



EE2028A

C Programming

Laboratory Exercise (LAB-I)

Name:

Matriculation Number:

Submission instructions:

1. **Test your code on your computer first** before submitting.
2. You must name your functions exactly as the question states.
3. **DEADLINE: Tuesday 4 Feb 2020 / Thursday 6 Feb 2020**
4. **LumiNUS “Lab 1 Assignment Submission Folder”**
5. **Grading: Your assignment will be graded out of 50 marks and the final weight of this assignment is 5%.**
6. You are expected to follow the guidelines given below:
 - a. Use meaningful variable names while programming. It’s a good practice to develop good programming skills and enables readability.
 - b. Explain the code with proper comments; Comments must be meaningful and descriptive;
 - c. Please adhere to the report deadlines and any late submissions are not accepted.
7. Please prepare the report in **PDF** format **only**.
8. Submit the following:

Submit the compressed file	Contains
MATRICULATION_NUMBER_ASSIGNMENT1_NAME (First Name).zip	REPORT_MATRICULATION_NUMBER_ASSIGNMENT1_NAME (First Name).pdf
	Your working C code, ONLY .c file

What you need to add into this report for submission? - YOUR OUTPUT:

- Program Code (attach in the ZIP file – see the guidelines for submission) addition to the .c file that you need to submit
 - Code should be well written with meaningful variables and **comments**.
- In THIS report, **screenshot your results and paste**. Make sure it is visible, readable and clear.
 - For Question 1:** Highlight the errors on the given code and give the reasoning with the correct code.
 - For Question 2:** Your code and your screenshot for the given problem.
 - For Question 3:** Your code and your screenshots for the given 4 cases.

DO NOT FORGET TO SEND YOUR .C FILE WITH ALL YOUR WORKING CODES BESIDES ATTACHING THE CODES INTO THE REPORT.

NOTE: Start your answers from here. Use as much space as needed.

PROBLEM 1:

mistakes

```
#include <stdio.h>
#include <math.h>
#define PI 3.14f
int main() //Debugging Exercise
{
    printf("Debugging Exercise \n");
    float radius_cylinder, radius_cone, height_cylinder, height_cone;
    float CYLINDER_Volume, CONE_Volume;
    float lCONE_SA, lCONE_SA_bottom;
    float SA_large_bottom-cone, SA_small_bottom-cone, circumference-cylinder;
    float SA_large_bottom-cone, SA_small_bottom-cone, circumference-cylinder;

    // ***** INPUT *****
    scanf("Please enter the radius of a cylinder:\n", &radius_cylinder);
    printf("Please enter the height of a cylinder:\n");
    scanf("%f", &height_cylinder);
    radius_cone = radius_cylinder*2;
    height_cone = height_cylinder/3;

    // ***** VOLUME *****
    CYLINDER_Volume = PI * radius_cylinder * radius_cylinder * height_cylinder;
    CONE_Volume = (1.0/3) * PI * radius_cone * radius_cone * height_cone;
    printf("Volume of a Cylinder = %.3f\n", CYLINDER_Volume);
    printf("Volume of cone is : %.3f\n", CONE_Volume);
    printf("Total volume of the arrow is : %.3f\n", (CONE_Volume+CYLINDER_Volume));

    // ***** CYLINDER AREA *****
    lCYLINDER_BSA = PI * radius_cylinder * radius_cylinder;
    printf("Bottom Surface Area of a cylinder = %.3f\n", lCYLINDER_BSA);
    circumference-cylinder = 2 * PI * radius_cylinder;
    lCYLINDER_LSA = circumference-cylinder * height_cylinder;
    printf("Lateral Surface Area of a cylinder = %.3f\n", lCYLINDER_LSA);

    // ***** CONE AREA *****
    lCONE_SA = PI * radius_cone * sqrt(radius_cone * radius_cone + height_cone * height_cone);
    SA_large_BSA = PI * radius_cone * radius_cone;
    SA_small_BSA = PI * radius_cylinder * radius_cylinder;
    lCONE_SA_bottom = SA_large_BSA - SA_small_BSA;
    printf("Surface area of cone is: %.3f, Surface bottom area of cone is: %.3f\n", lCONE_SA, lCONE_SA_bottom);

    // ***** TOTAL AREA OF ARROW *****
    printf("Total area of the arrow is : %.3f\n", (lCONE_SA+lCONE_SA_bottom+lCYLINDER_LSA+lCYLINDER_BSA));
    return 0;
}
```

