```c
/**
 * C program to all natural numbers in reverse from n to 1
 */

#include <stdio.h>

int main()
{
    int i, start;

    /* Input start range from user */
    printf("Enter starting value: ");
    scanf("%d", &start);

    /*
     * Run loop from 'start' to 1 and
     * decrement 1 in each iteration
     */
    for(i=start; i>=1; i--)
    {
        printf("%d\n", i);
    }

    return 0;
}

/**
 * C program to print all alphabets from a to z
 */

#include <stdio.h>

int main()
{
    char ch;

    printf("Alphabets from a - z are: \n");
    for(ch='a'; ch<='z'; ch++)
    {
        printf("%c\n", ch);
    }

    return 0;
}

/**
 * C program to print all even numbers from 1 to n
 */

#include <stdio.h>

int main()
{
    int i, n;

    /* Input upper limit of even number from user */
    printf("Print all even numbers till: ");
    scanf("%d", &n);

    printf("Even numbers from 1 to %d are: \n", n);

    /*
     * Start loop counter from 1, increment it by 1,
     * will iterate till n
     */
    for(i=1; i<=n; i++)
```

```
    {
        /* Check even condition before printing */
        if(i%2 == 0)
        {
            printf("%d\n", i);
        }
    }

    return 0;
}
```

```
/**
 * C program to print all Odd numbers from 1 to n
 */

#include <stdio.h>

int main()
{
    int i, n;

    /* Input upper limit from user */
    printf("Print odd numbers till: ");
    scanf("%d", &n);

    printf("All odd numbers from 1 to %d are: \n", n);

    /* Start loop from 1 and increment it by 1 */
    for(i=1; i<=n; i++)
    {
        /* If 'i' is odd then print it */
        if(i%2!=0)
        {
            printf("%d\n", i);
        }
    }

    return 0;
}
```

```
/**
 * C program to find sum of natural numbers between 1 to n
 */

#include <stdio.h>

int main()
{
    int i, n, sum=0;

    /* Input upper limit from user */
    printf("Enter upper limit: ");
    scanf("%d", &n);

    /* Find sum of all numbers */
    for(i=1; i<=n; i++)
    {
        sum += i;
    }

    printf("Sum of first %d natural numbers = %d", n, sum);

    return 0;
}

/**
 * C program to find sum of natural numbers between 1 to n
```

```
 */

#include <stdio.h>

int main()
{
    int i, n, sum=0;

    /* Input upper limit from user */
    printf("Enter upper limit: ");
    scanf("%d", &n);

    /* Find sum of all numbers */
    for(i=1; i<=n; i++)
    {
        sum += i;
    }

    printf("Sum of first %d natural numbers = %d", n, sum);

    return 0;
}

/**
 * C program to print the sum of all odd numbers from 1 to n
 */

#include <stdio.h>

int main()
{
    int i, n, sum=0;

    /* Input range to find sum of odd numbers */
    printf("Enter upper limit: ");
    scanf("%d", &n);

    /* Find the sum of all odd number */
    for(i=1; i<=n; i+=2)
    {
        sum += i;
    }

    printf("Sum of odd numbers = %d", sum);

    return 0;
}

/**
 * C program to print multiplication table of a number
 */

#include <stdio.h>

int main()
{
    int i, num;

    /* Input a number to print table */
    printf("Enter number to print table: ");
    scanf("%d", &num);

    for(i=1; i<=10; i++)
    {
        printf("%d * %d = %d\n", num, i, (num*i));
    }
```

```c
    return 0;
}

/**
 * C program to count number of digits in an integer
 */

#include <stdio.h>

int main()
{
    long long num;
    int count = 0;

    /* Input number from user */
    printf("Enter any number: ");
    scanf("%lld", &num);

    /* Run loop till num is greater than 0 */
    do
    {
        /* Increment digit count */
        count++;

        /* Remove last digit of 'num' */
        num /= 10;
    } while(num != 0);

    printf("Total digits: %d", count);

    return 0;
}

/**
 * C program to find last digit of a number
 */

#include <stdio.h>

int main()
{
    int n, lastDigit;

    /* Input number from user */
    printf("Enter any number: ");
    scanf("%d", &n);

    /* Get the last digit */
    lastDigit = n % 10;

    printf("Last digit = %d", lastDigit);

    return 0;
}

/**
 * C program to find first digit of a number
 */

#include <stdio.h>

int main()
{
    int n, first;
```

```
    /* Input number from user */
    printf("Enter any number: ");
    scanf("%d", &n);

    first = n;

    /* Remove last digit from number till only one digit is left */
    while(first >= 10)
    {
        first = first / 10;
    }

    printf("First digit = %d", first);

    return 0;
}

/**
 * C program to find sum of first and last digit of a number using loop
 */

#include <stdio.h>

