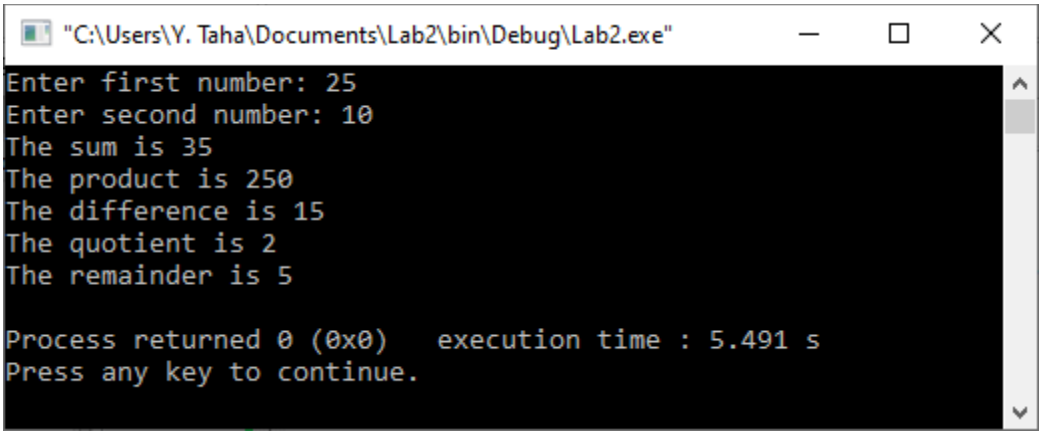
Lab 2

Objectives:

1. Defining **variables,** mathematical operations (+, - , *, / ,%), Precedence, Mathematical functions
2. **Integer division** in C, Increment and Decrement, Casting, Working with float numbers
3. Printing the value of a variable using **printf** - Using **%d, %f, %c**
4. Using **scanf** for reading values from user
5. Examples on operations, printf and scanf

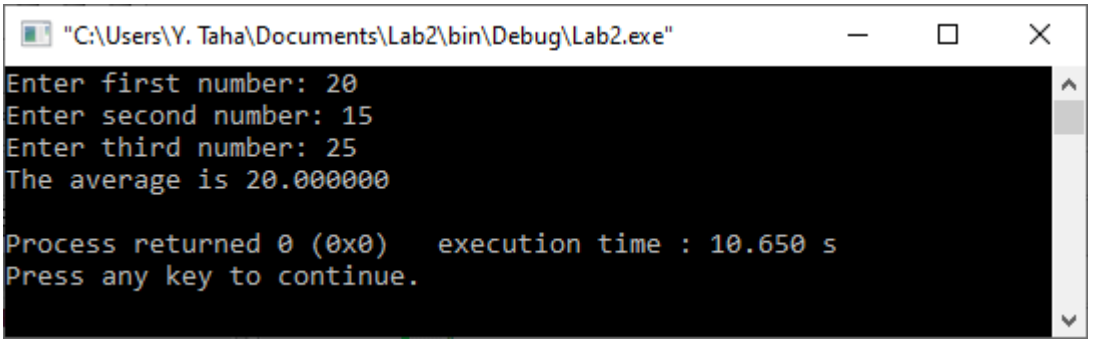
1. Write a program that asks the user to enter two integer numbers and prints their sum, product, difference, quotient, and remainder.

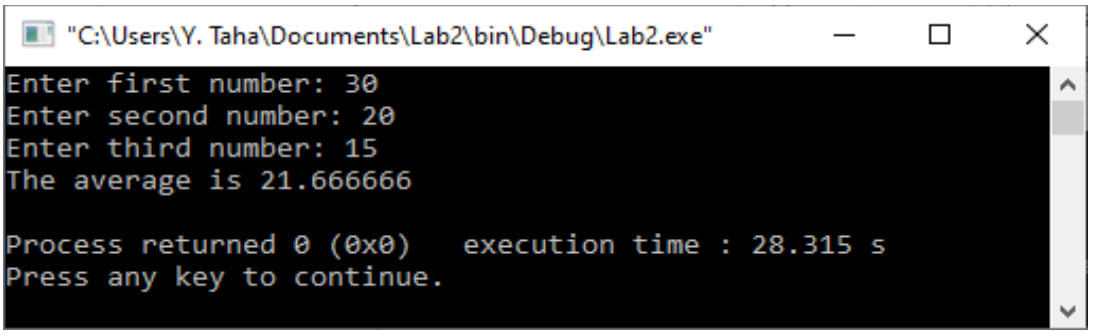
```
#include <stdio.h>
int main ( )
{
int x, y, sum, product, difference, quotient, remainder;
printf( "Enter first number: ");
scanf( "%d", &x);
printf( "Enter second number: ");
scanf( "%d", &y);
sum = x + y;
product = x * y;
difference = x - y;
quotient = x / y;
remainder = x % y;
printf( "The sum is %d\n", sum);
printf( "The product is %d\n", product);
printf( "The difference is %d\n", difference);
printf( "The quotient is %d\n", quotient );
printf( "The remainder is %d\n", remainder);
return 0;
}
```



