EE2028A

C Programming

Laboratory Exercise (LAB-I)

Name: Anurag Chatterjee

Matriculation Number: A0200533Y

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**: **Tuesday4 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:

a. 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.

b. In THIS report, **screenshot your results and paste.** Make sure it is visible, readable and clear.
   - **For Question1:**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 Question3:**Your code and your screenshots for the given 4 cases.

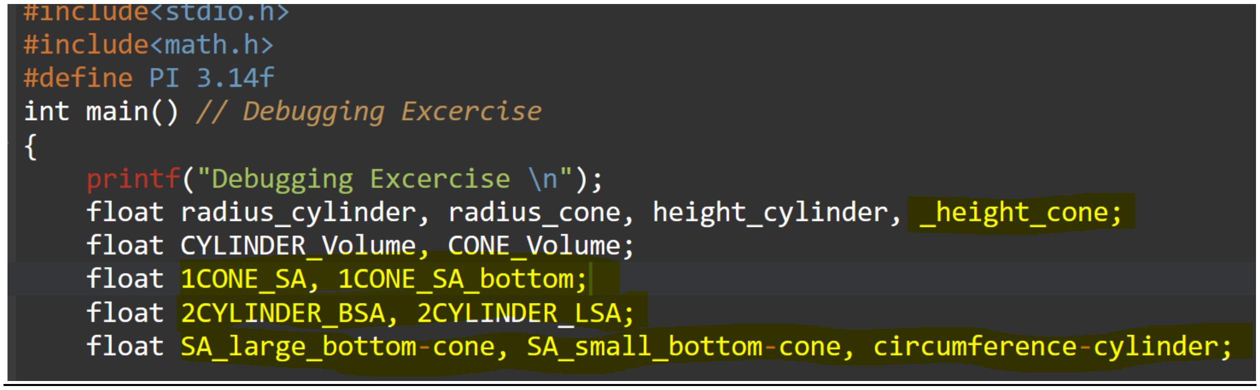
DO NOT FORGET TO SEND YOUR .C FILE WITH ALL YOUR WORKING CODES BESIDES ATTACHING THE CODES INTO THE REPORT.

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

RESULTS SCREENSHOTS

PROBLEM 1:

- **ERROR PART I:**



```
#include<stdio.h>
#include<math.h>
#define PI 3.14f
int main() // Debugging Excercise
{
    printf("Debugging Excercise \n");
    float radius_cylinder, radius_cone, height_cylinder, _height_cone;
    float CYLINDER_Volume, CONE_Volume;
    float 1CONE_SA, 1CONE_SA_bottom;
    float 2CYLINDER_BSA, 2CYLINDER_LSA;
    float SA_large_bottom-cone, SA_small_bottom-cone, circumference-cylinder;
```

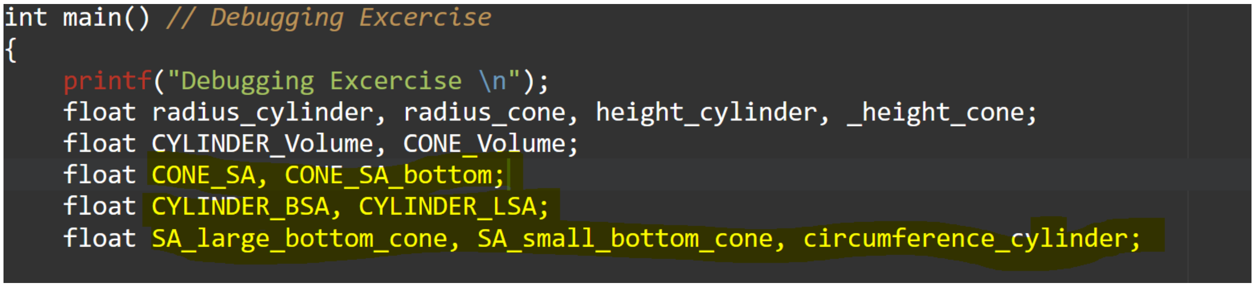
Highlighted Are The Errors:

a) 1CONE_SA, 1CONE_SA_bottom, 2CYLINDER_BSA and 2CYLINDER_LSA variables should not start with a digit

b) The "-" syntax should not be present in last highlighted line (between bottom cone and circumference cylinder)

