INSTITUTE OF ENGINEERING

Pulchowk Campus, Lalitpur

Subject: C Programming

Lab Report 7

Tittle: **Functions**

Submitted by:

Susheel Thapa 077BCT090

Submitted to:

Department of Electronics
and
Computer Engineering

Checked by

Content of Lab Report:

Background Information

What is C Programming?

C programming is a general-purpose, procedural, imperative computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system. C is the most widely used computer language.

Why to Learn C Programming?

- Easy to learn
- Structured language
- It produces efficient programs
- It can handle low-level activities
- It can be compiled on a variety of computer platforms

Editor

Here, I have used Visual Studio Code as my editor. You can download the editor from Download Visual Studio Code - Mac, Linux, Windows . Select your operating system and download it.

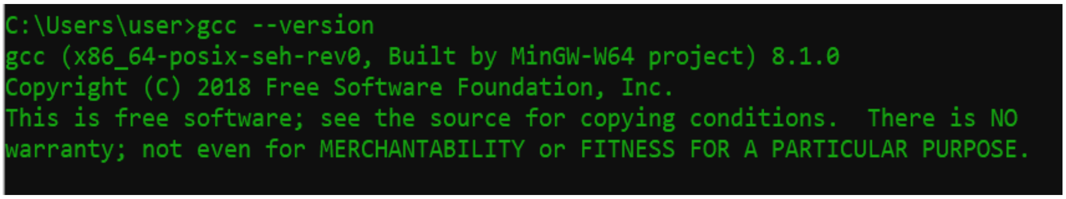
Compiler

Here, I have used **gcc** as my compiler provided by MinGWw64. You can download it via Download MinGW-w64 - for 32 and 64 bit Windows from SourceForge.net. Your download will start automatically. Run the downloaded .exe file. After, you have installed MinGW-w64, you need to configure it.

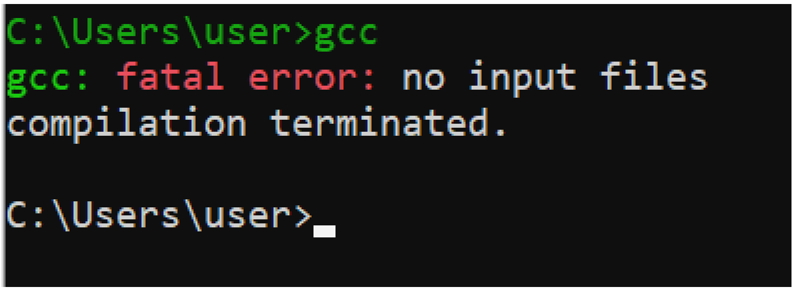
- In the Windows search bar, type 'settings' to open your Windows Settings.
- Search for Edit environment variables for your account.
- Choose the Path variable and then select Edit.
- Select New and add the Mingw-w64 destination folder path to the system path. The exact path depends on which version of Mingw-w64 you have installed and where you installed it. If you used the settings above to install Mingw-w64, then add this to the path: **C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin.**
- Select OK to save the updated PATH. You will need to reopen any console windows for the new PATH location to be available.

Check your installation

Open command prompt or power shell and type:



```
C:\Users\user>gcc --version
gcc (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```



```
C:\Users\user>gcc
gcc: fatal error: no input files
compilation terminated.

C:\Users\user>_
```

If you get similar result, you are good to go.

C User Defined Function

A function is a block of code that performs a specific task.

C allows you to define functions according to your need. These functions are known as user-defined functions. For example:

Suppose, you need to create a circle and color it depending upon the radius and color. You can create two functions to solve this problem:

createCircle() function

color() function

Function prototype

A function prototype is simply the declaration of a function that specifies function's name, parameters and return type. It doesn't contain function body.

A function prototype gives information to the compiler that the function may later be used in the program.

Syntax of function prototype

returnType functionName(type1 argument1, type2 argument2, ...);

The function prototype is not needed if the user-defined function is defined before the main() function.

Calling a function

Control of the program is transferred to the user-defined function by calling it.

Syntax of function call

```
functionName(argument1, argument2, ...);
```

Function definition

Function definition contains the block of code to perform a specific task. In our example, adding two numbers and returning it.

Syntax of function definition