2. Write a program that reads three integers and prints the average of the integers. (Using several types of casting)

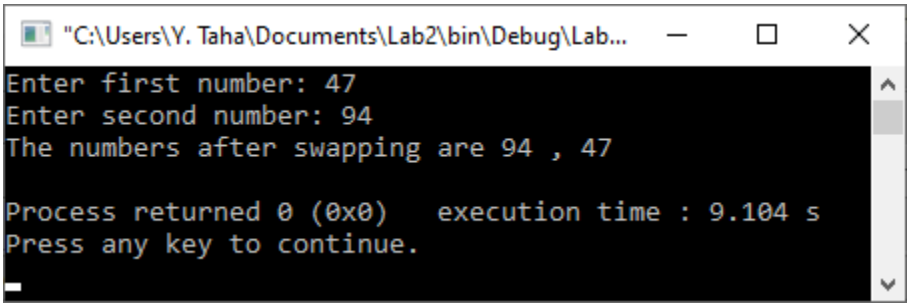
```
#include <stdio.h>
int main ( )
{
int x, y, z;
float average;
printf( "Enter first number: ");
scanf( "%d", &x);
printf( "Enter second number: ");
scanf( "%d", &y);
printf( "Enter third number: ");
scanf( "%d", &z);
average = (x + y + z) / 3.0 ;
printf( "The average is %f\n", average);
return 0;
}
```

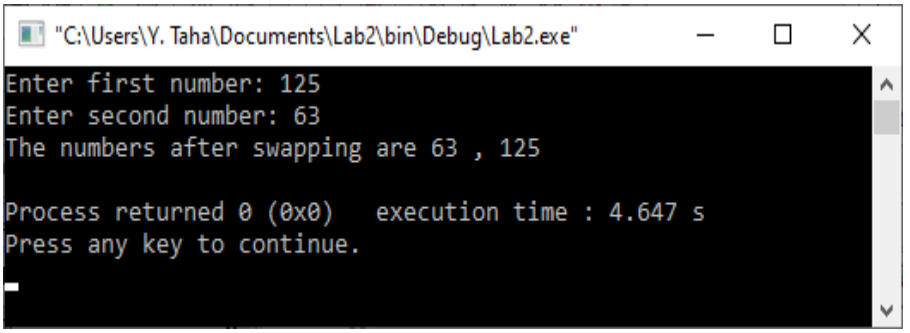


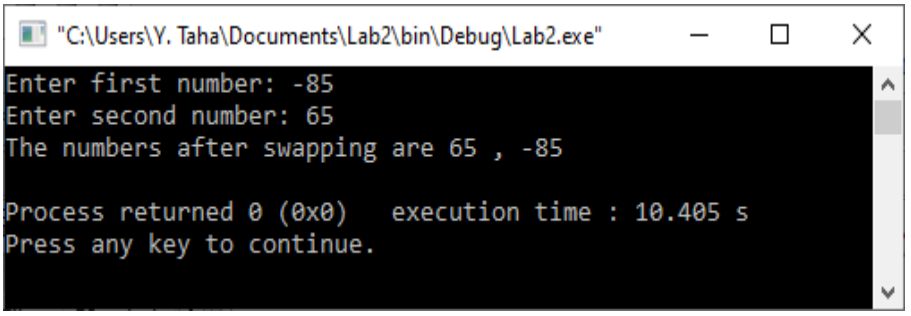


3. Write a program that reads two integers in variables, exchanges them, and prints the values after exchanging

```
#include <stdio.h>
int main ( )
{
int x, y, temp;
printf( "Enter first number: ");
scanf( "%d", &x);
printf( "Enter second number: ");
scanf( "%d", &y);
temp = x; x = y; y = temp;
printf( "The numbers after swapping are %d , %d\n", x,y);
return 0;
}
```