int main()
{
    int num, sum=0, firstDigit, lastDigit;

    /* Input a number from user */
    printf("Enter any number to find sum of first and last digit: ");
    scanf("%d", &num);

    /* Find last digit to sum */
    lastDigit = num % 10;


    /* Copy num to first digit */
    firstDigit = num;

    /* Find the first digit by dividing num by 10 until first digit is left */
    while(num >= 10)
    {
        num = num / 10;
    }
    firstDigit = num;


    /* Find sum of first and last digit*/
    sum = firstDigit + lastDigit;

    printf("Sum of first and last digit = %d", sum);

    return 0;
}

/**
 * C program to find sum of its digits of a number
 */

#include <stdio.h>

int main()
{
    int num, sum=0;

    /* Input a number from user */
```

```
    printf("Enter any number to find sum of its digit: ");
    scanf("%d", &num);

    /* Repeat till num becomes 0 */
    while(num!=0)
    {
        /* Find last digit of num and add to sum */
        sum += num % 10;

        /* Remove last digit from num */
        num = num / 10;
    }

    printf("Sum of digits = %d", sum);

    return 0;
}
```

```
/**
 * C program to calculate product of digits of a number
 */

#include <stdio.h>

int main()
{
    int num;
    long long product=1ll;

    /* Input number from user */
    printf("Enter any number to calculate product of digit: ");
    scanf("%d", &num);

    product = (num == 0 ? 0 : 1ll);

    /* Repeat the steps till num becomes 0 */
    while(num != 0)
    {
        /* Get the last digit from num and multiplies to product */
        product = product * (num % 10);

        /* Remove the last digit from n */
        num = num / 10;
    }

    printf("Product of digits = %lld", product);

    return 0;
}
```

```
/**
 * C program to find reverse of a number
 */

#include <stdio.h>

int main()
{
    int num, reverse = 0;

    /* Input a number from user */
    printf("Enter any number to find reverse: ");
    scanf("%d", &num);

    /* Repeat the till 'num' becomes 0 */
    while(num != 0)
    {
```

```
        /*
         * Increase place value of reverse and
         * add last digit to reverse
         */
        reverse = (reverse * 10) + (num % 10);

        /* Remove last digit from 'num' */
        num /= 10;
    }

    printf("Reverse = %d", reverse);

    return 0;

}

/**
 * C program to check whether a number is palindrome or not
 */

#include <stdio.h>

int main()
{
    int n, num, rev = 0;

    /* Input a number from user */
    printf("Enter any number to check palindrome: ");
    scanf("%d", &n);

    /* Copy original value to 'num' to 'n'*/
    num = n;

    /* Find reverse of n and store in rev */
    while(n != 0)
    {
        rev = (rev * 10) + (n % 10);
        n  /= 10;
    }

    /* Check if reverse is equal to 'num' or not */
    if(rev == num)
    {
        printf("%d is palindrome.", num);
    }
    else
    {
        printf("%d is not palindrome.", num);
    }

    return 0;

}

/**
 * C program to count frequency of digits in a given number
 */

#include <stdio.h>
#define BASE 10 /* Constant */

int main()
{
    long long num, n;
```

```
    int i, lastDigit;
    int freq[BASE];

    /* Input number from user */
    printf("Enter any number: ");
    scanf("%lld", &num);

    /* Initialize frequency array with 0 */
    for(i=0; i<BASE; i++)
    {
        freq[i] = 0;
    }

    /* Copy the value of 'num' to 'n' */
    n = num;

    /* Run till 'n' is not equal to zero */
    while(n != 0)
    {
        /* Get last digit */
        lastDigit = n % 10;

        /* Remove last digit */
        n /= 10;

        /* Increment frequency array */
        freq[lastDigit]++;
    }

    /* Print frequency of each digit */
    printf("Frequency of each digit in %lld is: \n", num);
    for(i=0; i<BASE; i++)
    {
        printf("Frequency of %d = %d\n", i, freq[i]);
    }

    return 0;

}

/**
 * C program to print number in words
 */

#include <stdio.h>

int main()
{
    int n, num = 0;

    /* Input number from user */
    printf("Enter any number to print in words: ");
    scanf("%d", &n);

    /* Store reverse of n in num */
    while(n != 0)
    {
        num = (num * 10) + (n % 10);
        n /= 10;
    }

    /*
     * Extract last digit of number and print corresponding digit in words
     * till num becomes 0
     */
```

```c
    while(num != 0)
    {
        switch(num % 10)
        {
            case 0:
                printf("Zero ");
                break;