```
returnType functionName(type1 argument1, type2 argument2, ...){    //body of the function}
```

When a function is called, the control of the program is transferred to the function definition. And, the compiler starts executing the codes inside the body of a function.

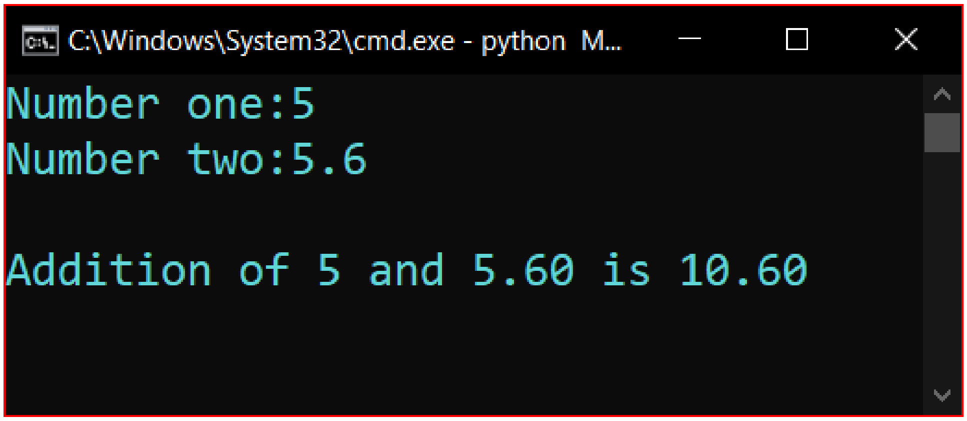
1.Write a program to create a function float add(int ,float). The task of this function is to calculate the sum of passed value and return it to the calling function. Call this function from main() and display result

Source Code:

```
/*Header Files*/
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

/*Function defination*/
float add(int, float);

int main()
```

```
{
    /*Variable Declaration*/
    int number_one;
    float number_two, sum;

    system("cls"); /*Clear the Screen*/

    /*Taking input*/
    printf("Number one:");
    scanf("%d", &number_one);
    printf("Number two:");
    scanf("%f", &number_two);

    /*Function call*/
    sum = add(number_one, number_two);

    /*Desplay the result*/
    printf("\nAddition of %d and %.2f is %.2f", number_one, number_two, sum);

    getch(); /*Waits till a character is pressed*/
    return 0;
}

/*Function body*/
float add(int a, float b)
{
    return (float)a + b;
}
```

Output:



```
Number one:5
Number two:5.6

Addition of 5 and 5.60 is 10.60
```

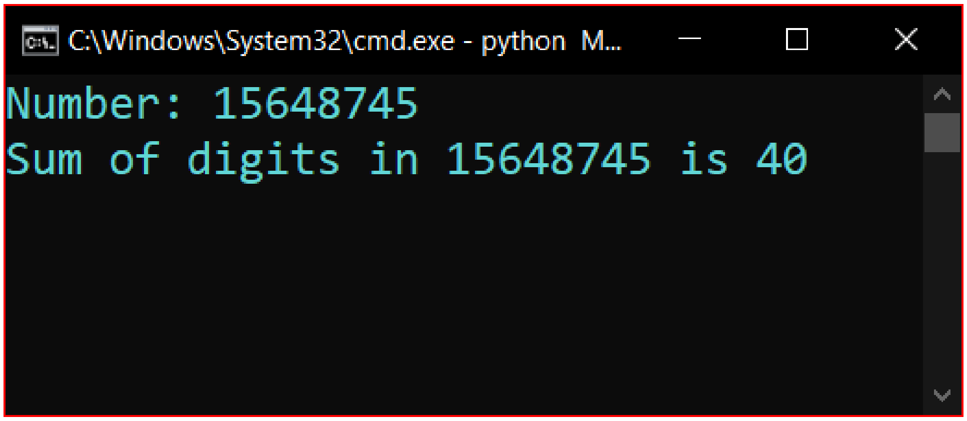
2.Write a program to create a function void sumOfDigits(int);. This function must calculate the sum of digits in the given number and displays the sum.

Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

/*Function Defination*/
int sumOfDigits(int);

int main()
{
   /*Varaible Declaration*/
   int number, sum;

   system("cls");/*Clear the screen*/

   /*Taking input*/
```

```
    printf("Number: ");
    scanf("%d", &number);

    sum = sumOfDigits(number);

    printf("Sum of digits in %d is %d", number, sum);

    getch();
    return 0;
}

/*Function Body*/
int sumOfDigits(int a)
{
    if (a < 0)
    {
        return 0;
    }
    else
    {
        while (a != 0)
        {

            int rem = a % 10;
            a = a / 10;
            return rem + sumOfDigits(a);
        }
    }
}
```

