Computer Engineering Department

TED University

Lab Assignment 05
CMPE 252 C Programming, Spring 2022

Part 1 (60 points)

In this part, you are asked to complete shape_part1.c program (available in Moodle) which keeps the list of shapes in a text file. Please check the content of the example shapes1.txt below.

Content of shapes1.txt
cube 4 -5 10 3
cylinder -3 4 6 4 5
cube 3 -2 -4 1
sphere 1 3 8 4
cube -4 -1 -2 5

Each line contains a shape data. The data format for each shape type is as follows:
sphere <center-x-coordinate> <center-y-coordinate> <center-z-coordinate> <radius>
cube <corner-x-coordinate> <corner-y-coordinate> <corner-z-coordinate> <side-length>
cylinder <base-center-x-coordinate> <base-center-y-coordinate> <base-center-z-coordinate> <base-radius> <height>

Follow the below steps in your program:

Create **point_t** structure with x (double), y (double) and z (double) coordinates.

Create **sphere_t** structure with center (point_t), radius (double) and volume (double).

Create **cube_t** structure with bottom left furthest corner (point_t), side (double) and volume (double).

Create **cylinder_t** structure with base center (point_t), base radius (double), height (double) and volume (double).

Create union type **shape_data_t** with sphere (sphere_t), cube (cube_t) and cylinder (cylinder_t).

Create enumerated type **class_t** with constants SPHERE, CUBE, CYLINDER.

Create **shape_t** structure with type (class_t) and shape (shape_data_t). type field determines which member of shape contains a value. If type is SPHERE, shape.sphere contains a value. If type is CUBE, shape.cube contains a value. If type is CYLINDER, shape.cylinder contains a value.

Write 3 functions:

- int scanShape(FILE *filep, shape_t *objp);
  scanShape function gets a pointer to FILE and a pointer to shape_t. Reads shape data from the file, and fills shape_t pointed to, by objp. Returns 1 if the read operation is successful; otherwise, returns 0.
  - If the shape is a sphere, the volume is equal to (4/3)*PI*radius*radius*radius. Take PI as 3.141593.
  - If the shape is a cube, the volume is equal to side*side*side.
  - If the shape is a cylinder, the volume is equal to PI*radius*radius*height.

- int loadShapes(shape_t shapes[]);
  loadShapes function gets an array of shape_t. Opens the text file with the entered name. For each array element, reads data by calling scanShape function. Stops reading when scanShape function returns 0. Returns the number of read shapes.

- void printShape(const shape_t *objp);
  printShape function gets a pointer to a constant shape_t. Prints shape information. The format for each shape type is as follows (also see example run). While printing double values, use %.2f as the format specifier.
  Sphere: <center-x-coordinate, center-y-coordinate, center-z-coordinate> <radius> V<volume>
  Cube: <corner-x-coordinate, corner-y-coordinate, corner-z-coordinate> <side-length> V<volume>
  Cylinder: <base-center-x-coordinate, base-center-y-coordinate, base-center-z-coordinate> <base-radius> <height> V<volume>

- **main** function is already provided to you (see shape_part1.c) and it is supposed to remain as it is (you should not change it). In main function, an array of shape_t is declared, loadShapes function is called, and all shapes are printed.

Example Run:

```
Enter the file name to read: shapes1.txt
Opening shapes1.txt
Loading complete
Closing shapes1.txt

Shapes:
Cube: <4.00 -5.00 10.00> <3.00> V<27.00>
Cylinder: <-3.00 4.00 6.00> <4.00> <5.00> V<251.33>
Cube: <3.00 -2.00 -4.00> <1.00> V<1.00>
Sphere: <1.00 3.00 8.00> <4.00> V<268.08>
Cube: <-4.00 -1.00 -2.00> <5.00> V<125.00>
```

Part 2 (40 points)

In this part, you will add the following function to your program in Part 1.

- int hasVolumeGreaterThan(const shape_t *objp, double minVol);
  hasVolumeGreaterThan function gets a pointer to a constant shape_t and a double minVol. Returns 1 if the volume of the shape is greater than minVol; otherwise, returns 0.

- **main** function is already provided to you (take main function from shape_part2.c) and it is supposed to remain as it is (you should not change it). In main function, an array of shape_t is declared, loadShapes function is called, all shapes are printed, and finally, only the shapes whose volume is greater than a user entered number are printed.

Example Run:

```
Enter the file name to read: shapes1.txt
Opening shapes1.txt
Loading complete
Closing shapes1.txt

Shapes:
Cube: <4.00 -5.00 10.00> <3.00> V<27.00>
Cylinder: <-3.00 4.00 6.00> <4.00> <5.00> V<251.33>
Cube: <3.00 -2.00 -4.00> <1.00> V<1.00>
Sphere: <1.00 3.00 8.00> <4.00> V<268.08>
Cube: <-4.00 -1.00 -2.00> <5.00> V<125.00>

Enter minimum volume: 100

The following shapes have are greater than 100.00:
Cylinder: <-3.00 4.00 6.00> <4.00> <5.00> V<251.33>
Sphere: <1.00 3.00 8.00> <4.00> V<268.08>
Cube: <-4.00 -1.00 -2.00> <5.00> V<125.00>
```