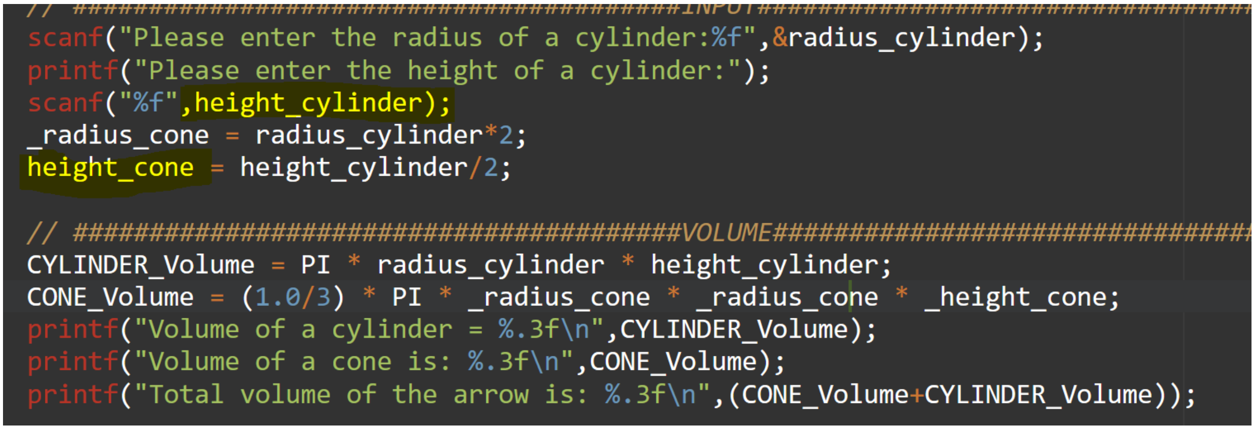
Note: Pls ignore highlighted code of _height_cone!

- **CORRECTION IN HIGHLIGHTED CODE: PART I**



```
int main() // Debugging Excercise
{
    printf("Debugging Excercise \n");
    float radius_cylinder, radius_cone, height_cylinder, _height_cone;
    float CYLINDER_Volume, CONE_Volume;
    float CONE_SA, CONE_SA_bottom;
    float CYLINDER_BSA, CYLINDER_LSA;
    float SA_large_bottom_cone, SA_small_bottom_cone, circumference_cylinder;
```

- **ERROR PART II:**



```
// ########################################INPUT########################################
scanf("Please enter the radius of a cylinder:%f",&radius_cylinder);
printf("Please enter the height of a cylinder:");
scanf("%f",height_cylinder);
_radius_cone = radius_cylinder*2;
height_cone = height_cylinder/2;

// ########################################VOLUME#####################################
CYLINDER_Volume = PI * 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 a cone is: %.3f\n",CONE_Volume);
printf("Total volume of the arrow is: %.3f\n",(CONE_Volume+CYLINDER_Volume));
```

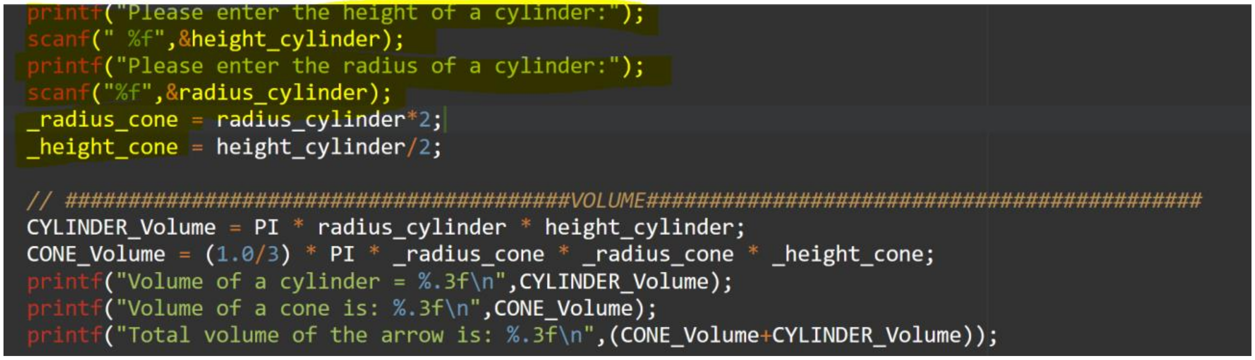
Highlighted Are The Errors:

a) height_cylinder should have an "&" before it in the scanf() function

b) height_cone undeclared in the old code where _height_cone is declared

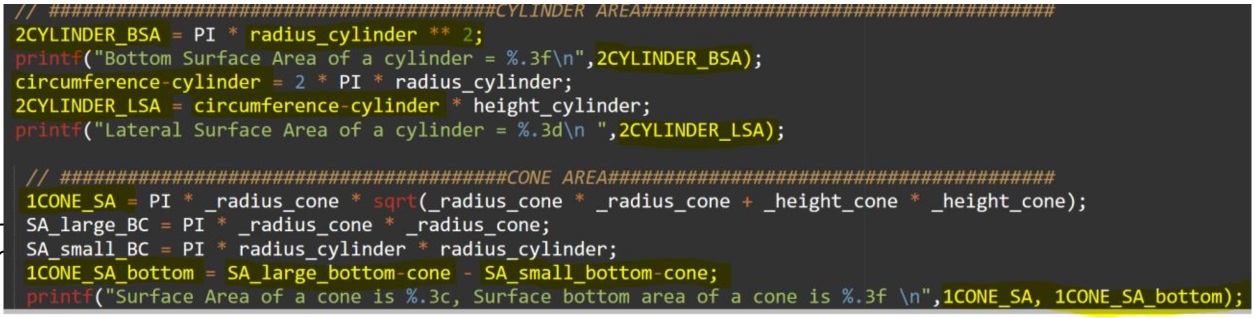
c) For scanf("Please enter the radius of a cylinder: %f"- Words don't get displayed. Need to insert this above printf("Please enter the radius of a cylinder"). Same goes for height of a cylinder which is printf("Please enter the height of a cylinder")