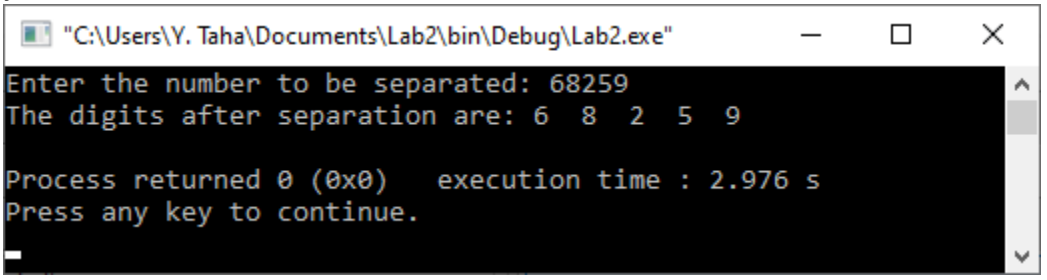


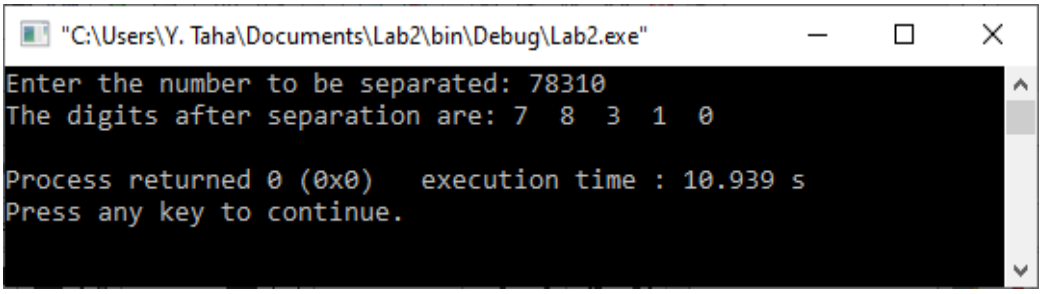




4. Write a program that reads one five-digit number, then separates the number into its individual digits, and prints the digits separated from one another by two spaces each. [Hint: Use combinations of integer division and the remainder operation]. For example, if the user types in 42139, the program should print 4 2 1 3 9 [Hint: Use 5 variables to split the integer, i.e. variable for each digit]

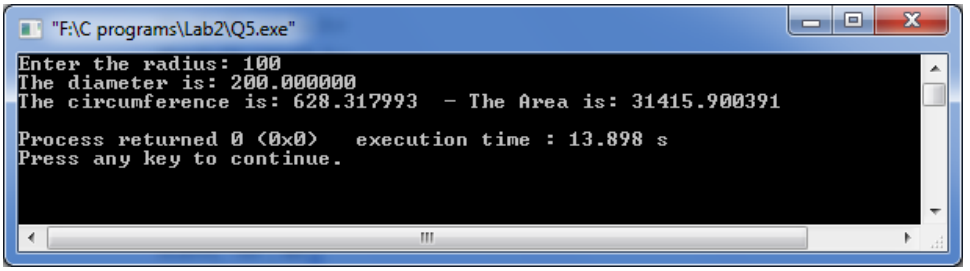
```
#include <stdio.h>
int main ( )
{
long x;
int d1, d2, d3, d4, d5;
printf( "Enter the number to be separated: ");
scanf( "%ld", &x);
d1 = x % 10; x = x /10;
d2 = x % 10; x = x /10;
d3 = x % 10; x = x /10;
d4 = x % 10; x = x /10;
d5 = x;
printf( "The digits after separation are: %d  %d  %d  %d  %d\n", d5,d4,d3,d2,d1);
return 0;
}
```