            case 1:
                printf("One ");
                break;
            case 2:
                printf("Two ");
                break;
            case 3:
                printf("Three ");
                break;
            case 4:
                printf("Four ");
                break;
            case 5:
                printf("Five ");
                break;
            case 6:
                printf("Six ");
                break;
            case 7:
                printf("Seven ");
                break;
            case 8:
                printf("Eight ");
                break;
            case 9:
                printf("Nine ");
                break;
        }

        num = num / 10;
    }

    return 0;

}

/**
 * C program to print ASCII values of all characters.
 */

#include <stdio.h>

int main()
{
    int i;

    /* Print ASCII values from 0 to 255 */
    for(i=0; i<=255; i++)
    {
        printf("ASCII value of character %c = %d\n", i, i);
    }

    return 0;

}

/**
```

```c
 * C program to find power of any number using for loop
 */

#include <stdio.h>

int main()
{
    int base, exponent;
    long long power = 1;
    int i;

    /* Input base and exponent from user */
    printf("Enter base: ");
    scanf("%d", &base);
    printf("Enter exponent: ");
    scanf("%d", &exponent);

    /* Multiply base, exponent times*/
    for(i=1; i<=exponent; i++)
    {
        power = power * base;
    }

    printf("%d ^ %d = %lld", base, exponent, power);

    return 0;

}

/**
 * C program to print all factors of a number
 */

#include <stdio.h>

int main()
{
    int i, num;

    /* Input number from user */
    printf("Enter any number to find its factor: ");
    scanf("%d", &num);

    printf("All factors of %d are: \n", num);

    /* Iterate from 1 to num */
    for(i=1; i<=num; i++)
    {
        /*
         * If num is exactly divisible by i
         * Then i is a factor of num
         */
        if(num % i == 0)
        {
            printf("%d, ",i);
        }
    }

    return 0;

}

/**
 * C program to calculate factorial of a number
```

```
 */

#include <stdio.h>

int main()
{
    int i, num;
    unsigned long long fact=1LL;

    /* Input number from user */
    printf("Enter any number to calculate factorial: ");
    scanf("%d", &num);

    /* Run loop from 1 to num */
    for(i=1; i<=num; i++)
    {
        fact = fact * i;
    }

    printf("Factorial of %d = %llu", num, fact);

    return 0;

}

/**
 * C program to find HCF of two numbers
 */

#include <stdio.h>

int main()
{
    int i, num1, num2, min, hcf=1;

    /* Input two numbers from user */
    printf("Enter any two numbers to find HCF: ");
    scanf("%d%d", &num1, &num2);

    /* Find minimum between two numbers */
    min = (num1<num2) ? num1 : num2;

    for(i=1; i<=min; i++)
    {
        /* If i is factor of both number */
        if(num1%i==0 && num2%i==0)
        {
            hcf = i;
        }
    }

    printf("HCF of %d and %d = %d\n", num1, num2, hcf);

    return 0;

}

/**
 * C program to find LCM of any two numbers
 */

#include <stdio.h>

int main()
```

```
{
    int i, num1, num2, max, lcm=1;

    /* Input two numbers from user */
    printf("Enter any two numbers to find LCM: ");
    scanf("%d%d", &num1, &num2);

    /* Find maximum between num1 and num2 */
    max = (num1 > num2) ? num1 : num2;

    /* First multiple to be checked */
    i = max;

    /* Run loop indefinitely till LCM is found */
    while(1)
    {
        if(i%num1==0 && i%num2==0)
        {
            /*
             * If 'i' divides both 'num1' and 'num2'
             * then 'i' is the LCM.
             */
            lcm = i;

            /* Terminate the loop after LCM is found */
            break;
        }

        /*
         * If LCM is not found then generate next
         * multiple of max between both numbers
         */
        i += max;
    }

    printf("LCM of %d and %d = %d", num1, num2, lcm);

    return 0;

}

/**
 * C program to whether a number is prime number or not
 */

#include <stdio.h>

int main()
{
    int i, num, isPrime;

    /*
     * isPrime is used as flag variable.
     * If isPrime = 0, then number is composite
     * else if isPrime = 1, then number is prime.
     * Initially I have assumed the number as prime.
     */
    isPrime = 1;

    /* Input a number from user */
    printf("Enter any number to check prime: ");
    scanf("%d", &num);

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