- **CORRECTION IN HIGHLIGHTED CODE: PART II**



```c
printf("Please enter the height of a cylinder:");
scanf(" %f",&height_cylinder);
printf("Please enter the radius of a cylinder:");
scanf("%f",&radius_cylinder);
_radius_cone = radius_cylinder*2;
_height_cone = height_cylinder/2;

// ########################################VOLUME##############################################
CYLINDER_Volume = PI * 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 a cone is: %.3f\n",CONE_Volume);
printf("Total volume of the arrow is: %.3f\n",(CONE_Volume+CYLINDER_Volume));
```

- **ERROR PART III:**



```c
// ########################################CYLINDER AREA########################################
2CYLINDER_BSA = PI * radius_cylinder ** 2;
printf("Bottom Surface Area of a cylinder = %.3f\n",2CYLINDER_BSA);
circumference-cylinder = 2 * PI * radius_cylinder;
2CYLINDER_LSA = circumference-cylinder * height_cylinder;
printf("Lateral Surface Area of a cylinder = %.3d\n ",2CYLINDER_LSA);

 // ########################################CONE AREA########################################
 1CONE_SA = PI * _radius_cone * sqrt(_radius_cone * _radius_cone + _height_cone * _height_cone);
 SA_large_BC = PI * _radius_cone * _radius_cone;
 SA_small_BC = PI * radius_cylinder * radius_cylinder;
 1CONE_SA_bottom = SA_large_bottom-cone - SA_small_bottom-cone;
 printf("Surface Area of a cone is %.3c, Surface bottom area of a cone is %.3f \n",1CONE_SA, 1CONE_SA_bottom);
```

Highlighted Are The Errors:

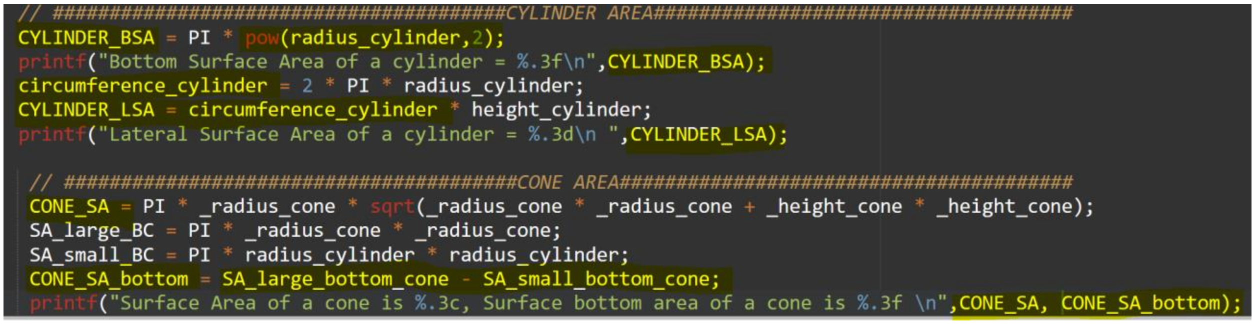
a) 2CYLINDER_BSA, 1CONE_SA, 2CYLINDER_LSA, and 1CONE_SA_bottom should not start with a digit

b) The "-" syntax should not be present in highlighted line (between bottom cone and circumference cylinder)

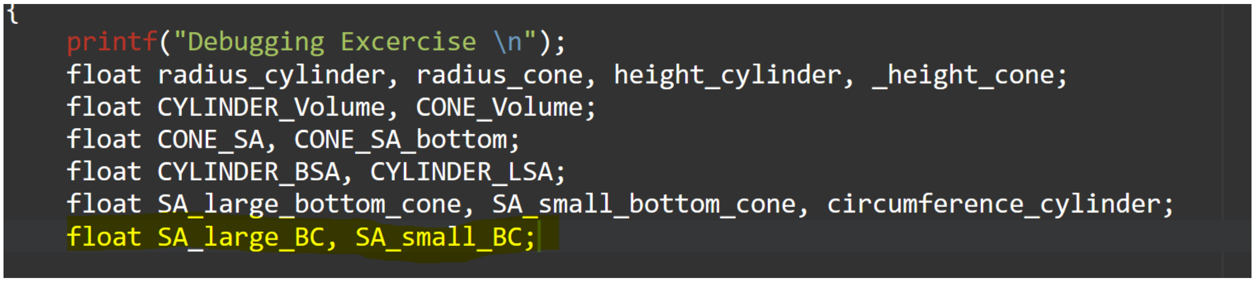
c) For power, pow() should be used between radius_cylinder and 2