Output:



```
Number: 15648745
Sum of digits in 15648745 is 40
```

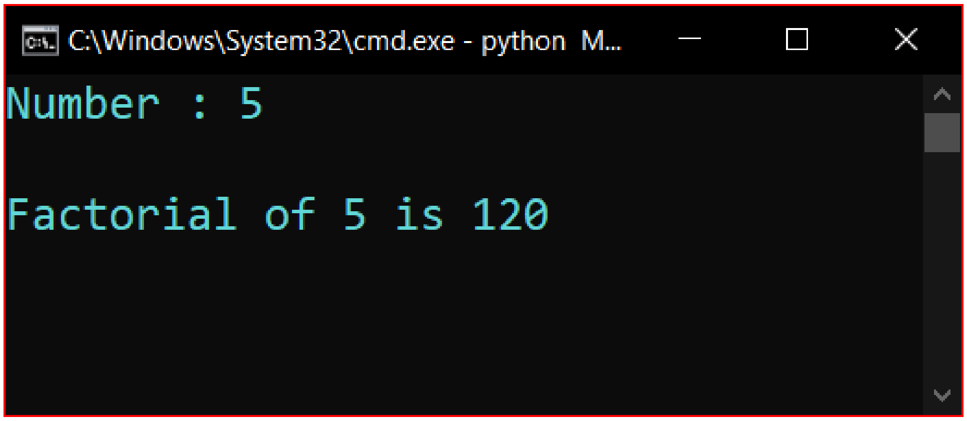
3.Write a program to read a non negative integer in main(). Pass this integer to a function fact() having return type unsigned integer. The function calculate the factorial of the received number and return to main() to display it.

Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

/*Function definition and body*/
unsigned int fact(unsigned int a)
{
    if (a == 0)
    {
        return 1;
    }
    return a * fact(a - 1);
}
```

```
int main()
{
    /*Variable Declaration*/
    unsigned number, factorial;

    system("cls");/*Clear the screen*/

    /*Taking input*/
    printf("Number : ");
    scanf("%u", &number);

    factorial = fact(number);/*Function call*/

    printf("\nFactorial of %u is %u", number, factorial);/*Meaningful result*/

    getch();
    return 0;
}
```

Output:



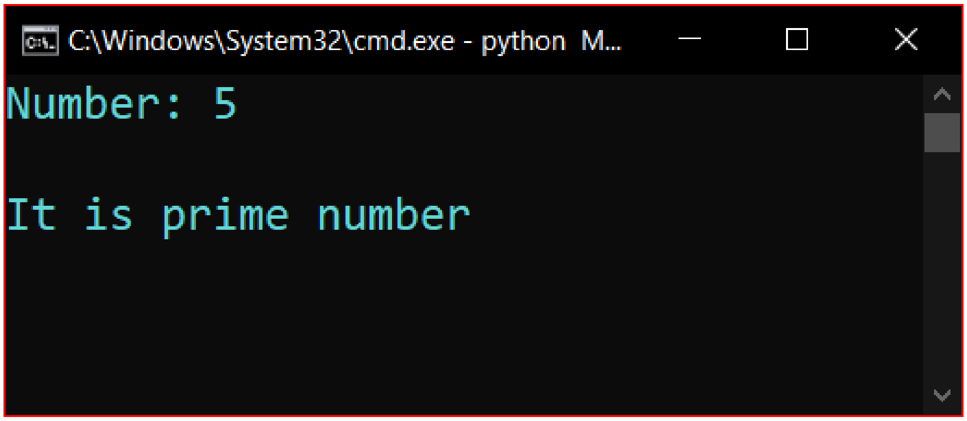
```
Number : 5

Factorial of 5 is 120
```

4.Write a program to check a function void check_prime(): The Task of this program is to read a number and check whether the number is prime or not and display the appropriate message. Be sure that a real number cannot be either prime or composite.What about negative number

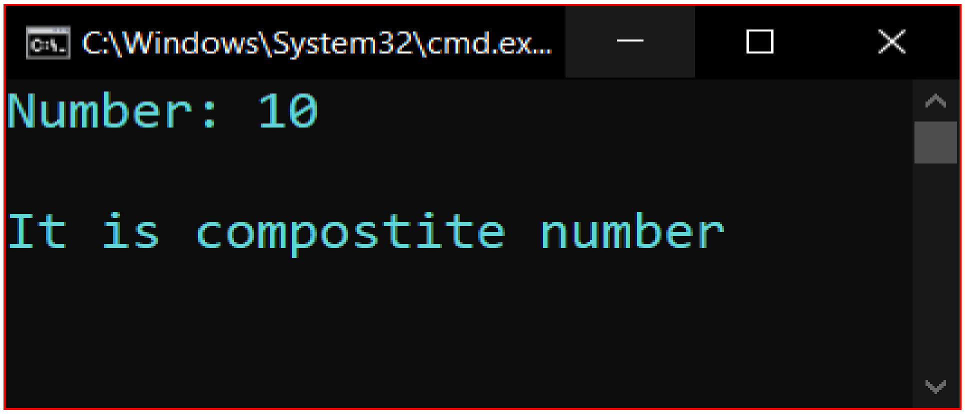
Source Code:

```
#include <stdio.h>

#include <conio.h>

#include <stdlib.h>


/*Function*/

void check_prime(int a)

{

    for (int i = 2; i < a; i++)

    {

        if (a % i == 0)

        {

            printf("\nIt is compostite number");

            exit(0);

        }

    }

    printf("\nIt is prime number");

}


int main()

{

    /*Variable Declaration*/

    int number;
    system("cls");/*clear the screen*/

    /*Taking input*/
    printf("Number: ");
    scanf("%d", &number);

    check_prime(number);/*Function call*/

    getch();
    return 0;
}
```

Output:



```
Number: 5

It is prime number
```



```
Number: 10

It is compostite number
```

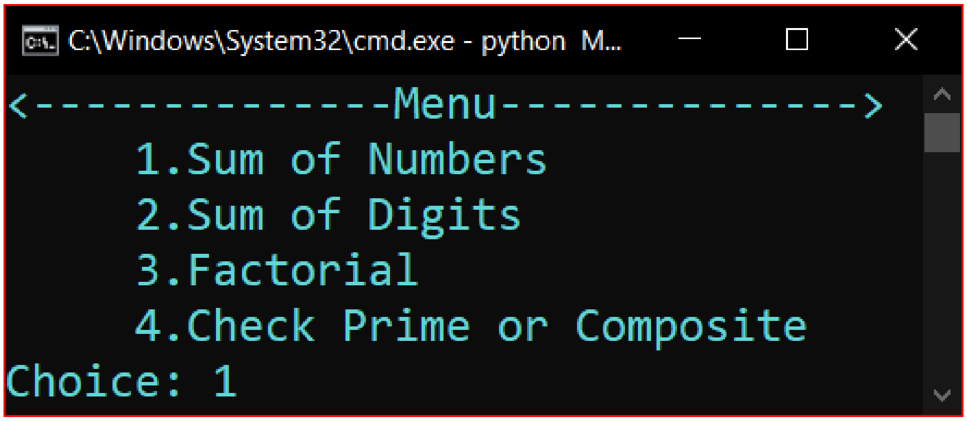
5.Combine Question 1 2 3 4 using switch statement. For this display a menu on the screen to prompt user whether he wants to sum two number or sumof digits of an integer or calculate factorial of an integer or to know whether number is prime or not

Source Code:

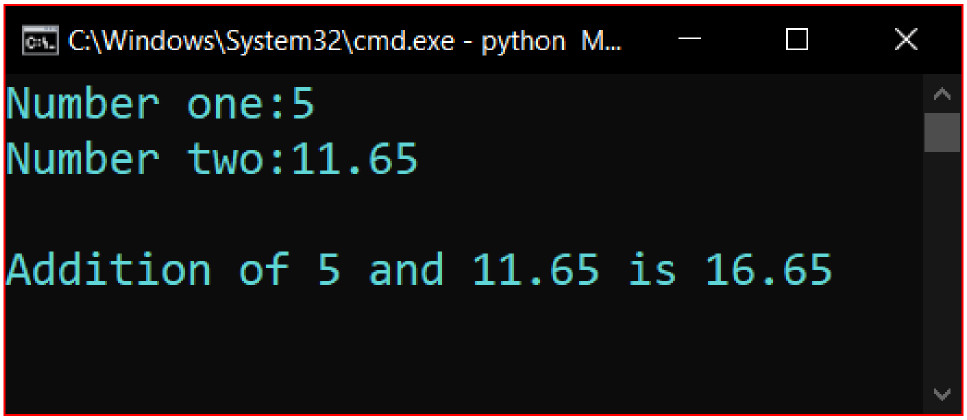
```
#include <stdio.h>

#include <conio.h>

#include <stdlib.h>



/*Function taken from Program 1,2,3,4*/

float add(int a, float b)

{

    return (float)a + b;

}


int sumOfDigits(int a)

{

    if (a < 0)

    {

        return 0;

    }

    else

    {

        while (a != 0)

        {


            int rem = a % 10;
            a = a / 10;
            return rem + sumOfDigits(a);
        }
    }
}

unsigned int fact(unsigned int a)
{
    if (a == 0)
    {
        return 1;
    }
    return a * fact(a - 1);
}

void check_prime(int a)
{
    for (int i = 2; i < a; i++)
    {
        if (a % i == 0)
        {
            printf("\nIt is compostite number");
            exit(0);
        }
    }
    printf("\nIt is prime number");
}
```