```
        /* Check divisibility of num */
        if(num%i==0)
        {
            /* Set isPrime to 0 indicating it as composite number */
            isPrime = 0;

            /* Terminate from loop */
            break;
        }
    }

    /*
     * If isPrime contains 1 then it is prime
     */
    if(isPrime == 1 && num > 1)
    {
        printf("%d is prime number", num);
    }
    else
    {
        printf("%d is composite number", num);
    }

    return 0;

}

/**
 * C program to print all prime numbers between 1 to n
 */

#include <stdio.h>

int main()
{
    int i, j, end, isPrime; // isPrime is used as flag variable

    /* Input upper limit to print prime */
    printf("Find prime numbers between 1 to : ");
    scanf("%d", &end);

    printf("All prime numbers between 1 to %d are:\n", end);

    /* Find all Prime numbers between 1 to end */
    for(i=2; i<=end; i++)
    {
        /* Assume that the current number is Prime */
        isPrime = 1;

        /* Check if the current number i is prime or not */
        for(j=2; j<=i/2; j++)
        {
            /*
             * If i is divisible by any number other than 1 and self
             * then it is not prime number
             */
            if(i%j==0)
            {
                isPrime = 0;
                break;
            }
        }

        /* If the number is prime then print */
        if(isPrime==1)
```

```
        {
            printf("%d, ", i);
        }
    }

    return 0;

}

/**
 * C program to find sum of prime numbers between 1 to n
 */

#include <stdio.h>

int main()
{
    int i, j, end, isPrime, sum=0;

    /* Input upper limit from user */
    printf("Find sum of all prime between 1 to : ");
    scanf("%d", &end);

    /* Find all prime numbers between 1 to end */
    for(i=2; i<=end; i++)
    {

        /* Check if the current number i is Prime or not */
        isPrime = 1;
        for(j=2; j<=i/2 ;j++)
        {
            if(i%j==0)
            {
                /* 'i' is not prime */
                isPrime = 0;
                break;
            }
        }

        /*
         * If 'i' is Prime then add to sum
         */
        if(isPrime==1)
        {
            sum += i;
        }
    }

    printf("Sum of all prime numbers between 1 to %d = %d", end, sum);

    return 0;

}

/**
 * C program to find all prime factors of a given number
 */

#include <stdio.h>

int main()
{
    int i, j, num, isPrime;
```

```
    /* Input a number from user */
    printf("Enter any number to print Prime factors: ");
    scanf("%d", &num);

    printf("All Prime Factors of %d are: \n", num);

    /* Find all Prime factors */
    for(i=2; i<=num; i++)
    {
        /* Check 'i' for factor of num */
        if(num%i==0)
        {
            /* Check 'i' for Prime */
            isPrime = 1;
            for(j=2; j<=i/2; j++)
            {
                if(i%j==0)
                {
                    isPrime = 0;
                    break;
                }
            }

            /* If 'i' is Prime number and factor of num */
            if(isPrime==1)
            {
                printf("%d, ", i);
            }
        }
    }

    return 0;

}

/**
 * C program to check Armstrong number
 */
#include <stdio.h>
#include <math.h>

int main()
{
    int originalNum, num, lastDigit, digits, sum;

    /* Input number from user */
    printf("Enter any number to check Armstrong number: ");
    scanf("%d", &num);

    sum = 0;

    /* Copy the value of num for processing */
    originalNum = num;

    /* Find total digits in num */
    digits = (int) log10(num) + 1;

    /* Calculate sum of power of digits */
    while(num > 0)
    {
        /* Extract the last digit */
        lastDigit = num % 10;

        /* Compute sum of power of last digit */
        sum = sum + round(pow(lastDigit, digits));
```

```
        /* Remove the last digit */
        num = num / 10;
    }

    /* Check for Armstrong number */
    if(originalNum == sum)
    {
        printf("%d is ARMSTRONG NUMBER", originalNum);
    }
    else
    {
        printf("%d is NOT ARMSTRONG NUMBER", originalNum);
    }

    return 0;

}

/**
 * C program to print Armstrong numbers from 1 to n
 */
#include <stdio.h>
#include <math.h>

int main()
{
    int num, lastDigit, digits, sum, i, end;

    /* Input upper limit from user */
    printf("Enter upper limit: ");
    scanf("%d", &end);

    printf("Armstrong number between 1 to %d are: \n", end);

    for(i=1; i<=end; i++)
    {
        sum = 0;

        /* Copy the value of num for processing */
        num = i;

        /* Find total digits in num */
        digits = (int) log10(num) + 1;