d) NOTE: Undeclared variables: SA_large_BC and SA_small_BC

- **CORRECTION IN HIGHLIGHTED CODE: PART III**

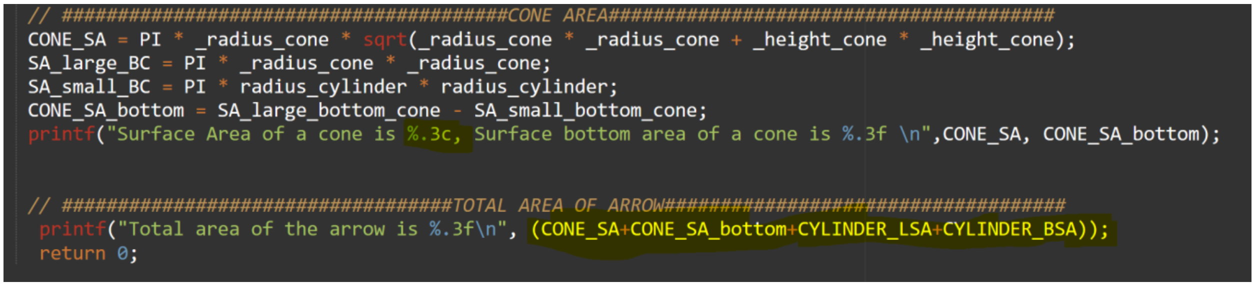


```
// ########################################CYLINDER AREA########################################
CYLINDER_BSA = PI * pow(radius_cylinder,2);
printf("Bottom Surface Area of a cylinder = %.3f\n",CYLINDER_BSA);
circumference_cylinder = 2 * PI * radius_cylinder;
CYLINDER_LSA = circumference_cylinder * height_cylinder;
printf("Lateral Surface Area of a cylinder = %.3d\n ",CYLINDER_LSA);

 // ########################################CONE AREA########################################
 CONE_SA = PI * _radius_cone * sqrt(_radius_cone * _radius_cone + _height_cone * _height_cone);
 SA_large_BC = PI * _radius_cone * _radius_cone;
 SA_small_BC = PI * radius_cylinder * radius_cylinder;
 CONE_SA_bottom = SA_large_bottom_cone - SA_small_bottom_cone;
 printf("Surface Area of a cone is %.3c, Surface bottom area of a cone is %.3f \n",CONE_SA, CONE_SA_bottom);
```



```
{
    printf("Debugging Excercise \n");
    float radius_cylinder, radius_cone, height_cylinder, _height_cone;
    float CYLINDER_Volume, CONE_Volume;
    float CONE_SA, CONE_SA_bottom;
    float CYLINDER_BSA, CYLINDER_LSA;
    float SA_large_bottom_cone, SA_small_bottom_cone, circumference_cylinder;
    float SA_large_BC, SA_small_BC;
```

- **ERROR PART IV:**



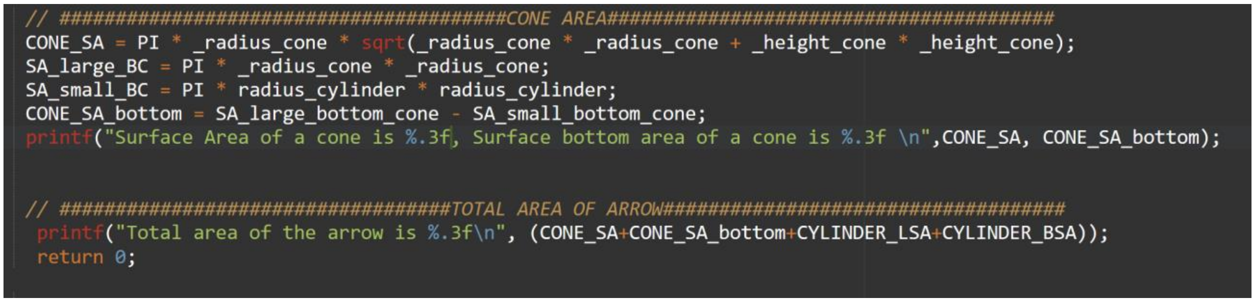
```
 // ########################################CONE AREA########################################
 CONE_SA = PI * _radius_cone * sqrt(_radius_cone * _radius_cone + _height_cone * _height_cone);
 SA_large_BC = PI * _radius_cone * _radius_cone;
 SA_small_BC = PI * radius_cylinder * radius_cylinder;
 CONE_SA_bottom = SA_large_bottom_cone - SA_small_bottom_cone;
 printf("Surface Area of a cone is %.3c, Surface bottom area of a cone is %.3f \n",CONE_SA, CONE_SA_bottom);


 // #####################################TOTAL AREA OF ARROW########################################
  printf("Total area of the arrow is %.3f\n", (CONE_SA+CONE_SA_bottom+CYLINDER_LSA+CYLINDER_BSA));
  return 0;
```