My code:

```
#include <stdio.h>
#include <math.h>
#define PI 3.14f
int main() //Debugging Exercise
{
    printf("Debugging Exercise \n");
    float radius_cylinder, _radius_cone, height_cylinder, _height_cone;
    float CYLINDER_Volume, CONE_Volume;
    float CONE_SA_1, CONE_SA_bottom_1; //variable name cannot start with number
    float CYLINDER_BSA_2, CYLINDER_LSA_2; //variable name cannot start with number
    float SA_large_bottom_cone, SA_small_bottom_cone, circumference_cylinder; //cannot use hyphen as can be mistaken for a minus sign

    // ***** INPUT*****
    printf("Please enter the radius of a cylinder: ");
    scanf("%f", &radius_cylinder); // had to separate the scanf statement to a printf and scanf as the user will not receive
    //the prompt to input their radius value in the original code
    printf("Please enter the height of a cylinder:");
    scanf("%f", &height_cylinder); //never add ampersand
    _radius_cone = radius_cylinder*2;
    _height_cone = height_cylinder/2; //variable was initialized as _height_cone NOT height_cone

    //*****VOLUME*****
    CYLINDER_Volume = PI * radius_cylinder * radius_cylinder * height_cylinder;
    CONE_Volume = (1.0/3) * PI * _radius_cone * _height_cone;
    printf("Volume of a Cylinder = %.3f\n", CYLINDER_Volume);
    printf("Volume of a cone is : %.3f\n", CONE_Volume);
    printf("Total volume of the arrow is : %.3f \n", (CONE_Volume + CYLINDER_Volume));

    //*****CYLINDER AREA*****
    CYLINDER_BSA_2 = PI * pow(radius_cylinder,2); // to square need to use pow(radius_cylinder, 2)
    printf("Bottom Surface Area of a cylinder = %.3f\n", CYLINDER_BSA_2); //never add ;
    circumference_cylinder = 2 * PI * radius_cylinder;
    CYLINDER_LSA_2 = circumference_cylinder * height_cylinder;
    printf("Lateral Surface Area of a cylinder = %.3f\n", CYLINDER_LSA_2); //supposed to be %.3f and not %.3d

    //*****CONE AREA*****
    CONE_SA_1 = PI * _radius_cone * sqrt(_radius_cone * _radius_cone + _height_cone * _height_cone);
    SA_large_bottom_cone = PI * _radius_cone * _radius_cone; //previously written as SA_large_bottom_cone, SA_large_BC
    //has not been initialized
    SA_small_bottom_cone = PI * radius_cylinder * radius_cylinder; //previously written as SA_small_bottom_cone, SA_small_BC
    // has not been initialized
    CONE_SA_bottom_1 = SA_large_bottom_cone - SA_small_bottom_cone; //hyphen change to underscore
    printf("Surface area of cone is: %.3f, Surface bottom area of cone is : %.3f \n", CONE_SA_1, CONE_SA_bottom_1); // %3c does
    //not work since CONE_SA_1 is a float and not a character/string ,variable name problem is same

    //*****TOTAL AREA OF ARROW*****
    printf("Total area of the arrow is : %.3f\n", (CONE_SA_1 + CONE_SA_bottom_1 + CYLINDER_LSA_2 + CYLINDER_BSA_2)); // % sign is
    //missing in front of the .3f

    return 0;
}
```

PROBLEM 2: my code

```

#include <stdio.h>
#include <math.h>
#define rydberg_Constant 2.179e-18

int main(void) {
    int Z = 1; // atomic number of hydrogen
    float energy_levels[6];
    printf("The atomic number of the hydrogen %d, the Rydberg constant %g J. \n", Z, rydberg_Constant);
    int n;
    double E;
    for(n = 1; n < 7; n++) {
        E = -(rydberg_Constant * pow(Z, 2)) / pow(n, 2);
        energy_levels[n-1] = E;
        printf("Energy of the electron when n=%d is E(%d) = %g J. \n", n, n, E);
    }
    printf("%g", energy_levels[0]);
    int k;
    float energy_dif;
    for(k=0; k < 6; k++) {
        energy_dif = energy_levels[k+1] - energy_levels[k];
        printf("The energy difference between levels %d-%d: E(%d) - E(%d) = %g J. \n", (k+1), k, (k+1), k, energy_dif);
    }
}

