

Assignment 4:

Arjun Patel - FRN006

Q1) Print Armstrong number in range 1 to n

```
#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 findPower(int base, int exponent)
{
    int power = 1;
    while (exponent--)
    {
        power *= base;
    }
    return power;
}

int main()
{
    int num;
    printf("Enter a number upto which u want to armstrong nums:\n");
    scanf("%d", &num);

    for (int i = 1; i <= num; i++)
    {
        int temp = i, sum = 0, count = 0;

        // find length of number to find exponent
        while (temp > 0)
        {
            count++;
            temp /= 10;
        }

        // temp becomes 0, so reassign for further use
        temp = i;
        while (temp > 0)
        {
            int rem = temp % 10;
            // cal power of rem
            int power = 1, exponent = count;
            sum += findPower(rem, exponent);
            temp /= 10;
        }
        sum == i && printf("%d ", i);
    }
}
```

```

    }

    return 0;
}

```

```

PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\Assignments\Assignment 4" & gcc 1_armstrongForGivenRange.c -o 1_armstrongForGivenRange ; if ($?) { .\1_armstrongForGivenRange }
Enter a number upto which u want to armstrong nums:
1634
1 2 3 4 5 6 7 8 9 153 370 371 407 1634
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 4> |

```

Q2) Check Prime number in Range

```

#include <stdio.h>
void checkPrime(int num)
{
    int isPrime;
    for (int i = 2; i * i <= num; i++)
    {
        isPrime = 1;
        if (num % i == 0)
        {
            isPrime = 0;
            break;
        }
    }
    if (isPrime)
    {
        printf("%d ", num);
    }
}

int main()
{
    int num, isPrime = 1;
    printf("Enter number upto which u want to check prime of\n");
    scanf("%d", &num);

    for (int i = 1; i <= num; i++)
    {
        checkPrime(i);
    }

    return 0;
}

```

```

● PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\Assignments\
InRange.c -o 2_primeDayInRange } ; if ($?) { .\2_primeDayInRange }
Enter number upto which u want to check prime of
100
1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
○ PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 4> |

```

Q3) Check Perfect num in range

```

#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

void checkPerfect(int num)
{
    int sum = 0;
    for (int i = 1; i <= num/2; i++)
    {
        if(num%i==0) sum += i;
    }

    if (num == sum)
        printf("%d ", num);
}

int main()
{
    int num;
    printf("Upto which range u want to check perfect num: ");
    scanf("%d", &num);

    for (int i = 1; i <= num; i++)
    {
        checkPerfect(i);
    }

    return 0;
}

```

```

● PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\Assign
fectNoInRange.c -o 3_checkPerfectNoInRange } ; if ($?) { .\3_checkPerfe
Upto which range u want to check perfect num: 500
6 28 496
○ PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 4>

```

Q4) Check Strong number in range 1 to n

```

#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

void checkStrong(int 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 ", num);
}

int main(){
    int num;
    printf("Enter a number:\n");
    scanf("%d", &num);

    for (int i = 1; i <= num; i++)
    {
        checkStrong(i);
    }
    return 0;
}

```

```

PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\Assignment 4"
mInRange.c -o 4_strongNumInRange } ; if ($?) { .\4_strongNumInRange
Enter a number:
1000
1 2 145
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 4>

```

Q5) Print Fibonacci upto n number

```

#include<stdio.h>
//0 1 1 2 3 5 8 13 21 34 55
int main(){
    int num, first =0, second = 1, next = 0;

```

```

printf("Enter a number\n");
scanf("%d", &num);

while (next<=num)
{
    printf("%d ", next);
    first = second;
    second = next;
    next = first + second;
}

return 0;
}

```

```

PS D:\Firstbit Solutions> cd "d:\Firstbit Solutions\C Programming\
i.c -o 5_fibonacci } ; if ($?) { .\5_fibonacci }
Enter a number
200
0 1 1 2 3 5 8 13 21 34 55 89 144
PS D:\Firstbit Solutions\C Programming\Assignments\Assignment 4>

```