Figure 3-54
Output generated from the pieChart procedure.

long windowID = openGraphics (*argv, 400, 100);
Fig f;
  /* Center positions for each figure */
  wcPt2 center[] = { 50, 50, 100, 50, 175, 50, 250, 50, 300, 50 };

  /* Parameters to define each figure. First four need one parameter.
     Fifth figure (limacon) needs two. */
  int p[5][2] = { 5, -1, 20, -1, 30, -1, 30, -1, 40, 10 };

  setBackground (WHITE);
  setColor (BLACK);
  for (f=spiral; f<=limacon; f++)
    drawCurlyFig (f, center[f], p[f]);
  sleep (10);
  closeGraphics (windowID);
}

REFERENCES

Information on Bresenham's algorithms can be found in Bresenham (1965, 1977). For midpoint methods, see Kappel (1985). Parallel methods for generating lines and circles are discussed in Pang (1990) and in Wright (1990).

Additional programming examples and information on PHIGS primitives can be found in Howard, et al. (1991), Hopgood and Duce (1991), Gaskins (1992), and Blake (1993). For information on GKS output primitive functions, see Hopgood et al. (1983) and Enderle, Kansy, and Pfaff (1984).

EXERCISES

3-1. Implement the polyline function using the DDA algorithm, given any number (n) of input points. A single point is to be plotted when n = 1.

3-2. Extend Bresenham's line algorithm to generate lines with any slope, taking symmetry between quadrants into account. Implement the polyline function using this algorithm as a routine that displays the set of straight lines connecting the n input points. For n = 1, the routine displays a single point.
3.3. Devise a consistent scheme for implementing the polyline function, for any set of input line endpoints, using a modified Bresenham line algorithm so that geometric magnitudes are maintained (Section 3-10).

3.4. Use the midpoint method to derive decision parameters for generating points along a straight-line path with slope in the range $0 < m < 1$. Show that the midpoint decision parameters are the same as those in the Bresenham line algorithm.

3.5. Use the midpoint method to derive decision parameters that can be used to generate straight line segments with any slope.

3.6. Set up a parallel version of Bresenham’s line algorithm for slopes in the range $0 < m < 1$.

3.7. Set up a parallel version of Bresenham’s algorithm for straight lines of any slope.

3.8. Suppose you have a system with an 8-inch by 10-inch video monitor that can display 100 pixels per inch. If memory is organized in one-byte words, the starting framebuffer address is 0, and each pixel is assigned one byte of storage, what is the framebuffer address of the pixel with screen coordinates $(x, y)$?

3.9. Suppose you have a system with an 8-inch by 10-inch video monitor that can display 100 pixels per inch. If memory is organized in one-byte words, the starting framebuffer address is 0, and each pixel is assigned 6 bits of storage, what is the framebuffer address (or addresses) of the pixel with screen coordinates $(x, y)$?

3.10. Implement the `setPixel` routine in Bresenham's line algorithm using iterative techniques for calculating framebuffer addresses (Section 3-3).

3.11. Revise the midpoint circle algorithm to display so that geometric magnitudes are maintained (Section 3-10).

3.12. Set up a procedure for a parallel implementation of the midpoint circle algorithm.

3.13. Derive decision parameters for the midpoint ellipse algorithm assuming the start position is $(r_x, 0)$ and points are to be generated along the curve path in counterclockwise order.

3.14. Set up a procedure for a parallel implementation of the midpoint ellipse algorithm.

3.15. Devise an efficient algorithm that takes advantage of symmetry properties to display a sine function.

3.16. Devise an efficient algorithm, taking function symmetry into account, to display a plot of damped harmonic motion:

$$y = Ae^{-x} \sin(\omega x + \theta)$$

where $\omega$ is the angular frequency and $\theta$ is the phase of the sine function. Plot $y$ as a function of $x$ for several cycles of the sine function or until the maximum amplitude is reduced to $A/10$.

3.17. Using the midpoint method, and taking symmetry into account, develop an efficient algorithm for scan conversion of the following curve over the interval $-10 \leq x \leq 10$:

$$y = \frac{1}{12} x^2$$

3.18. Use the midpoint method and symmetry considerations to scan convert the parabola

$$y = 100 - x^2$$

over the interval $-10 \leq x \leq 10$.

3.19. Use the midpoint method and symmetry considerations to scan convert the parabola

$$x = y^2$$

for the interval $-10 \leq y \leq 10$. 

Exercises