Assignment 3:

Arjun Patel – FRN006

Q1)Print 1 to 10

```c
#include<stdio.h>
int main(){
    // int i=1;
    // while(i<11){
    //     printf("%d\n", i);
    //     i++;
    // }
    for (int i = 1; i < 11; i++)
    {
        printf("%d\n", i);
    }

    return 0;
}
```

```
PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\As
nnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile
1
2
3
4
5
6
7
8
9
10
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 3>
```

Q2) Print table for the given number.

```c
#include <stdio.h>
int main()
{
    int n, i = 1;
    printf("Enter Any Number u want to print table of\n");
    scanf("%d", &n);

    // while (i < 11)
    // {
    //     printf("%d x %d = %d\n", n, i, n * i);
    //     i++;
    // }
    for (int i = 1; i <= 10; i++)
    {
        printf("%d x %d = %d\n", n, i, n * i);
```

```
    }
    return 0;
}
```

```
le.c -o 2_printTable } ; if ($?) { .\2_printTable }
Enter Any Number u want to print table of
5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 3>
```

Q3)Calculate sum of numbers in range

```c
#include<stdio.h>
int main(){
    int lower, upper, sum=0;
    printf("Enter num from which u want sum(Lower limit)\n");
    scanf("%d", &lower);
    printf("Enter num to which u want sum(Upper limit)\n");
    scanf("%d", &upper);
    // int lowerTemp = lower, upperTemp = upper;

    // while(lower<=upper){
    //     sum += lower;
    //     lower++;2
    // }
    for (int i = lower; i <= upper; i++)
    {

        sum += i;
    }

    // printf("Sum of numbers from %d to %d is %d", lowerTemp, upperTemp,
sum);
    printf("Sum of numbers from %d to %d is %d", lower, upper, sum);
    return 0;
}
```

```
PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\A
OfRange.c -o 3_sumOfNumOfRange } ; if ($?) { .\3_sumOfNumOfRange }
Enter num from which u want sum(Lower limit)
2
Enter num to which u want sum(Upper limit)
10
Sum of numbers from 2 to 10 is 54
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 3>
```

Q4)Check Prime number

```c
#include <stdio.h>
int main()
{
    int num, isPrime = 1;
    printf("Enter number u want to check prime of\n");
    scanf("%d", &num);

    // if num can be divided with 2 to num-1, its not prime
    // int i = 2;
    // while (i < num)
    // {
    //     if (num % i == 0)
    //     {
    //         isPrime = 0;
    //         break;
    //     }
    //     i++;
    // }
    for (int i = 2; i*i <= num; i++)
    {
        if (num % i == 0)
        {
            isPrime = 0;
            break;
        }
    }
    if (isPrime)
        printf("%d is a Prime Number\n", num);
    else
        printf("%d is NOT a Prime Number\n", num);

    return 0;
}
```

```
PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\A
meNum.c -o 4_checkPrimeNum } ; if ($?) { .\4_checkPrimeNum }
Enter number u want to check prime of
29
29 is a Prime Number
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 3>
```

Q5)Check Armstrong Number

```c
#include <stdio.h>
#include <math.h>
// if 123 is num, and 1^3 + 2^3 + 3^3 = 123, then its armstrong num
//example 153 = 1 + 125+ 27 is armstrong num
//1634 = 1^4 + 6^4 + 3^4 + 4^4
int main()
{
    int num;
    printf("Enter a number:\n");
    scanf("%d", &num);

    int temp = num, sum = 0, count =0;

    //find length of number to find exponent
    while(temp>0){
        count++;
        temp /= 10;
    }
    printf("Count = %d\n", count);

    //temp becomes 0, so ressign for further use
    temp = num;
    while (temp > 0)
    {
        int rem = temp % 10;
        //cal power of rem
        int power = 1, tempCount = count;
        // while(tempCount--){
        //     power *= rem;
        // }
        for (int i = 1; i <= tempCount; i++)
        {
            power *= rem;
        }

        printf("Power = %d\n", power);
        sum += power;
        temp /= 10;
    }

    sum == num ? printf("%d is an armstrong number\n", num) : printf("%d is
not An Armstrong number\n", num);
    return 0;
}
```

```
PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\Assi
gNum.c -o 5_armstrongNum } ; if ($?) { .\5_armstrongNum }
Enter a number:
1634
Count = 4
Power = 256
Power = 81
Power = 1296
Power = 1
1634 is an armstrong number
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 3>
```