Highlighted Are The Errors:

a) 2CYLINDER_BSA, 1CONE_SA, 2CYLINDER_LSA, and 1CONE_SA_bottom should not start with a digit

b) %.3c should be %.3f as it's a float not a character

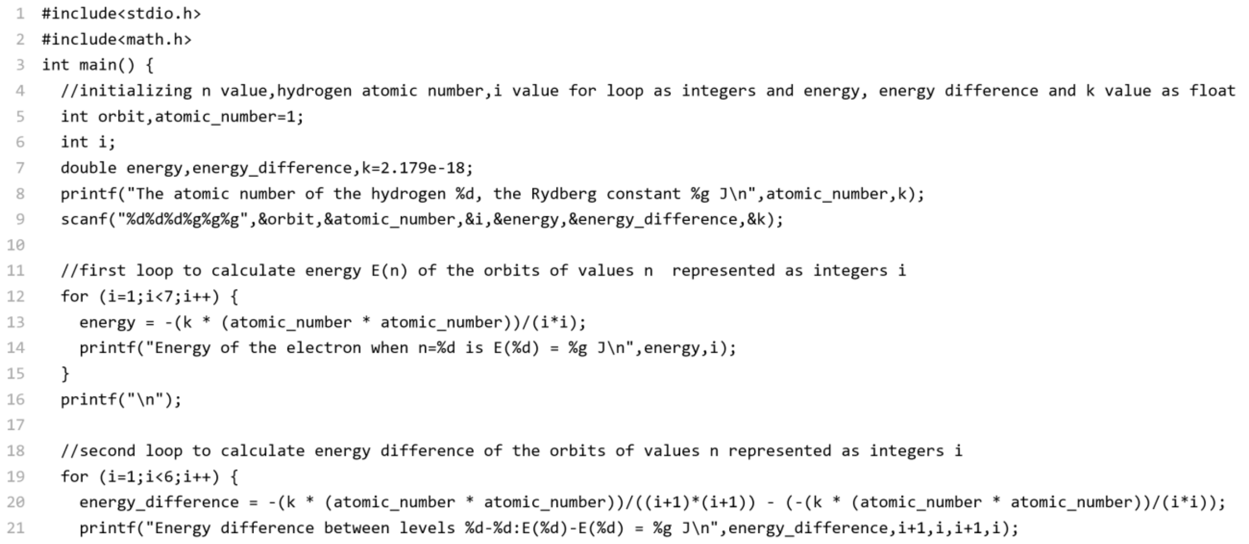
- **CORRECTION IN HIGHLIGHTED CODE: PART IV**



```
// ########################################CONE AREA########################################
CONE_SA = PI * _radius_cone * sqrt(_radius_cone * _radius_cone + _height_cone * _height_cone);
SA_large_BC = PI * _radius_cone * _radius_cone;
SA_small_BC = PI * radius_cylinder * radius_cylinder;
CONE_SA_bottom = SA_large_bottom_cone - SA_small_bottom_cone;
printf("Surface Area of a cone is %.3f, Surface bottom area of a cone is %.3f \n",CONE_SA, CONE_SA_bottom);


// ###################################TOTAL AREA OF ARROW####################################
 printf("Total area of the arrow is %.3f\n", (CONE_SA+CONE_SA_bottom+CYLINDER_LSA+CYLINDER_BSA));
 return 0;
```

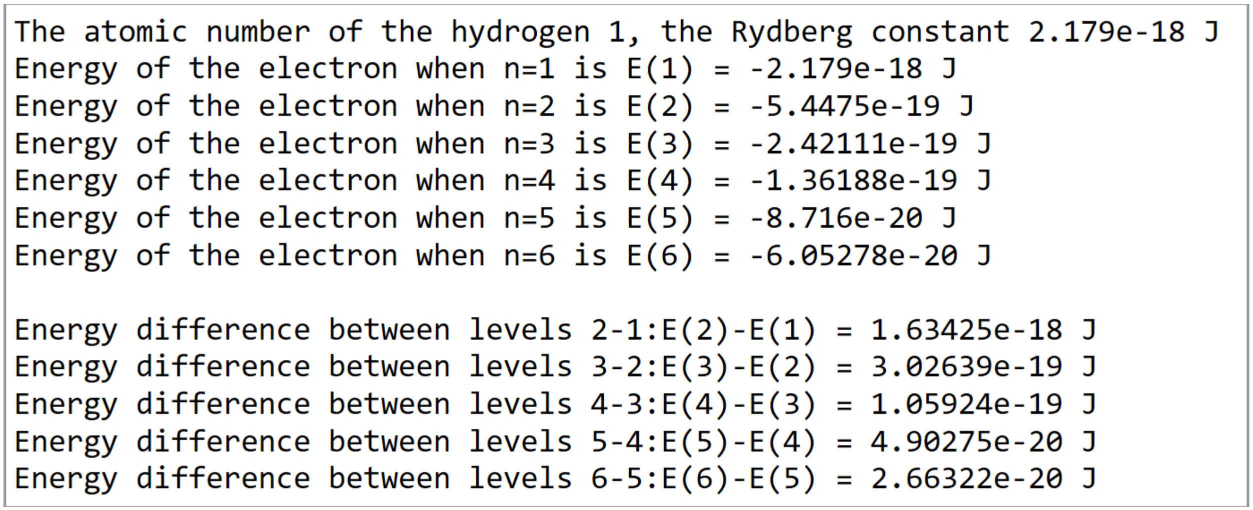
PROBLEM 2:

- **PART I: CODE**



```
#include<stdio.h>
#include<math.h>
int main() {
  //initializing n value,hydrogen atomic number,i value for loop as integers and energy, energy difference and k value as float
  int orbit,atomic_number=1;
  int i;
  double energy,energy_difference,k=2.179e-18;
  printf("The atomic number of the hydrogen %d, the Rydberg constant %g J\n",atomic_number,k);
  scanf("%d%d%d%g%g%g",&orbit,&atomic_number,&i,&energy,&energy_difference,&k);

  //first loop to calculate energy E(n) of the orbits of values n  represented as integers i
  for (i=1;i<7;i++) {
    energy = -(k * (atomic_number * atomic_number))/(i*i);
    printf("Energy of the electron when n=%d is E(%d) = %g J\n",energy,i);
  }
  printf("\n");

  //second loop to calculate energy difference of the orbits of values n represented as integers i
  for (i=1;i<6;i++) {
    energy_difference = -(k * (atomic_number * atomic_number))/((i+1)*(i+1)) - (-(k * (atomic_number * atomic_number))/(i*i));
    printf("Energy difference between levels %d-%d:E(%d)-E(%d) = %g J\n",energy_difference,i+1,i,i+1,i);
```

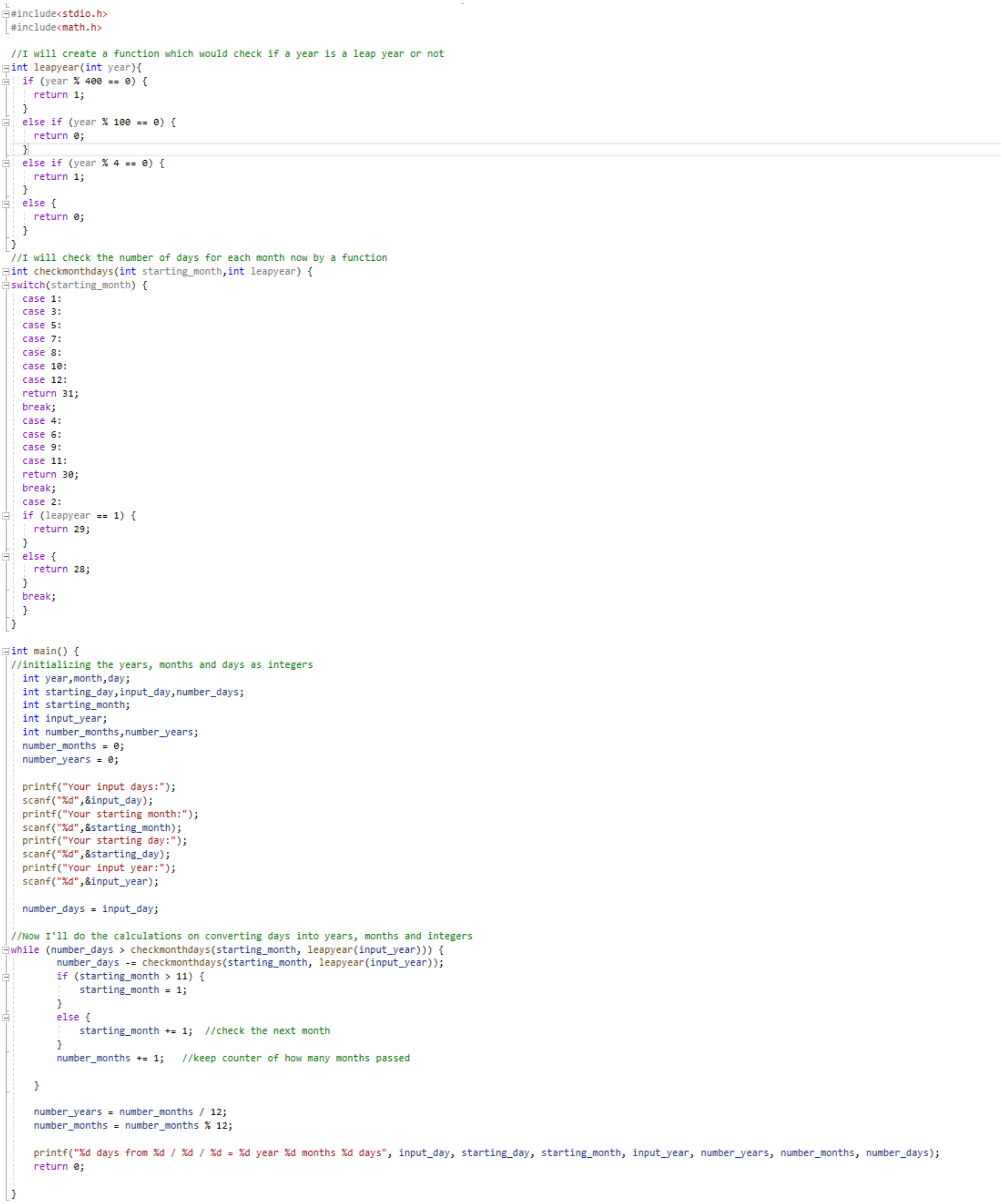
- **OUTPUT: PART I**



```
The atomic number of the hydrogen 1, the Rydberg constant 2.179e-18 J
Energy of the electron when n=1 is E(1) = -2.179e-18 J
Energy of the electron when n=2 is E(2) = -5.4475e-19 J
Energy of the electron when n=3 is E(3) = -2.42111e-19 J
Energy of the electron when n=4 is E(4) = -1.36188e-19 J
Energy of the electron when n=5 is E(5) = -8.716e-20 J
Energy of the electron when n=6 is E(6) = -6.05278e-20 J

Energy difference between levels 2-1:E(2)-E(1) = 1.63425e-18 J
Energy difference between levels 3-2:E(3)-E(2) = 3.02639e-19 J
Energy difference between levels 4-3:E(4)-E(3) = 1.05924e-19 J
Energy difference between levels 5-4:E(5)-E(4) = 4.90275e-20 J
Energy difference between levels 6-5:E(6)-E(5) = 2.66322e-20 J
```