```
/*Function to display Option*/
void menu()
{
    printf("<--------------Menu-------------->\n");
    printf("%5s%s\n", "", "1.Sum of Numbers");
    printf("%5s%s\n", "", "2.Sum of Digits");
    printf("%5s%s\n", "", "3.Factorial");
    printf("%5s%s\n", "", "4.Check Prime or Composite");
}

int main()
{
    /*Variable Declaration*/
    int one_number_one;
    float one_number_two, one_sum;

    int two_number, two_sum;

    unsigned three_number, three_factorial;

    int four_number;

    int choice;

    /*Clear the screen*/
    system("cls");
    menu();/*Function call*/



    /*Taking input*/
    printf("Choice: ");
    scanf("%d", &choice);

    /*Simple switch case and in every case code copy from Program 1,2,3,4*/
    switch (choice)
    {
    case 1:

        system("cls"); /*Clear the Screen*/

        /*Taking input*/
        printf("Number one:");
        scanf("%d", &one_number_one);
        printf("Number two:");
        scanf("%f", &one_number_two);

        /*Function call*/
        one_sum = add(one_number_one, one_number_two);

        /*Desplay the result*/
        printf("\nAddition of %d and %.2f is %.2f", one_number_one, one_number_two, one_sum);
        break;

    case 2:

        system("cls");

        printf("Number: ");
        scanf("%d", &two_number);

        two_sum = sumOfDigits(two_number);
        printf("Sum of digits in %d is %d", two_number, two_sum);

    case 3:

        system("cls");

        printf("Number : ");
        scanf("%u", &three_number);

        three_factorial = fact(three_number);

        printf("\nFactorial of %u is %u", three_number, three_factorial);

        break;

    case 4:
        system("cls");

        printf("Number: ");
        scanf("%d", &four_number);

        check_prime(four_number);

    default:
        printf("You didn't choose right option.");
        break;
    }

    getch();
    return 0;
}
```

Output:



```
<---------------Menu--------------->
     1.Sum of Numbers
     2.Sum of Digits
     3.Factorial
     4.Check Prime or Composite
Choice: 1
```



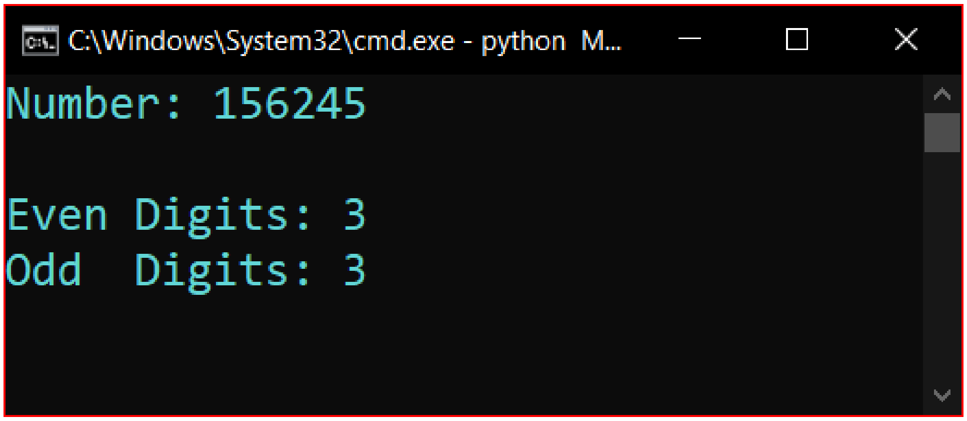
```
Number one:5
Number two:11.65

Addition of 5 and 11.65 is 16.65
```

6.Write a program to read an unsigned integer in main() pass it to function.(void countsDigits(int*,int *);). This function counts the number of odd and even digits in it. Display the count from main. Use concept of passing argument by refrence

Source Code:

```
#include <stdio.h>

#include <conio.h>

#include <stdlib.h>


/*Function definition*/

void countsDigits(int *, int *);


int main()

{

    /*Variable declaration*/

    int digits, even_count = 0, odd_count = 0;


    system("cls");/*Clear the screen*/


    /*Taking input*/

    printf("Number: ");

    scanf("%d", &digits);
    even_count = digits;

    /*Function call*/
    countsDigits(&even_count, &odd_count);

    /*Meaningful result*/
    printf("\nEven Digits: %d\n", even_count);
    printf("Odd  Digits: %d", odd_count);

    getch();
    return 0;
}

/*Function Body*/
void countsDigits(int *e, int *o)
{
    int number = *e;
    *e = 0;
    int even = 0, odd = 0;
    while (number != 0)
    {
        if (number % 2 == 0)
        {
            even++;
        }
        else if (number % 2 == 1)
        {
            odd++;
        }
        number /= 10;
    }
    *e = even;
    *o = odd;
}
```

Output:



