

Lab Assignment 06

CMPE 252 C Programming, Spring 2023

Part 1 (60 points)

In this part, you are asked to complete `shape3d_part1.c` program (available in Moodle) which keeps the list of shapes in a text file. Please check the content of the example `shapes3d_1.txt` below.

Content of `3dshapes1.txt`

```
cube 4 -5 3 5
square_prism -3 4 4 5 2
sphere 3 -2 1 3
square_prism 3 1 -2 1 2
cube -4 -1 4 3
```

Each line contains a shape data. The data format for each shape type is as follows:

cube <center x coordinate> <center y coordinate> <center z coordinate> <side-length>

square_prism <center x coordinate> <center y coordinate> <center z coordinate> <base-side-length> < height>.

sphere <center-x-coordinate> <center-y-coordinate> <center z coordinate> <radius>

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) and radius (double).

Create **cube_t** structure with center (point_t) and side (double).

Create **square_prism_t** structure with center (point_t), base-side-length (double) and height (double).

Create union type **shape3d_data_t** with cube (cube_t), square_prism (square_prism_t) and cube (cube_t).

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

Create **shape_t** structure with type (class_t) and shape (shape3d_data_t). type field determines which member of 3d shape contains a value. If type is SPHERE, shape.sphere contains a value. If type is SQUARE_PRISM, shape.square_prism contains a value. If type is CUBE, shape.cube contains a value.

Write 3 functions:

- `int scanShape(FILE *filep, shape_t *objp);`
scanShape function gets a pointer to FILE and a pointer to shape3d_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.
- `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 `%.2lf` as the format specifier.

Cube: <center-x-coordinate center-y-coordinate center-z-coordinate> <side-length> <Volume>

Square_prism: <center-x-coordinate center-y-coordinate center-z-coordinate> <base-side-length height> <Volume>

Sphere: <center-x-coordinate center-y-coordinate center-z-coordinate> <radius> <Volume>

You can use following formulas to calculate volume of each 3d shape:

- Cube: side-length^3
- Square_prism : $\text{base-side-length}^2 \times \text{height}$
- Sphere: $\frac{4}{3} \times \pi \times \text{radius}^3$
- **main** function is already provided to you (see `shape3d_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: `shapes3d_1.txt`

Opening `shapes3d_1.txt`

Loading complete

Closing `shapes3d_1.txt`

Shapes 3D:

Cube: <4.00 -5.00 3.00> <5.00> <125.00>

Square_prism: <-3.00 4.00 4.00> <5.00 2.00> <50.00>

Sphere: <3.00 -2.00 1.00> <3.00> <84.82>

Square_prism: <3.00 1.00 -2.00> <1.00 2.00> <2.00>

Cube: <-4.00 -1.00 4.00> <3.00> <27.00>

Part 2 (40 points)

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

- `int isVolumeBetween(double minVolume, double maxVolume, const shape_t *objp);`
isVolumeBetween function gets two double variables `minVolume` and `maxVolume`, and a pointer to a constant `shape_t`. Returns 1 if the given shape's volume is between `minVolume` and `maxVolume`, otherwise returns 0.
- **main** function is already provided to you (take main function from `shape3d_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 which contain a user entered point are printed.

Example Run:

Enter the file name to read: `shapes3d_1.txt`

Opening `shapes3d_1.txt`

Loading complete

Closing `shapes3d_1.txt`

Shapes 3D:

Cube: <4.00 -5.00 3.00> <5.00> <125.00>

Square_prism: <-3.00 4.00 4.00> <5.00 2.00> <50.00>

Sphere: <3.00 -2.00 1.00> <3.00> <84.82>

Square_prism: <3.00 1.00 -2.00> <1.00 2.00> <2.00>

Cube: <-4.00 -1.00 4.00> <3.00> <27.00>

Enter min and max volumes: `1 29`

Volumes of the following shapes are between 1.00 and 29.00:

Square_prism: <3.00 1.00 -2.00> <1.00 2.00> <2.00>

Cube: <-4.00 -1.00 4.00> <3.00> <27.00>