

1. Write a C program to find the factorial value of any number using function function should take the number as argument and return the value

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

```
int factorial(int);
```

```
int main()
```

```
{
```

```
    int num, result;
```

```
    printf("Enter a number: ");
```

```
    scanf("%d", &num);
```

```
    result = factorial(num);
```

```
    printf("Factorial of %d is %d\n", num, result);
```

```
    return 0;
```

```
}
```

```
int factorial(int num)
{
    int i, fact=1;

    for(i=1; i<=num; i++)
    {
        fact = fact * i;
    }

    return fact;
}
```

```
//Output:
//Enter a number: 5
//Factorial of 5 is 120
```

2. Write a C program to find the maximum value of two number using function function shit receive two numbers as argument and return the max of two numbers

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

```
int maximum(int, int);
```

```
int main()
```

```
{
```

```
    int num1, num2, max;
```

```
    printf("Enter first number: ");
```

```
    scanf("%d", &num1);
```

```
    printf("Enter second number: ");
```

```
    scanf("%d", &num2);
```

```
    max = maximum(num1, num2);
```

```
    printf("Maximum of %d and %d is %d\n", num1,  
num2, max);  
  
    return 0;  
}
```

```
int maximum(int a, int b)  
{  
    int c;  
    c = a >= b ? a : b;  
    return c;  
}
```

Output:

```
Enter first number: 25  
Enter second number: 18  
Maximum of 25 and 18 is 25
```

3. Write a C program to check whether a number is prime or not using function should receive the number as a given and return 1 if the number is prime otherwise it should return 0

```
#include<stdio.h>

int isPrime(int);

int main()
{
    int num, result;

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

    result = isPrime(num);

    if(result == 1)
        printf("%d is prime\n", num);
```

```
else  
    printf("%d is not prime\n", num);  
  
return 0;  
}
```

```
int isPrime(int num)  
{  
    int i;  
  
    if(num <= 1)  
        return 0;  
  
    for(i=2; i<num; i++)  
    {  
        if(num % i == 0)  
            return 0;  
    }  
  
    return 1;
```

}

Output:

Enter a number: 7

7 is prime

Enter a number: 10

10 is not prime

4. Write a C program to print Reverse of a number using function function should receive the numbers as argument and print the rivers number

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

```
void reverse(int);
```

```
int main()
```

```
{
```

```
    int num;
```

```
    printf("Enter a number: ");
```

```
    scanf("%d", &num);
```

```
    printf("Reverse of %d is ", num);
```

```
    reverse(num);
```

```
    return 0;
```

```
}
```

```
void reverse(int num)
{
    int rem, rev=0;

    while(num != 0)
    {
        rem = num % 10;
        rev = rev * 10 + rem;
        num = num / 10;
    }

    printf("%d\n", rev);
}
```

Output:

```
Enter a number: 1234
Reverse of 1234 is 4321
```

Enter a number: 567

Reverse of 567 is 765

5. Write a C program to calculate the area of a circle using function function should receive the radius as argument and return the area

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

```
float area(float);
```

```
int main()
```

```
{
```

```
    float radius, result;
```

```
    printf("Enter radius: ");
```

```
    scanf("%f", &radius);
```

```
    result = area(radius);
```

```
    printf("Area of circle is %.2f\n", result);
```

```
    return 0;
```

```
}
```

```
float area(float r)
```

```
{
```

```
    float a;
```

```
    a = 3.14 * r * r;
```

```
    return a;
```

```
}
```

Output:

Enter radius: 5

Area of circle is 78.50

Enter radius: 7

Area of circle is 153.86

6. Write a C program to display the Fibonacci series of a given term using function function should receive the number as argument and print the series

```
#include<stdio.h>

void fibonacci(int);

int main()
{
    int n;

    printf("Enter number of terms: ");
    scanf("%d", &n);

    printf("Fibonacci series: ");
    fibonacci(n);

    return 0;
}
```

```
}
```

```
void fibonacci(int n)
{
```

```
    int i, a=0, b=1, next;
```

```
    for(i=1; i<=n; i++)
    {
```

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

```
        next = a + b;
```

```
        a = b;
```

```
        b = next;
```

```
}
```

```
    printf("\n");
```

```
}
```

Output:

Enter number of terms: 7

Fibonacci series: 0 1 1 2 3 5 8

Enter number of terms: 10

Fibonacci series: 0 1 1 2 3 5 8 13 21 34

7. Write a function to convert temperature from Celsius to Fahrenheit

```
#include<stdio.h>

float convertToFahrenheit(float);

int main()
{
    float celsius, fahrenheit;

    printf("Enter temperature in Celsius: ");
    scanf("%f", &celsius);

    fahrenheit = convertToFahrenheit(celsius);

    printf("%.2f Celsius = %.2f Fahrenheit\n", celsius,
           fahrenheit);

    return 0;
}
```

```
float convertToFahrenheit(float c)
{
    float f;
    f = (c * 9.0 / 5.0) + 32.0;
    return f;
}
```

Output:

Enter temperature in Celsius: 37
37.00 Celsius = 98.60 Fahrenheit

Enter temperature in Celsius: 0
0.00 Celsius = 32.00 Fahrenheit

8. Write a menu driven C program to addition subtraction multiplication and division to integers using function

```
#include<stdio.h>

int add(int, int);
int subtract(int, int);
int multiply(int, int);
float divide(int, int);

int main()
{
    int num1, num2, choice, result;
    float fresult;

    printf("Enter first number: ");
    scanf("%d", &num1);

    printf("Enter second number: ");
```

```
scanf("%d", &num2);

printf("\nMenu:\n");
printf("1. Addition\n");
printf("2. Subtraction\n");
printf("3. Multiplication\n");
printf("4. Division\n");
printf("Enter your choice: ");
scanf("%d", &choice);
```

```
if(choice == 1)
{
    result = add(num1, num2);
    printf("Result: %d\n", result);
}

else if(choice == 2)
{
    result = subtract(num1, num2);
    printf("Result: %d\n", result);
}
```

```
else if(choice == 3)
{
    result = multiply(num1, num2);
    printf("Result: %d\n", result);
}

else if(choice == 4)
{
    fresult = divide(num1, num2);
    printf("Result: %.2f\n", fresult);
}

else
{
    printf("Invalid choice\n");
}

return 0;
}

int add(int a, int b)
{
```

```
    return a + b;  
}
```

```
int subtract(int a, int b)  
{  
    return a - b;  
}
```

```
int multiply(int a, int b)  
{  
    return a * b;  
}
```

```
float divide(int a, int b)  
{  
    return (float)a / b;  
}
```

Output:

Enter first number: 10

Enter second number: 5

Menu:

1. Addition
2. Subtraction
3. Multiplication
4. Division

Enter your choice: 1

Result: 15

9. Write a C program to add first 7 terms of the following series using $1/!1+2/!2+3/!3\dots$

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

```
int factorial(int);
```

```
int main()
```

```
{
```

```
    int i, n=7;
```

```
    float sum=0.0;
```

```
    for(i=1; i<=7; i++) // Fixed: i<=7 instead of 1<=7
```

```
{
```

```
    sum = sum + (float)i/factorial(i);
```

```
    printf("%d/%d!+", i, i);
```

```
}
```

```
printf("\b\b = %f\n", sum);
```

```
return 0;
```

}

```
int factorial(int num)
{
    int j, fact=1;
    for(j=1; j<=num; j++)
    {
        fact = fact*j;
    }
    return fact;
}
```

Output:

$1/1! + 2/2! + 3/3! + 4/4! + 5/5! + 6/6! + 7/7! = 2.718056$

10. Write a C program to print out all Armstrong numbers between 1 and 500

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

```
int isArmstrong(int);
```

```
int main()
```

```
{
```

```
    int i;
```

```
    printf("Armstrong numbers between 1 and  
500:\n");
```

```
    for(i=1; i<=500; i++)
```

```
{
```

```
    if(isArmstrong(i) == 1)
```

```
{
```

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

```
}
```

```
}
```

```
    printf("\n");
    return 0;
}

int isArmstrong(int num)
{
    int original, remainder, result=0;

    original = num;

    while(num != 0)
    {
        remainder = num % 10;
        result = result + remainder * remainder *
remainder;
        num = num / 10;
    }

    if(result == original)
```

```
    return 1;  
else  
    return 0;  
}
```

Output:

Armstrong numbers between 1 and 500:

1 153 370 371 407