```
Number: 156245

Even Digits: 3
Odd  Digits: 3
```

7.Write a progrma to create a function: int findLowest(int,int ,int); and int findHighest(int,int,int); The task of findLowest() is to find the lowest of three integers and return an integers to the calling functions. Similarly, the task of findHighest() is to find the highest of three integers and return an integers to the calling function. Call these functions in main() giving appropriate arguments.

Note: Use conditional operator (test expression? expression1: expression2) to find highest and lowest

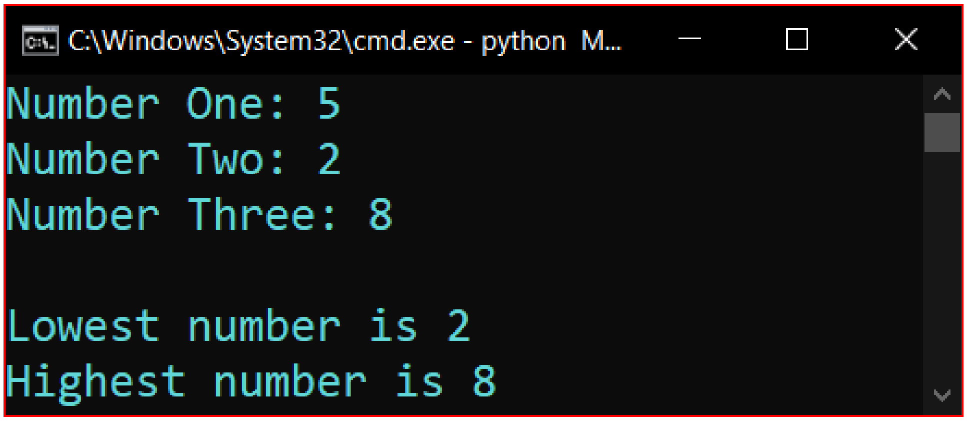
Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

/*Function definition*/
```

```
int findLowest(int, int, int);
int findHighest(int, int, int);

int main()
{
    /*Variable Declaration*/
    int number_one, number_two, number_three;

    system("cls");/*Clear the screen*/

    /*Taking Input*/
    printf("Number One: ");
    scanf("%d", &number_one);
    printf("Number Two: ");
    scanf("%d", &number_two);
    printf("Number Three: ");
    scanf("%d", &number_three);

    /*Function call and printing*/
    printf("\nLowest number is %d", findLowest(number_one, number_two,
number_three));
    printf("\nHighest number is %d", findHighest(number_one, number_two,
number_three));

    getch();
    return 0;
}
```

```
/*Function body*/
int findLowest(int a, int b, int c)
{
    int low;
    (a < b && a < c) ? ((low = a)) : ((b < c) ? ((low = b)) : ((low = c)));
    return low;
}
int findHighest(int a, int b, int c)
{
    int high;
    (a > b && a > c) ? ((high = a)) : ((b > c) ? ((high = b)) : ((high = c)));
    return high;
}
```

Output:



```
Number One: 5
Number Two: 2
Number Three: 8

Lowest number is 2
Highest number is 8
```

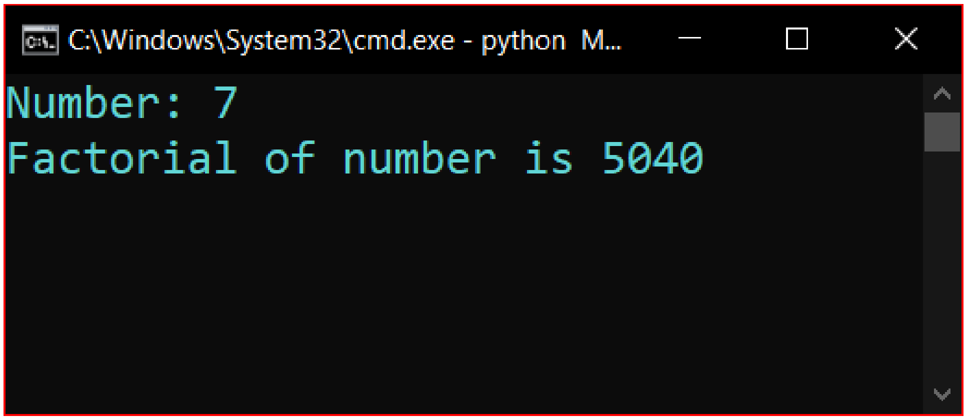
8.1Factorial of n by recursive function

Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
```

```
/*Fucntion definition and declaration*/
int factorial(int n)
{
    if (n == 0)
    {
        return 1;
    }
    else
    {
        return n * factorial(n - 1);
    }
}
int main()
{
    /*Variable Declaration*/
    int number;

    system("cls");/*Clear the screen*/

    /*Taking input from user*/
    printf("Number: ");
    scanf("%d", &number);

