QUESTION 1 - Functions

/\* Flat Washer Program \*/

/\*

\* Computes the weight of a batch of flat washers.

\*/

#include <stdio.h>

#define PI 3.14159

int main(void)

{

double hole\_diameter; /\* input - diameter of hole \*/

double edge\_diameter; /\* input - diameter of outer edge \*/

double thickness; /\* input - thickness of washer \*/

double density; /\* input - density of material used \*/

double quantity; /\* input - number of washers made \*/

double weight; /\* output - weight of washer batch \*/

double hole\_radius; /\* radius of hole \*/

double edge\_radius; /\* radius of outer edge \*/

double rim\_area; /\* area of rim \*/

double unit\_weight; /\* weight of 1 washer \*/

/\* Get the inner diameter, outer diameter, and thickness.\*/

printf("Inner diameter in centimeters> ");

scanf("%lf", &hole\_diameter);

printf("Outer diameter in centimeters> ");

scanf("%lf", &edge\_diameter);

printf("Thickness in centimeters> ");

scanf("%lf", &thickness);

/\* Get the material density and quantity manufactured. \*/

printf("Material density in grams per cubic centimeter> ");

scanf("%lf", &density);

printf("Quantity in batch> ");

scanf("%lf", &quantity);

/\* Compute the rim area. \*/

hole\_radius = hole\_diameter / 2.0;

edge\_radius = edge\_diameter / 2.0;

rim\_area = PI \* edge\_radius \* edge\_radius - PI \* hole\_radius \* hole\_radius;

/\* Compute the weight of a flat washer. \*/

unit\_weight = rim\_area \* thickness \* density;

/\* Compute the weight of the batch of washers. \*/

weight = unit\_weight \* quantity;

/\* Display the weight of the batch of washers. \*/

printf("\nThe expected weight of the batch is %.2f", weight);

printf(" grams.\n");

return (0);

}

Revise the flat washer program to use function subprograms ***find\_area***, ***find\_rim\_area***, ***find\_unit\_weight*** and ***instruct***.

/\* **sample output is going to be the same for the revised one:**

\*/

#include <stdio.h>

#include <math.h>

#define PI 3.14159

/\* Add function prototypes \*/

int main (void)

{

/\* Declare variables \*/

/\* Give the user instructions. \*/

/\* Get the inner diameter, outer diameter, and thickness. \*/

/\* Get the material density and quantity manufactured. \*/

/\* Compute the rim area. \*/

/\* Compute the weight of a single flat washer. \*/

/\* Compute the weight of the batch of washers. \*/

/\* Display the weight of the batch of washers. \*/

return (0);

}

/\* Displays instructions to a user of program to compute the weight of a batch of flat washers \*/

------------- instruct(----------------)

{

}

/\* Computes the area of a circle with radius r.

\* Pre: r is defined and is > 0.

\* PI is a constant macro representing an approximation of pi.

\* Library math.h is included.

\*/

---------------- find\_area(----------------------)

{

}

/\*

\* Computes the area of an annular ring with inner radius of inner

\* and outer radius of outer.

\* Pre: inner and outer are defined and are > 0.

\* Function find\_area() is defined.

\*/

----------------- find\_rim\_area(---------------------------------------------)

{

}

/\*

\* Computes the unit weight of a flat object with an area of area,

\* with a thickness of thickness, and with a density of density.

\* Pre: area, thickness and density are defined and are > 0.

\*/

---------------------------- find\_unit\_weight(-----------------------------------------------------------)

{

}

**Sample Output:**

Inner diameter in centimeters> 1.2

Outer diameter in centimeters> 2.4

Thickness in centimeters> 0.1

Material density in grams per cubic centimeter> 7.87

Quantity in batch> 1000

The expected weight of the batch is 2670.23 grams.

QUESTION 2 - Functions

Write a function which computes the departure time required to reach a destination that is a given (positive) distance away, based on supplied arrival time and estimated average speed. Arrival must be on same day as departure. Also write a driver to test your function.

Sample Output is:

|  |
| --- |
| Enter arrival time as integer on a 24 hour clock. For example, 8:30 PM would be entered as 2030  Arrival time>2100  Enter the distance in km>5  Enter anticipated average speed (including stops) in km/hr> 6  You need to leave at 2010. |

QUESTION 3 – Numerical Analysis in C

Iterative approximations for root finding is an important concept in numerical analysis. There are various methods used for this purpose, e.g. Bisection.

