JAIN
DEEMED-TO-BE UNIVERSITY

# Programming in C

## Lab File

Experiments No.: 29 to 33

Subject Code: 16BCA1C05L
Class: I Year I Semester (BCA)

Prepared By
Suman Garai
JU2020BCAS19059

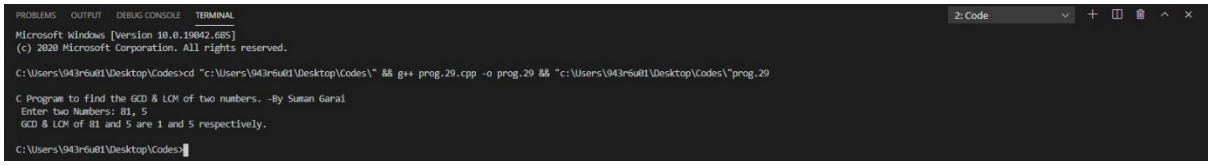
**Experiment 29:**

*C program using functions to find GCD and LCM of two numbers.*

**Code:**

```
#include <stdio.h>
void comp(int, int);
void work(int, int);
main()
{
    int n1, n2, a;
    printf("\nC Program to find the GCD & LCM of two
    numbers. -By Suman Garai");
    printf("\n Enter two Numbers: ");
    scanf("%d, %d", &n1, &n2);
    printf(" GCD & LCM of %d and %d ", n1, n2);
    comp(n1, n2);
}
void comp(int n1, int n2)
{
    int num, den;
    if (n1 > n2) { num = n1; den = n2; }
    else { num = n2; den = n1; }
    work(num, den);
}
void work(int num, int den)
{
    int rem, gcd, lcm;
    rem = num % den;
    while (rem!=0)
    {
        num = den;
        den = rem;
        rem = num % den;
    }
    gcd = den; lcm = num * den / gcd;
    printf("\b are %d and %d respectively.\n", gcd,
lcm);
}
```

# Output:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
2: Code
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\943r6u01\Desktop\Codes>cd "c:\Users\943r6u01\Desktop\Codes\" && g++ prog.29.cpp -o prog.29 && "c:\Users\943r6u01\Desktop\Codes\"prog.29
C Program to find the GCD & LCM of two numbers. -By Suman Garai
Enter two Numbers: 81, 5
GCD & LCM of 81 and 5 are 1 and 5 respectively.
C:\Users\943r6u01\Desktop\Codes>

**Experiments 30 & 31:**

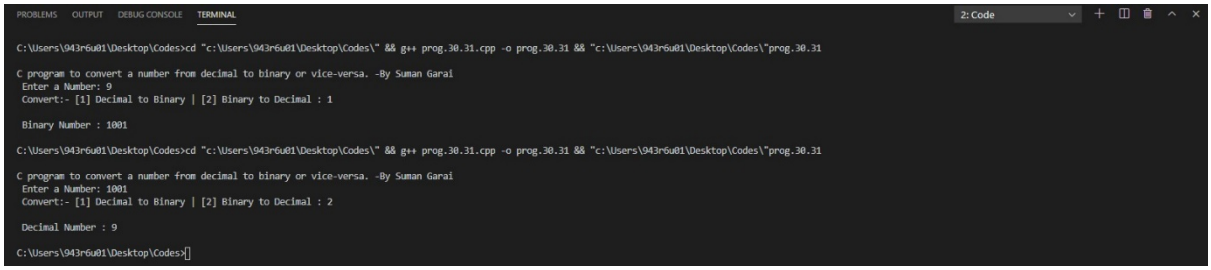
*C program using functions to convert a decimal number to its binary equivalent and vice-versa.*

**Code:**

```
#include <stdio.h>
#include <math.h>
void dec(int);
void bin(int);
main()
{
    int num, o;
    printf("\nC program to convert a number from
    decimal to binary or vice-versa. -By Suman
    Garai");
    printf("\n Enter a Number: ");
    scanf("%d", &num);
    Label:printf(" Convert:- [1] Decimal to Binary |
    [2] Binary to Decimal : ");
    scanf("%d",&o);
    switch (o)
    {
        case 1: dec(num); break;
        case 2: bin(num); break;
       default : printf("Invalid Entry. Please
       Enter Value Correctly !"); goto Label; break;
    }
}
void dec(int num)
{
    int r, bin = 0, k = 1;
    do
    {
        r = num % 2;
        num = num / 2;
        bin = bin + r * k;
        k = k * 10;
    } while (num!=0);
```

```
    printf("\n Binary Number : %d\n", bin);
}
void bin(int num)
{
    int dec = 0, r, k, i = 0;
    do
    {
        r = num % 10;
        num = num / 10;
        dec = dec + r * pow(2, i);
        i++;
    } while (num!=0);
    printf("\n Decimal Number : %d\n", dec);
}
```

## Output:



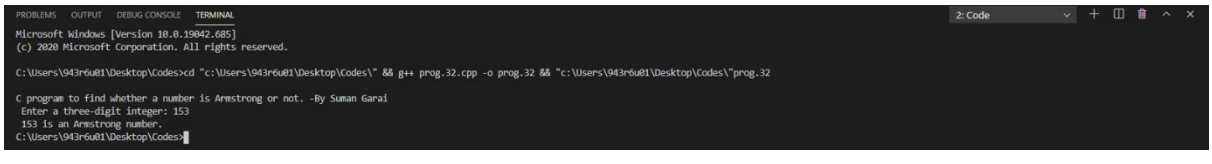
## Experiment 32:

*C program using functions to check whether a three-digit number is an Armstrong number or not.*

## Code:

```
#include <stdio.h>
int work (int);
int main()
{
    int num, result = 0;
    printf("\nC program to find whether a number is
    Armstrong or not. -By Suman Garai");
    printf(" Enter a three-digit integer: ");
    scanf("%d", &num);
    result = work (num);
    if (result == num)
        printf(" %d is an Armstrong number.", num);
    else
        printf(" %d is not an Armstrong number.",
num);
}
int work (int num)
{
    int originalNum, remainder, result = 0;
    originalNum = num;
    while (originalNum != 0)
    {
       remainder = originalNum % 10;
       result += remainder * remainder * remainder;
       originalNum /= 10;
    }
    return result;
}
```

## Output:



## Experiment 33:

*C program using functions to calculate compound interest.*

## Code:

```
#include <stdio.h>
#include <math.h>
int main()
{
 float p, r, t, CI;
 float compoundInterest(float,float,float);
 printf("\nC program using functions to calculate
 compound interest.  -By Suman Garai\n");
 printf("Enter principle (amount): ");
 scanf("%f", &p);
 printf("Enter time: ");
 scanf("%f", &t);
 printf("Enter rate: ");
 scanf("%f", &r);
 CI=compoundInterest(p,r,t);
 printf("Compound Interest = %f",(CI));
 return 0;
}
float compoundInterest(float p,float r,float t)
{
float ci;
 ci=p*pow((1 + r / 100),t)-p;
 return ci;
}
```

## Output:

-- THE END --