    /*Printing result*/
    (number < 0) ? ((printf("Invalid number."))) : (printf("Factorial of number is %d", factorial(number)));
    getch();
    return 0;
}
```

Output:



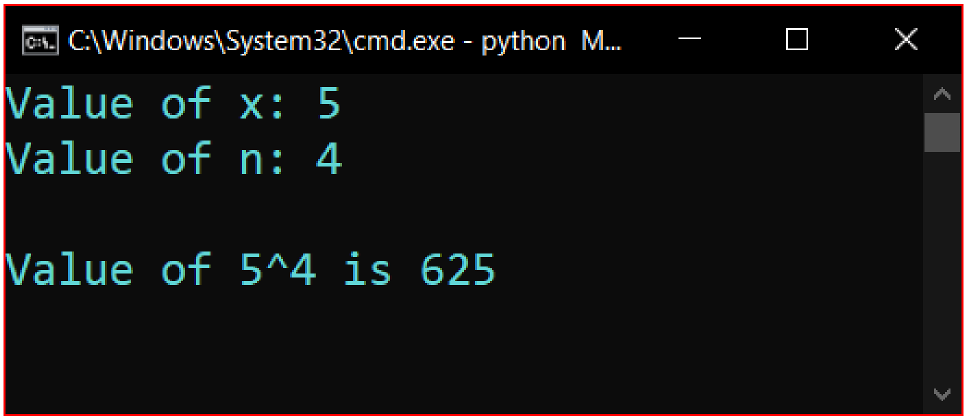
```
Number: 7
Factorial of number is 5040
```

8.2Compute x^n

Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

/*Fucnction definition and declaration*/
int powerOfBase(int base, int power)
{
    if (power == 0)
    {
        return 1;
    }
    else
    {
        return base * powerOfBase(base, power - 1);
    }
}

int main()
{
    /*Variable declaration*/
    int x, n;

    system("cls");/*Clear the screen*/

    /*Taking user input*/
    printf("Value of x: ");
    scanf("%d", &x);
    printf("Value of n: ");
    scanf("%d", &n);

    /*Meaningful Output*/
    printf("\nValue of %d^%d is %d", x, n, powerOfBase(x, n));

    getch();
    return 0;
}
```

Output:



```
Value of x: 5
Value of n: 4

Value of 5^4 is 625
```

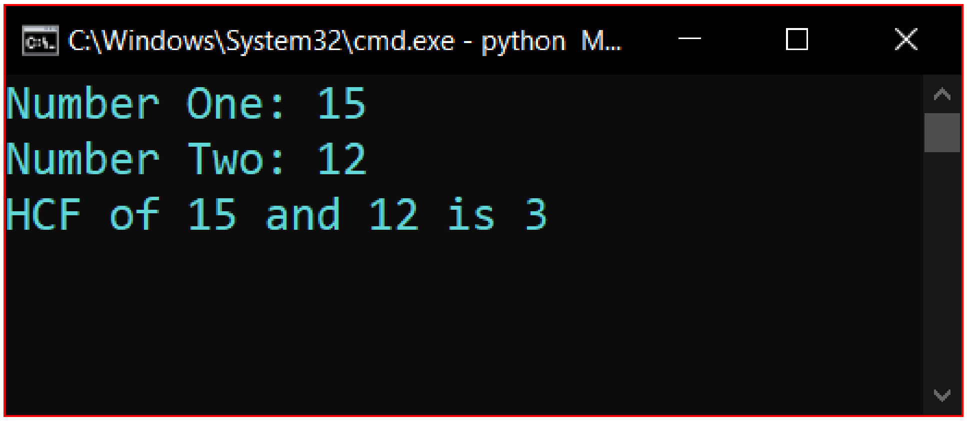
8.3HCF of number

Source Code:

```
#include <stdio.h>

#include <conio.h>

#include <stdlib.h>


/*Function defination and declaration*/

int HCF(int a, int b)

{


   /*Below if just swap the value of a and b.For more clarity, once look Swap_two_numbers.c*/

   if (a < b)

   {

      b = a + b - (b = a);

   }


   /*Termination condition of recursion*/

   if (b == 0)
```

```
    {
        return a;
    }

    /*Declaration of variable*/
    int quotient;
    int remainder;

    /*Explained Above*/
    quotient = a / b;
    remainder = a - quotient * b;

    /*Calling to it self*/
    HCF(b, remainder);
}


int main()
{
    /*Variable declaration*/
    int a, b;

    system("cls");/*Clear the screen*/

    /*Taking input*/
    printf("Number One: ");
    scanf("%d", &a);
    printf("Number Two: ");
    scanf("%d", &b);

    /*Meaningful result*/
    printf("HCF of %d and %d is %d", a, b, HCF(a, b));

    getch();
    return 0;
}
```