First lets see the basics of the problem:

is a root of an equation if

Bisection methods continues iteratively until either the true root is found or approximates by less than epsilon. The method first checks whether there is a sign change in the given interval, e.g. :

1 root

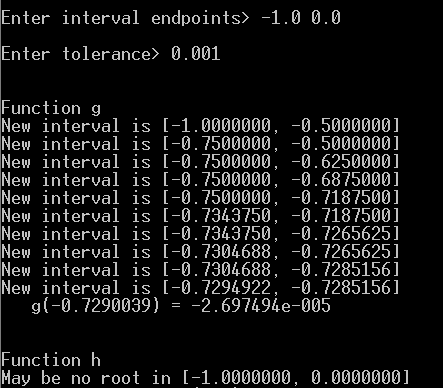
3 roots

There are 3 possibilities when the interval [, ] is bisected, either the root is in the half interval [, ], root is in the half interval [, ], or

For an arbitrary given function g🡪, and h🡪, how can you apply Bisection method in C?

Hint: Your program should contain 4 functions:

* int main(void)
* double bisect(double xleft, double xright, double epsilon, double f(double fargs))
* double g(double x)
* double h(double x)



QUESTION 4 - Recursion

Fill in the following recursive function that calculates the value of a number (base) raised to a power. Assume that power is a nonnegative integer.

int power\_raiser(int base, int power)

{

int ans;

if(power == ......................)

ans = ..............................;

else

ans = .........................\*...................................;

return ans;

}

QUESTION 5 – Recursion

Write a recursive function which checks whether a given number is prime or not.

Introduction to Graphics in C

# OpenGL Setup / Functions:

* Download GLUT (you can use the files I shared in Moodle)
* Copy
  + glut32.dll to C:\Windows\System32
  + glut32.lib to ......\CodeBlocks\MinGW\lib
  + glut.h to .......\CodeBlocks\MinGW\includeGL
* Open CodeBlocks
  + New Project 🡪 GLUT project
  + Name your project
  + Please select GLUT’s location: ........\CodeBlocks\MinGW
  + Finish
  + Add the file I shared under the project
  + Right click project🡪Build Options🡪Linker Settings

|  |  |
| --- | --- |
|  |  |

* + - Add glut32.lib

Figure 1

* + Run the shared code and you will see a colored 3D cube:

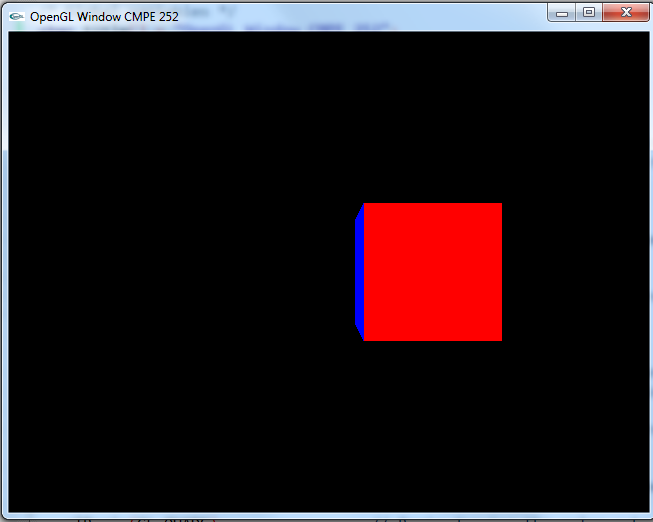


Figure 2

* **Investigate GLUT and all graphics related functions.**

# Part 1 / Switch:

* Open your old code and add a global variable called *zRotation* which is initially 0.
* Call ***glRotatef*** function before drawing your cube with *zRotation* in z-axis.
* Now add a Key function: ***static void Key(unsigned char key)***
  + This function checks the key using Switch:
    - If the case is 27, exits the program.
    - If the case is X, increments *zRotation* by 5 and calls ***glutPostRedisplay***().
* Do not forget to call your ***Key*** function using ***glutKeyboardFunc*** inside ***main*** function.

Output should look like Figure 3-a, i.e. cube should be rotated in z-axis by 5 degrees whenever X key is pressed.

# Part 2 / For-Loop:

* Draw a green spiral using GL\_LINE\_STRIP.
* Hint: You should use a for-loop which starts from angle=0, finishes when angle is larger than and angle is incremented by 0.1 in each iteration.
* In each step, x position is sin(angle), z position is cos(angle) and y is incremented by 0.01 (y is initially -0.5)

The output should look like Figure 3-b.

|  |  |
| --- | --- |
|  |  |
| Figure 3-a | Figure 3-b |