Chapter 9 Exercises

1. Add the following complex numbers:
   (a) $2 + 3i$ and $3 - 5i$  
   (b) $-7 - 9i$ and $12i$  
   (c) $5$ and $6 - 7i$

2. Multiply the complex numbers of exercise 1.

3. When necessary, use diagrams like those on page 147 to convert the following points from rectangular to polar form:
   (a) $(1, -1)$  
   (b) $(-3,0)$  
   (c) $(-\sqrt{3},1)$  
   (d) $(-\pi,-\pi)$  
   (e) $(0,1)$

4. Similarly convert the following points from polar to rectangular form:
   (a) $(5\sqrt{2}, 225^\circ)$  
   (b) $(6,270^\circ)$  
   (c) $(7\sqrt{3},210^\circ)$  
   (d) $(4,330^\circ)$  
   (e) $(2,135^\circ)$

5. Convert the following to polar form as $r \text{ cis } A$ with $A$ in degrees:
   (a) $1 - i$  
   (b) $-3$  
   (c) $-\sqrt{3} + i$  
   (d) $-\pi - \pi$  
   (e) $i$

6. Convert your answers in 5 to $r \text{ cis } A$ with $A$ in radian measure.

7. One of the interesting powers that polar representation gives you is the ability to draw new kinds of graphs. To do so, you will need to prepare your calculator. In \textbf{MODE} set angle measure to \textbf{Radian} and function type to \textbf{Pol}. In \textbf{WINDOW} set $\theta_{\text{min}}$ to 0, $\theta_{\text{max}}$ to $4\pi$, $X_{\text{min}}$ to -8 and $X_{\text{max}}$ to 8. Here are a few of the many graphs you can now draw:
   (a) $r = \theta/4$  
   Set this in the \textit{Y=} window. Note that $\theta$ is on the \textit{X,T,θ,n} key.
   (b) $r = 3(1-\cos(\theta))$  
   This curve is called a cardioid. The etymology of that Latin word relates to the heart as in cardiology. Can you see why that name is assigned to this curve?
   (c) $r = 5\cos(2\theta)$  
   (d) $r = 5\cos(3\theta)$  
   (e) $r = 5\cos(2\theta)$  
   These are called roses.

8. In Chapter 10 you will be working with the individual dots or pixels that make up your calculator screen. These pixels are in rows and columns. What then must be the functions built into your calculator that will allow you to type in equations of the form $r = f(\theta)$ and have it graphed?

9. Recall that we have from page 153 the multiplier that rotates a ray $A$ degrees. It is $1 + i \tan A$. Thus, if we have a ray through the point $x + iy$ and we wish to rotate it $A$ degrees, we form the product $(1 + i \tan A)(x + iy) = (x - y \tan A) + i(y + x \tan A)$. Use this last expression to find a point in complex form through which the ray resulting from the given rotation will pass:
   (a) $1 + i$ rotated $45^\circ$  
   (b) $1 - i$ rotated $90^\circ$  
   (c) $1 + i\sqrt{3}$ rotated $60^\circ$

10. Write a program that will enter the numbers 10, 9, 8, 7,...,3,2,1 in a matrix and then print them out one at a time. \textit{Information about processing matrices in programs is on pages 186 and 187 of Appendix A.}

11. This exercise will show how the program \texttt{CORDSIM} converges on an angle as it calculates cosine.
(a) Enter the program **CORDSIM2** in your calculator.\(^1\)

```
CORDSIM2
: Prompt D
: 1→T
: 1→X
: 0→Y
: For (N,1,13)
: [A](1,N)→A
: D→A→D
: Disp A,D
: Pause
: While D≥0
: X→K
: X→T*Y→X
: Y+T*K→Y
: D→A→D
: Disp A,D
: Pause
: End (While)
: D+A→D
: Disp A,D
: Pause
: 0.1*T→T
: End (For)
: Disp X/(X^2+Y^2)
```

(b) Enter, and if you have not yet done so, run the program **CORDMAT** to enter the appropriate tangent values in the matrix \([A]\).

(c) Now run **CORDSIM2** with the value \(D = 63.5\). Your program will display \(A\), the angle being subtracted (or added back) at each step, and \(D\), the angle remaining. What you will see will be like the table on page 152 of Inside Your Calculator, but it will also include additional steps: the subtraction that makes \(D\) negative and the addition of that value of \(A\) back to give you the previous \(D\). You will see that it takes your calculator many of these steps to converge on the angle for which cosine is calculated. Count the number of these steps (one for each time your calculator stops to report). The number of them should impress you with how much work this program and the program **CORDSIM** are doing, the latter in about 2 seconds.

(d) Decide what would happen if you ran **CORDSIM2** for \(D = 135\). Then run the program to see if what you guessed is correct.

\(^1\) You can shorten the process of entering this modification of **CORDSIM** by copying that program into **CORDSIM2** and then adding the new program lines. To do this, once you have named your new program **CORDSIM2**, press ENTER, then 2nd RCL, then PRGM and choose EXEC. Scroll down to **CORDSIM** and press ENTER. This will copy the program steps of **CORDSIM** into **CORDSIM2** ready for editing.
(e) You may be surprised to find in (d), that, even though $D = 0$ after a few steps, the program continues to process subsequent values. You can fix this by the addition of four more lines of code. Within the While loop after the line $Y + T*K \rightarrow Y$ enter:

$$\begin{array}{l}
: \text{If} \ D=0 \\
: \text{Goto} \ 1
\end{array}$$

and after the End of the For loop enter:

$$\begin{array}{l}
: \text{Lbl} \ 1
\end{array}$$

Do this and test your program CORDSIM2 for $D = 135$, $D = 90$ and $D = 50.710593137$.

(f) Notice that when you check the last value, $D = 50.710593137$, no zero appears. Trace the program with that input value to see why zero fails to occur this time.