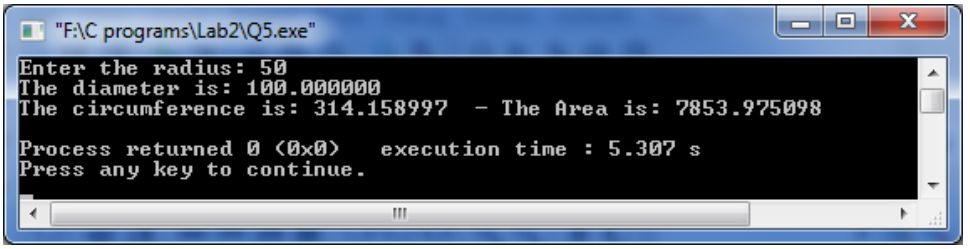


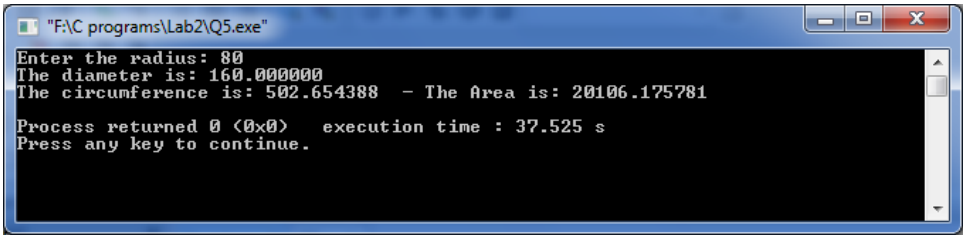


5. Write a program that reads in a float number representing the radius of a circle and then prints the circle's diameter, circumference and area. Use **math.h** library for your calculations. Use the **constant** value 3.14159 for π.

```
#include <stdio.h>
#include <math.h>
#define PI  3.14159
int main ( )
{
float r, cir, area;
printf( "Enter the radius: ");
scanf( "%f", &r);
cir = 2 * PI * r;
area = PI * pow( r , 2);
printf("The diameter is: %f\n", 2*r);
printf( "The circumference is: %f  - The Area is: %f \n", cir, area);
return 0;
}
```







Take home Assignments:

1. John is responsible for planting the street with trees; he can give you the **length** of the street in meters, the **distance** between each two trees in a meter, and the **cost** of planting each tree in dollars. Write a program that should read this information and then print the **number** of trees needed and the **total cost**.

2. Write a program that calculates the **squares, cubes, square root, and exponent (e^x)** of the numbers from 0 to 5 and uses tabs to print the following table of values:

Number	Square	Cube	Root	Exponent
0	0	0	0.0	1.0
..	..	..	..	..
..	..	..	..	..

Please read the following list of common errors:

1. If you open a curly bracket **{then** do not forget to close it at the correct place}
2. Not putting; at the end of every statement, you wrote except for ***int main ()***
3. Your program logic goes right after **the first {and** ends before ***return 0;***
4. Forgetting one or both of the **double quotes (“ ”)** surrounding the format control string in a ***printf*** or ***scanf***
5. Placing variable definitions among executable statements causes syntax errors
6. A calculation in an assignment statement must be on the **right side** of the = operator. It is a compilation error to place a calculation on the **left side** of an assignment operator
7. Forgetting the **%** in a **conversion specification** in the format control string of a ***printf*** or ***scanf***
8. Placing an **escape sequence “format char”** such as ***n*** outside the format control string **“”** of a ***printf*** or ***scanf***
9. Forgetting to include the **expressions** whose values are to be printed in a ***printf*** containing **conversion specifiers**
10. Not providing a **conversion specifier** when one is needed in a ***printf*** format control string to print the value of an **expression**
11. Placing inside the format control string the **comma (,)** that is supposed to separate the format control string from the expressions to be printed
12. Using the incorrect format conversion specifier when reading data with ***scanf***
13. Forgetting to precede a variable in a ***scanf*** statement with an **ampersand (&)** when that variable should, in fact, be preceded by an ampersand
14. Preceding a variable included in a ***printf*** statement with an **ampersand (&)** when, in fact, that variable should not be preceded by an ampersand
15. An attempt to **divide by zero** is normally undefined on computer systems and generally results in a fatal error, i.e., an error that causes the program to terminate immediately without having successfully performed its jobs. Nonfatal errors allow programs to run to completion, often producing incorrect results.
16. If your program won’t run, check the red flag on the right and read the description of the error in the output panel below
17. Be careful with ***scanf***, you cannot leave a **space** or put **\n** at the end of format control string. But the following is correct: scanf("%d%d%d", &x, &y, &z); And you can enter their values by either **leaving space** between each value, or by **pressing “Enter”** after each value

Good Luck