```

My answer:

```

C:\Users\Vignesh\OneDrive\Desktop\EE2028A\lab1_question2.exe
The atomic number of the hydrogen 1, the Rydberg constant 2.179e-018 J.
Energy of the electron when n=1 is E(1) = -2.179e-018 J.
Energy of the electron when n=2 is E(2) = -5.4475e-019 J.
Energy of the electron when n=3 is E(3) = -2.42111e-019 J.
Energy of the electron when n=4 is E(4) = -1.36188e-019 J.
Energy of the electron when n=5 is E(5) = -8.716e-020 J.
Energy of the electron when n=6 is E(6) = -6.05278e-020 J.
-2.179e-018The energy difference between levels 1-0: E(1) - E(0) = 1.63425e-018 J.
The energy difference between levels 2-1: E(2) - E(1) = 3.02639e-019 J.
The energy difference between levels 3-2: E(3) - E(2) = 1.05924e-019 J.
The energy difference between levels 4-3: E(4) - E(3) = 4.90275e-020 J.
The energy difference between levels 5-4: E(5) - E(4) = 2.66322e-020 J.
The energy difference between levels 6-5: E(6) - E(5) = 8.716e-020 J.

Process returned 0 (0x0)   execution time : 0.040 s
Press any key to continue.

```

PROBLEM 3:MY CODE

```

#include <stdio.h>
#include <math.h>
#include <stdbool.h>
bool is_year_leap(int any_year)
{
    if((any_year%4==0)){
        return true;
    }
    else
    {
        return false;
    }
}

int main(){
    int input_days, starting_month, starting_day, starting_year;
    printf("Your input days: ");
    scanf("%d", &input_days);
    printf("Your starting month: ");
    scanf("%d", &starting_month);
    printf("Your starting day: ");
    scanf("%d", &starting_day);
    printf("Your input year: ");
    scanf("%d", &starting_year);
    printf("%d days from %d/%d/%d = ", input_days, starting_day, starting_month, starting_year);

    int days_in_leap_months[12] = {31,29,31,30,31,30,31,31,30,31,30,31};
    int days_in_non_leap_months[12] = {31,28,31,30,31,30,31,31,30,31,30,31};
    int year_counter = 0, month_counter = 0;

    while((input_days>=366 && is_year_leap(starting_year) == true) || (input_days>=365 && is_year_leap(starting_year)==false)){
        if((is_year_leap(starting_year)==true && starting_month<=2) || (is_year_leap(starting_year+1) && starting_month>2)){
            input_days = input_days - 366;
            year_counter+=1;
            starting_year+=1;
        }
        else{
            input_days = input_days - 365;
            year_counter+=1;
            starting_year+=1;
        }
    }
    while(((input_days >= days_in_leap_months[starting_month-1] && is_year_leap(starting_year)==true) || ((input_days>=days_in_non_leap_months[starting_month-1] && is_year_leap(starting_year)==false) && starting_month>2)){
        if(is_year_leap(starting_year)){
            input_days = input_days - days_in_leap_months[starting_month-1];
            month_counter+=1;
            starting_month+=1;
        }
        else{
            input_days = input_days-days_in_non_leap_months[starting_month-1];
            month_counter+=1;
            starting_month+=1;
        }
    }
    printf("%d years %d months %d days", year_counter, month_counter, input_days);

    return 0;
}