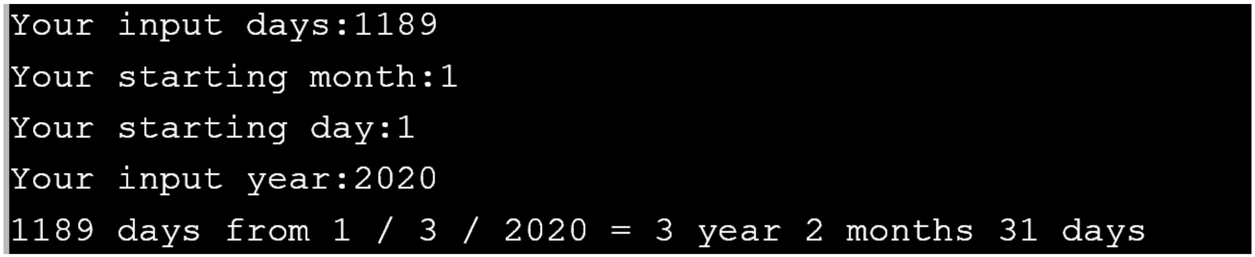
PROBLEM 3:

PART I: CODE



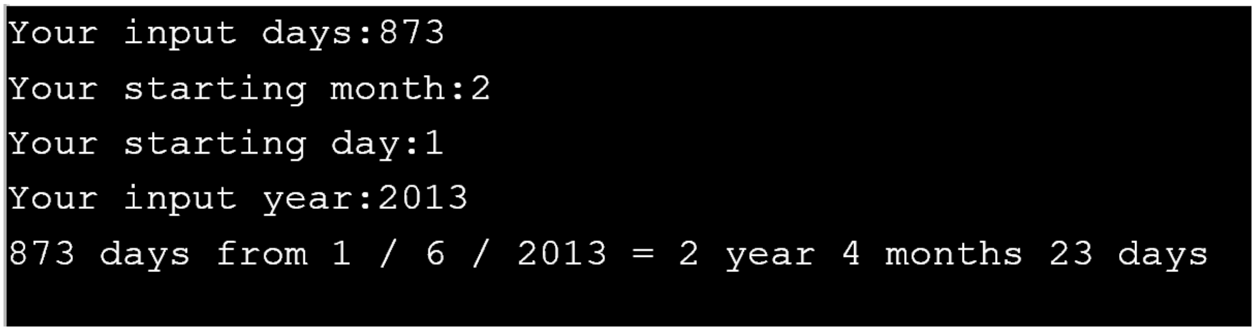
```
#include<stdio.h>
#include<math.h>

//I will create a function which would check if a year is a leap year or not
int leapyear(int year){
  if (year % 400 == 0) {
    return 1;
  }
  else if (year % 100 == 0) {
    return 0;
  }
  else if (year % 4 == 0) {
    return 1;
  }
  else {
    return 0;
  }
}
//I will check the number of days for each month now by a function
int checkmonthdays(int starting_month,int leapyear) {
switch(starting_month) {
  case 1:
  case 3:
  case 5:
  case 7:
  case 8:
  case 10:
  case 12:
  return 31;
  break;
  case 4:
  case 6:
  case 9:
  case 11:
  return 30;
  break;
  case 2:
  if (leapyear == 1) {
    return 29;
  }
  else {
    return 28;
  }
  break;
  }
}

int main() {
//initializing the years, months and days as integers
  int year,month,day;
  int starting_day,input_day,number_days;
  int starting_month;
  int input_year;
  int number_months,number_years;
  number_months = 0;
  number_years = 0;

  printf("Your input days:");
  scanf("%d",&input_day);
  printf("Your starting month:");
  scanf("%d",&starting_month);
  printf("Your starting day:");
  scanf("%d",&starting_day);
  printf("Your input year:");
  scanf("%d",&input_year);

  number_days = input_day;

//Now I'll do the calculations on converting days into years, months and integers
while (number_days > checkmonthdays(starting_month, leapyear(input_year))) {
        number_days -= checkmonthdays(starting_month, leapyear(input_year));
        if (starting_month > 11) {
            starting_month = 1;
        }
        else {
            starting_month += 1;  //check the next month
        }
        number_months += 1;   //keep counter of how many months passed

    }

    number_years = number_months / 12;
    number_months = number_months % 12;

    printf("%d days from %d / %d / %d = %d year %d months %d days", input_day, starting_day, starting_month, input_year, number_years, number_months, number_days);
    return 0;

}
```