        /* Calculate sum of power of digits */
        while(num > 0)
        {
            /* Extract last digit */
            lastDigit = num % 10;

            // Find sum of power of digits
            // Use ceil() function to overcome any rounding errors by pow()
            sum = sum + ceil(pow(lastDigit, digits));

            /* Remove the last digit */
            num = num / 10;
        }

        /* Check for Armstrong number */
        if(i == sum)
        {
            printf("%d, ", i);
        }
```

```
    }

    return 0;

}

/**
 * C program to check whether a number is Perfect number or not
 */

#include <stdio.h>

int main()
{
    int i, num, sum = 0;

    /* Input a number from user */
    printf("Enter any number to check perfect number: ");
    scanf("%d", &num);

    /* Calculate sum of all proper divisors */
    for(i = 1; i <= num / 2; i++)
    {
        /* If i is a divisor of num */
        if(num%i == 0)
        {
            sum += i;
        }
    }

    /* Check whether the sum of proper divisors is equal to num */
    if(sum == num && num > 0)
    {
        printf("%d is PERFECT NUMBER", num);
    }
    else
    {
        printf("%d is NOT PERFECT NUMBER", num);
    }

    return 0;

}

/**
 * C program to print all Perfect numbers between 1 to n
 */

#include <stdio.h>

int main()
{
    int i, j, end, sum;

    /* Input upper limit to print perfect number */
    printf("Enter upper limit: ");
    scanf("%d", &end);

    printf("All Perfect numbers between 1 to %d:\n", end);

    /* Iterate from 1 to end */
    for(i=1; i<=end; i++)
    {
        sum = 0;
```

```
        /* Check whether the current number i is Perfect number or not */
        for(j=1; j<i; j++)
        {
            if(i % j == 0)
            {
                sum += j;
            }
        }

        /* If the current number i is Perfect number */
        if(sum == i)
        {
            printf("%d, ", i);
        }
    }

    return 0;

}

/**
 * C program to check whether a number is Strong Number or not
 */

#include <stdio.h>

int main()
{
    int i, originalNum, num, lastDigit, sum;
    long fact;

    /* Input a number from user */
    printf("Enter any number to check Strong number: ");
    scanf("%d", &num);

    /* Copy the value of num to a temporary variable */
    originalNum = num;

    sum = 0;

    /* Find sum of factorial of digits */
    while(num > 0)
    {

        /* Get last digit of num */
        lastDigit = num % 10;

        /* Find factorial of last digit */
        fact = 1;
        for(i=1; i<=lastDigit; i++)
        {
            fact = fact * i;
        }

        /* Add factorial to sum */
        sum = sum + fact;

        num = num / 10;
    }

    /* Check Strong number condition */
    if(sum == originalNum)
    {
        printf("%d is STRONG NUMBER", originalNum);
```

```
    }
    else
    {
        printf("%d is NOT STRONG NUMBER", originalNum);
    }

    return 0;

}

/**
 * C program to print all Strong Numbers between 1 to n
 */

#include <stdio.h>

int main()
{
    int i, j, cur, lastDigit, end;
    long long fact, sum;

    /* Input upper limit from user */
    printf("Enter upper limit: ");
    scanf("%d", &end);

    printf("All Strong numbers between 1 to %d are:\n", end);

    /* Iterate from 1 to end */
    for(i=1; i<=end; i++)
    {
        /* Number to check for strong number */
        cur = i;

        sum = 0;

        /* Find the sum of factorial of digits */
        while(cur > 0)
        {
            fact = 1ll;
            lastDigit = cur % 10;

            /* Find factorial of last digit of current num. */
            for( j=1; j<=lastDigit; j++)
            {
                fact = fact * j;
            }

            sum += fact;

            cur /= 10;
        }

        /* Print 'i' if it is strong number */
        if(sum == i)
        {
            printf("%d, ", i);
        }
    }

    return 0;

}

/**
```

```c
 * C program to print Fibonacci series up to n terms
 */

#include <stdio.h>

int main()
{
    int a, b, c, i, terms;

    /* Input number from user */
    printf("Enter number of terms: ");
    scanf("%d", &terms);

    /* Fibonacci magic initialization */
    a = 0;
    b = 1;
    c = 0;

    printf("Fibonacci terms: \n");

    /* Iterate through n terms */
    for(i=1; i<=terms; i++)
    {
        printf("%d, ", c);

        a = b;     // Copy n-1 to n-2
        b = c;     // Copy current to n-1
        c = a + b; // New term
    }

    return 0;

}
```