Q6)Perfect Number

```c
#include<stdio.h>
//number can be called perfect if, sum of its divisors is same as number
itself
//ex: 6 because 1 + 2 + 3 = 6
//28 beacuse, 1 +2 + 4 + 7 + 14 =28
int main(){
    int num;
    printf("Enter a num:\n");
    scanf("%d", &num);

    int temp = num, sum =0;
    // int temp = num, sum =0, divisor=1;
    // while(divisor<=temp/2){
    //     if(temp % divisor == 0){
    //         sum += divisor;
    //     }
    //     divisor++;
    // }
    for (int i = 1; i < temp; i++)
    {
        if(num%i==0) sum += i;
    }

    if(temp==sum) printf("%d is a Perfect number\n", num);
    else printf("%s is not a Perfect num\n", num);
    return 0;
}
```

```
PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\
Perfect.c -o 6_checkNumPerfect } ; if ($?) { .\6_checkNumPerfect }
Enter a num:
28
28 is a Perfect number
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 3>
```

Q7)Find Factorial

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

```c
int main(){
    int num, fact = 1;
    printf("Enter a number:\n");
    scanf("%d", &num);
    int temp =num;

    // while(num>0){

    //     fact *=  num;
    //     num--;
    // }
    for(int i=num; i>0;i--){
        fact *= i;
    }
    printf("%d! = %d",temp, fact);

    return 0;
}
```

```
PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\As
orial.c -o 7_findFactorial } ; if ($?) { .\7_findFactorial }
Enter a number:
4
4! = 24
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 3>
```

Q8)Strong Number

```c
#include <stdio.h>
// num is called strong if its sum of its digit's factorial is same as num
// ex: 145, 1! + 4!+ 5! = 145
int main()
{
    int num;
    printf("Enter a number:\n");
    scanf("%d", &num);

    int temp = num, rem, sum = 0;
    while (temp > 0)
    {
        rem = temp % 10;

        //--------Factorial Part-------
        // find factorial of rem
        int factorial = 1;
        while (rem > 0)
        {
            factorial *= rem;
            rem--;
        }
        // add factorial of rem to sum
```

```
        sum += factorial;

        // continue
        temp /= 10;
    }

    if (sum == num)
        printf("%d is a Strong Number\n", num);
    else
        printf("%d is NOT a Strong Number\n", num);
    return 0;
}
```

```
PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\Assi
mber.c -o 8_strongNumber } ; if ($?) { .\8_strongNumber }
Enter a number:
145
145 is a Strong Number
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 3>
```

Q9) Palindrome

```c
#include <stdio.h>
// 121, 1331, 12321
int main()
{
    int num;
    printf("Enter a number:\n");
    scanf("%d", &num);

    int temp = num, rem, rev = 0;
    while (temp > 0)
    {
        rem = temp % 10;
        rev = rev * 10 + rem;
        temp /= 10;
    }

    if (num == rev)
        printf("%d is a Palindrome number\n", num);
    else
        printf("%d is NOT a Palindrome number\n", num);
    return 0;
}
```

```
PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\As
indrome.c -o 9_checkPalindrome } ; if ($?) { .\9_checkPalindrome }
Enter a number:
123321
123321 is a Palindrome number
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 3>
```

Q10) Summation of first digit and last digit of a number

```c
#include<stdio.h>
#include<math.h>
//add first and and last digit of given num
int main(){
    int num;
    printf("Enter a number:\n");
    scanf("%d", &num);

    int temp = num, lastDigit, firstDigit, lengthOfNum=0;
    lastDigit = temp%10;

    //logic 1 for find 1st digit of num
    // while(temp>0){
    //     lengthOfNum++;
    //     temp /= 10;
    // }
    // //reassign temp to num
    // temp = num;
    // firstDigit = temp / pow(10,lengthOfNum-1);

    //logic 2 for find 1st digit of num
    while (temp>0)
    {
        // if(temp/10==0){
        //     firstDigit =temp;
        //     break;
        // }
        firstDigit = temp%10;
        temp /= 10;
    }


    printf("Sum of first digit(%d) + last Digit(%d) = %d", firstDigit,
lastDigit, firstDigit+lastDigit);
    return 0;
}
```

```
PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\As
tNdLastDigit.c -o 10_addFirstNdLastDigit } ; if ($?) { .\10_addFirst
Enter a number:
1235
Sum of first digit(1) + last Digit(5) = 6
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 3>
```