

## Assignment 4

**// 1. Print armstrong number in the the given range 1 to n?**

```
int start = 1, end = 1000;

printf("Armstrong numbers between %d and %d:\n", start, end);

for (int num = start; num <= end; num++) {

    int original = num, sum = 0, digits = 0, temp = num;

    while (temp > 0) {

        temp /= 10;

        digits++;

    }

    temp = num;

    while (temp > 0) {

        int digit = temp % 10;

        int i=0;

        int power=1;

        while (i<digits)

        {

            power = power * digit;

            i++;

        }

        sum += power;

        temp /= 10;

    }

    if (sum == original) {

        printf("%d ", num);

    }

}
```

### Output:

```
PS C:\Users\manis\OneDrive\Desktop\New folder\output> & .\'assignment4.exe'  
Armstrong numbers between 1 and 1000:  
1 2 3 4 5 6 7 8 9 153 370 371 407
```

## // 2. Print prime number in the given range 1 to n?

```
int start = 1, end = 1000;
```

```
printf("Prime numbers between %d and %d:\n", start, end);
```

```
for (int num = start; num <= end; num++) {
```

```
    if (num < 2) continue;
```

```
    int is_prime = 1;
```

```
    for (int i = 2; i * i <= num; i++) {
```

```
        if (num % i == 0) {
```

```
            is_prime = 0;
```

```
            break;
```

```
        }
```

```
    }
```

```
    if (is_prime) {
```

```
        printf("%d ", num);
```

```
    }
```

```
}
```

## Output:

```
PS C:\Users\manis\OneDrive\Desktop\New folder\output> & .\assignment4.exe
Prime numbers between 1 and 1000:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223 227 229 233 239 241 251
257 263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 359 367 373 379 383 389 397 401 409 419 421 431 433 439 443 449 457 461 463 467 479 487 491 499 503 509 521 523 541 5
47 557 563 569 571 577 587 593 599 601 607 613 617 619 631 641 643 647 653 659 661 673 677 683 691 701 709 719 727 733 739 743 751 757 761 769 773 787 797 809 811 821 823 827 829 839 853
857 859 863 877 881 883 887 907 911 919 929 937 941 947 953 967 971 977 983 991 997
```

**// 3. check perfect number in the given range 1 to n?**

```
int start = 1, end = 1000;
```

```
printf("Perfect numbers between %d and %d:\n", start, end);
```

```
for (int num = start; num <= end; num++) {
```

```
    int sum = 0;
```

```
    for (int i = 1; i <= num / 2; i++) {
```

```
        if (num % i == 0) {
```

```
            sum += i;
```

```
        }
```

```
    }
```

```
    if (sum == num) {
```

```
        printf("%d ", num);
```

```
    }
```

```
}
```

**Output:**

```
PS C:\Users\manis\OneDrive\Desktop\New folder\output> & .\'assignment4.exe'  
Perfect numbers between 1 and 1000:  
6 28 496
```

**// 4. check strong number in the given range 1 to n?**

```
int start = 1, end = 1000;

printf("Strong numbers between %d and %d:\n", start, end);

for (int num = start; num <= end; num++) {

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

    while (temp > 0) {

        int digit = temp % 10;

        int factorial = 1;

        for (int i = 1; i <= digit; i++) {

            factorial *= i;

        }

        sum += factorial;

        temp /= 10;

    }

    if (sum == original) {

        printf("%d ", num);

    }

}
```

**Output:**

```
PS C:\Users\manis\OneDrive\Desktop\New folder\output> & .\'assignment4.exe'
Strong numbers between 1 and 1000:
1 2 145
```

### // 5. Print fibonacci series?(optional)

```
int no = 10;

int first_term = 0;

int second_term = 1;

printf("Fibonacci Series up to %d term: ", no);

for (int i = 0; i < no; i++)
{
    printf("%d, ", first_term);

    int next_term = first_term + second_term;

    first_term = second_term;
    second_term = next_term;

}
```

### Output:

```
PS C:\Users\manis\OneDrive\Desktop\New folder\output> & .\'assignment4.exe'
Fibonacci Series up to 10 term: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```