```




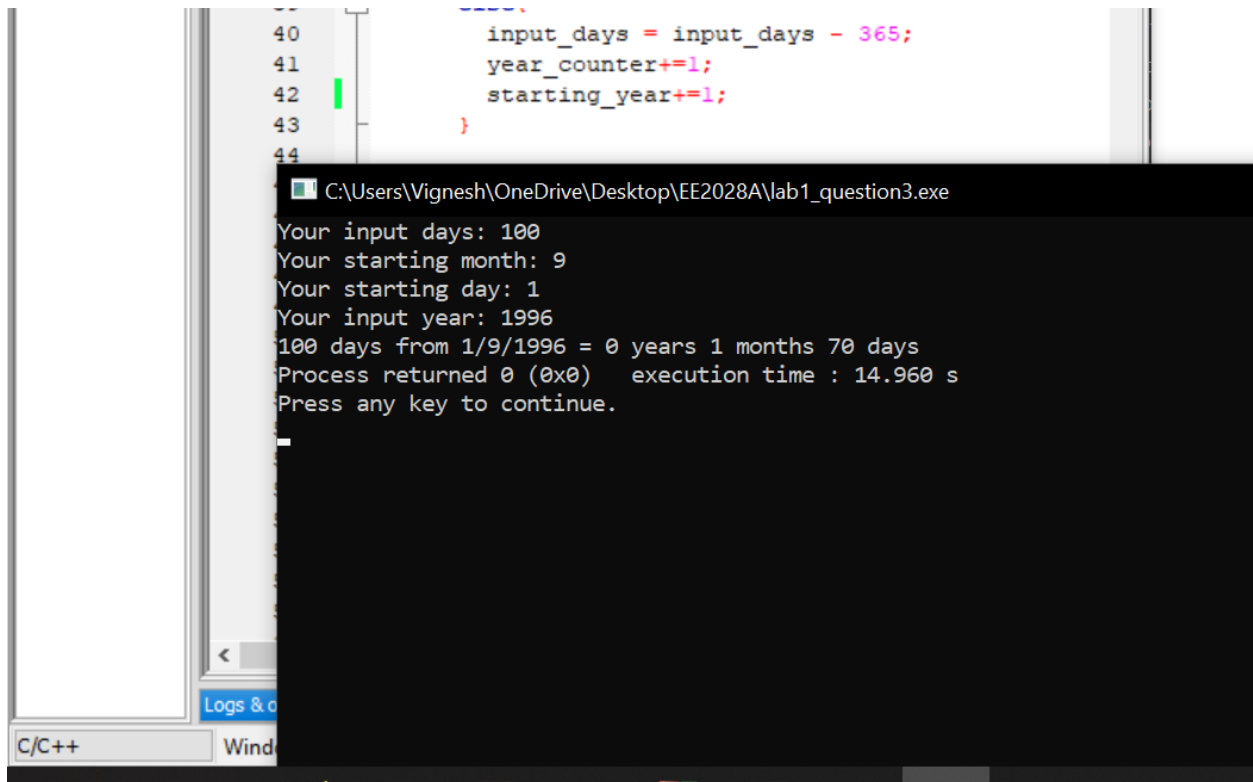
Screenshot saved
The screenshot was added to your
OneDrive.

```
C:\Users\Vignesh\OneDrive\Desktop\EE2028A\lab1_question3.exe
Your input days: 1189
Your starting month: 1
Your starting day: 1
Your input year: 2020
1189 days from 1/1/2020 = 3 years 1 months 62 days
Process returned 0 (0x0)   execution time : 7.691 s
Press any key to continue.
```

```
C:\Users\Vignesh\OneDrive\Desktop\EE2028A\lab1_question3.exe
Your input days: 873
Your starting month: 2
Your starting day: 1
Your input year: 2013
873 days from 1/2/2013 = 2 years 1 months 115 days
Process returned 0 (0x0)   execution time : 11.011 s
Press any key to continue.
```

```
C:\Users\Vignesh\OneDrive\Desktop\EE2028A\lab1_question3.exe
Your input days: 367
Your starting month: 6
Your starting day: 15
Your input year: 2015
367 days from 15/6/2015 =
Process returned 0 (0x0)   execution time : 14.021 s
Press any key to continue.
```





```
40         input_days = input_days - 365;  
41         year_counter+=1;  
42         starting_year+=1;  
43     }  
44  
C:\Users\Vignesh\OneDrive\Desktop\EE2028A\lab1_question3.exe  
Your input days: 100  
Your starting month: 9  
Your starting day: 1  
Your input year: 1996  
100 days from 1/9/1996 = 0 years 1 months 70 days  
Process returned 0 (0x0)   execution time : 14.960 s  
Press any key to continue.  
C/C++ Wind
```