OUTPUT: PART 1



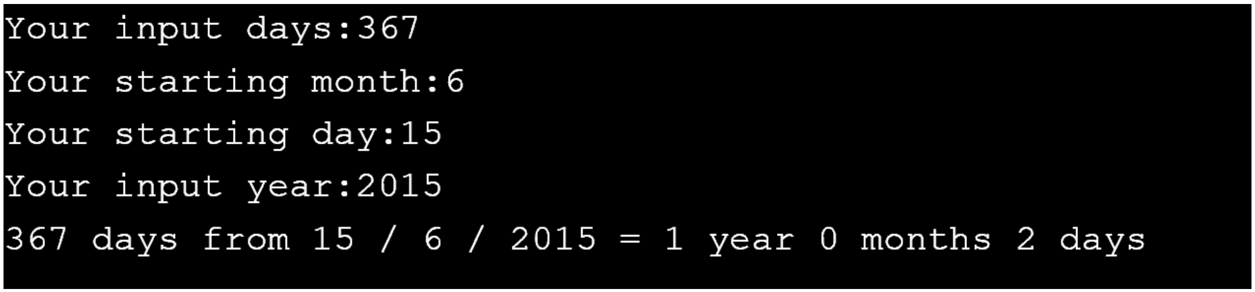
```
Your input days:1189
Your starting month:1
Your starting day:1
Your input year:2020
1189 days from 1 / 3 / 2020 = 3 year 2 months 31 days
```

OUTPUT: PART 2



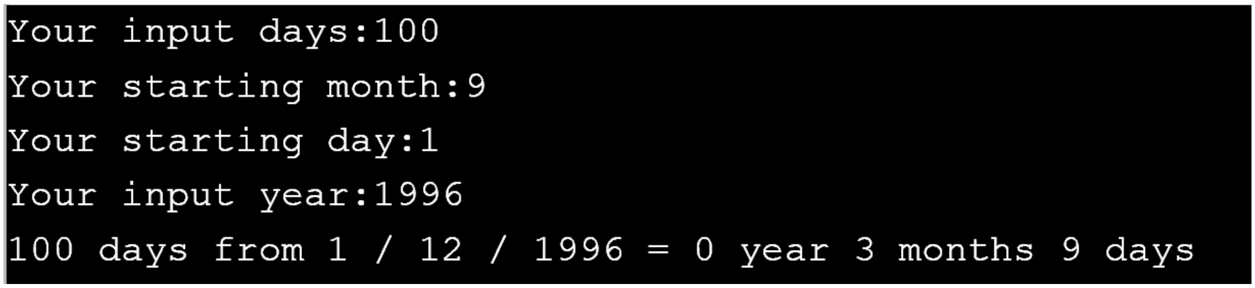
```
Your input days:873
Your starting month:2
Your starting day:1
Your input year:2013
873 days from 1 / 6 / 2013 = 2 year 4 months 23 days
```

OUTPUT: PART 3



```
Your input days:367
Your starting month:6
Your starting day:15
Your input year:2015
367 days from 15 / 6 / 2015 = 1 year 0 months 2 days
```

OUTPUT: PART 4



```
Your input days:100
Your starting month:9
Your starting day:1
Your input year:1996
100 days from 1 / 12 / 1996 = 0 year 3 months 9 days
```