BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ.)
CS F111 Computer Programming
LAB SESSION #6
(Functions and Storage Classes)

Create a directory "**lab6**" inside "**myprogs**" directory for all your programs in this week's lab. We will write programs related to functions and arrays.

Functions in C:

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. Functions are used to perform certain actions, and they are important for reusing code: Define the code once, and use it many times. We have already studied how C programs can be modularized using functions and we have also seen how variables are passed into functions using pass by value. Let us now write a few programs:

1. An Armstrong number is an n-digit number such that the sum of its digits raised to the power n is the number itself. For example, the number 153 has three digits, and $1^3+5^3+3^3 = 153$. Therefore 153 is an Armstrong number. Write a C program that accepts two numbers and finds all Armstrong numbers in that range. You should create a function IsArmstrong() for checking whether a number is an Armstrong number or not, and call it repeatedly from within main().

2. Without running the following program, predict its output. Now run the code and match your output. If it's not matching, track down each of the variables after each statement.

```
#include <stdio.h>
int x = -1, y = -2, z = -3;

void f1 (int x, int z)
{
      y = ++x + --z + (y+=2);
      printf("%d %d %d\n",x,y,z);
}

int main()
{
      int z = 3;
      x += y += ++z;
      printf("%d %d %d\n", x, y, z);

      f1(x,z);

      {     /* new block begins */
            float y = 4.0;
            int x = 0;
            x += z = 5 * y;
            printf("%d %.2f %d\n", x, y, z);
      }     /* end of block */

      printf("%d %d %d\n", x, y, z);
}
```

Notice that a block is a piece of code enclosed within { and }and it introduces its own separate scope. Every function is also a block, and hence has its own scope – called the ***function scope.***

3. What would be the output of each of the following pieces of code/functions? First attempt answering the question yourself, and then try and check your answer by compiling the programs.

a.
```
#include<stdio.h>
void func(int a,int b);
int main(void)
{  int x;
   x=func(2,3);
   return 0;
}
void func(int a,int b)
{  int s;
   s=a+b;
   return;
}
```

b.
```
#include<stdio.h>
int diff(int x,int y)
{  return x-y; }
int sum(int x,int y)
{  return x+y; }
int main(void)
{
   int a=20, b=5, c=2, d=6;
   printf("%d\t", a + diff(d,c));
   printf("%d\n", diff(a,sum(diff(b,c),d)));
   return 0;
}
```

c.
```
#include <stdio.h>
void func(void);
int main(void)
{
   int i=5;
   for(i=i+1; i<8; i++)
         func();

}

void func(void)
{
   int j;
   for(j=1; j<3; j++)
         printf("%d\t",++j);
}
```

4. Write a function in C that takes a number **n** as the input and compute its square, its cube and the area of the circle with **n** as the radius. Your function should display these values.

5. Write a function that takes an number **n** as input and computes the sum of the following series: 1!/1 + 2!/2 + 3!/3 + 4!/4 + 5!/5 + ... + n!/n. The function should then return the computed sum. The main function which is calling the above function, should print the sum of the above series. Implement the function in both **iterative** and **recursive** way.

6. Write a function that takes a number **n** as input and checks if that number is a prime number or not. The function should return 1 if it is a prime, 0 otherwise.

7. A Fibonacci sequence is a sequence in which each number is the sum of the two preceding ones. The sequence starts from 0 and 1, the first few values are: 0, 1, 1, 2, 3, 5, 8, 13 .... Given an input n, implement a function that prints the n-th Fibonacci number. Implement the function in both **iterative** and **recursive** way.

Additional Practice Exercises

8. Write a program in C to convert decimal number to binary number using the function. For eg. On 65 as an input, the function should return 1000001. [Hint: Think about the appropriate datatypes of the function input and return parameters]

9. Consider the following code:

```
#include <stdio.h>

void swap (int a, int b)
{
   int temp;
   temp = a;
   a = b;
   b = temp;
}
int main(void)
{

    int i=5, j=2;
    swap(i,j);
    printf("%d %d", i, j);
}
```

What is the expected output of the following program? Do you see any problem?

10. Write a C program to check whether a given number can be represented as the sum of two prime numbers. This program takes a positive integer from the user and checks whether that number can be expressed as the sum of two prime numbers. If the number can be expressed as the sum of two prime numbers, the output shows the combination of the prime numbers.

For this you need to create a separate function **checkPrime()** which takes an integer and returns 0 or 1 depending whether a number is prime or not. The main function uses this function to achieve the final goal.

11. What is the effect of calling show(4)?

```
int show(int x) {
   printf("%d %d\n", x, x*x);
   return x*x;
   printf("%d %d\n", x, x*x*x);
   return x*x*x;
}
```

12. What values are printed out by the following C program?

```
#include <stdio.h>

int confusion(int x, int y) {
   x = 2*x + y;
   return x;
}
int main(void) {
    int x = 2, y = 5;
    y = confusion(y, x);
    x = confusion(y, x);
    printf("%d %d\n", x, y);
    return 0;
}
```

13. Write a function that takes a positive integer as input and returns the leading digit in its decimal representation. For example, the leading digit of 234567 is 2.
14. Write a function that takes a number n as input. Calculate its factorial through a **recursive function**.