Output:



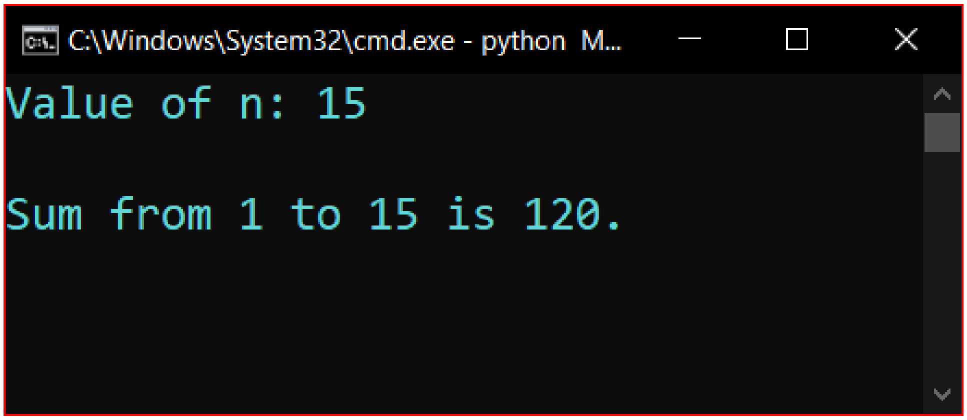
```
Number One: 15
Number Two: 12
HCF of 15 and 12 is 3
```

8.4Sum from 1 to n

Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

/*Fucntion declaration*/
int sum(int n)
{
    if (n == 0)
    {
        return 0;
    }
    else
    {
        return n + sum(n - 1);
    }
}
int main()
{
    /*Varaible declaration*/
    int n;

    system("cls");/*Clear the screen*/

    /*Taking user input*/
    printf("Value of n: ");
    scanf("%d", &n);

    /*Meaningful output*/
    printf("\nSum from 1 to %d is %d.", n, sum(n));

    getch();
    return 0;
}
```

Output:



```
Value of n: 15

Sum from 1 to 15 is 120.
```

9.Write a program using recursive function to compute series 1^2-2^2 +3^2 ....(-1)^(n+1).n^2

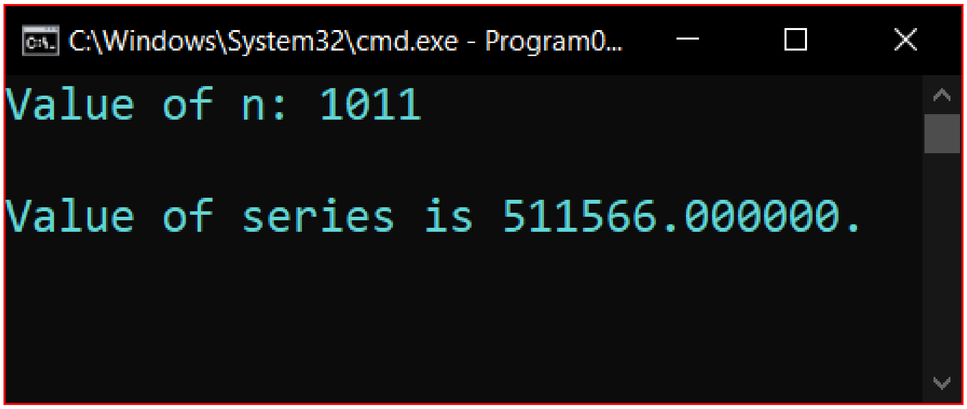
You cannot use pow function

Source Code:

```
#include <stdio.h>

#include <conio.h>

#include <stdlib.h>


/*Function declaration*/

float series(int n)

{

   if (n == 1)

   {

      return 1;

   }

   else

   {

      return (n % 2 == 1) ? (n * n + series(n - 1)) : (-1 * n * n + series(n - 1));

   }

}
```

```
int main()
{
    /*Variable declaration*/
    int n;

    system("cls");/*Clear the screen*/

    /*Taking user input*/
    printf("Value of n: ");
    scanf("%d", &n);

    /*Output*/
    printf("\nValue of series is %f.", series(n));

    getch();
    return 0;
}
```

Output:



```
Value of n: 1011

Value of series is 511566.000000.
```

Analysis:

Here we learn about declaring function in C language how to user it. Along with declaration of function we also came to know the working nature of function. We also came how does pass by value and pass by reference works.

Conclusion

Here we learn to work with function.