose of this function is to calculate the
resultant force and plot the individual forces and the
resultant force
%  input argument: n is the number if forces the user
wishes to enter
%  output arguments: magnitude is the the magnitude of the
resultant force
%                    angele is the angle of the resultant
force
%  usage: [ magnitude , angle ] = ResultantForce( f )
   X = zeros(1,n);
   Y = zeros(1,n);
   fx=0;
   fy=0;
   for i=1:n
       form = menu('What is the form of the force you wish
to enter?','Rectangular form','Polar form');
       if form == 1
          choose = 2;
          while choose == 2
              X(i)=input('The x component of the force:');
              Y(i)=input('The y component of the force:');
              fprintf('The x component is %f and y
component is %f',X(i),Y(i))
              choose = menu('Are these values
```

```
correct?','yes','no');
            end
        elseif form == 2
            choose = 2;
            while choose == 2
                m = input('The x component of the force:');
                a = input('The y component of the force:');
                X(i)=m*cosd(a);
                Y(i)=m*sind(a);
                fprintf('The values of magnitude is %f and
angle is %f',X(i),Y(i))
                choose = menu('Are these values
correct?','yes','no');
            end
        end
        fx = fx+X(i);
        fy = fy+Y(i);
        plot([0,X(i)],[0,Y(i)],'k-')
        hold on
    end
    title('Resultant Force')
    plot([0,fx],[0,fy],'--b')
    grid on
    magnitude = sqrt(fx.^2+fy.^2);
    angle = atan2d(fy,fx);
    fprintf('The magnitude is');
    disp(magnitude);
    fprintf('The angle is degree');
    disp(angle)
    fprintf('X-component is');
    disp(fx);
    fprintf('Y-component is');
    disp(fy);
end
```