1.wap to add two numbers using add function without passing any parameters.

```
#include <stdio.h>

void add() {
   int num1 = 5;
   int num2 = 10;
   int result = num1 + num2;
   printf("The sum of %d and %d is: %d\n", num1, num2, result);
}

int main() {
   add();
   return 0;
}
```

OUTPUT

The sum of 5 and 10 is: 15

2. wap to add two numbers using add function passing any parameters

```
#include <stdio.h>

int add(int num1, int num2) {
   return num1 + num2;
}

int main() {
   int result = add(5, 10);
   printf("The sum of 5 and 10 is: %d\n", result);
```

```
    return 0;
}
```

OUTPUT

The sum of 5 and 10 is: 15

3. Create a C program that defines a function to increment an integer by 1. The function should demonstrate call by value, showing that the original value remains unchanged.

```
#include <stdio.h>

void incrementByOne(int num) {
    num = num + 1;
    printf("Inside incrementByOne function, num = %d\n", num);
}

int main() {
    int originalNum = 5;
    printf("Before calling incrementByOne, originalNum = %d\n", originalNum);

    incrementByOne(originalNum);

    printf("After calling incrementByOne, originalNum = %d\n", originalNum);

    return 0;
}
```

OUTPUT

Before calling incrementByOne, originalNum = 5

Inside incrementByOne function, num = 6

After calling incrementByOne, originalNum = 5

4. Write a C program that attempts to swap two integers using a function that employs call by value. Show that the original values remain unchanged after the function call.

```
#include <stdio.h>


void swap(int a, int b) {
   int temp;
   temp = a;
   a = b;
   b = temp;
   printf("Inside swap function: a = %d, b = %d\n", a, b);
}


int main() {
   int x = 10, y = 20;
   printf("Before calling swap function: x = %d, y = %d\n", x, y);


   swap(x, y);


   printf("After calling swap function: x = %d, y = %d\n", x, y);


   return 0;
}
```

OUTPUT

Before calling swap function: x = 10, y = 20

Inside swap function: a = 20, b = 10

After calling swap function: x = 10, y = 20

5. Develop a C program that calculates the factorial of a number using call by value.

```
#include <stdio.h>

int factorial(int n) {
   int fact = 1;
   for (int i = 1; i <= n; i++) {
      fact *= i;
   }
   return fact;
}

int main() {
   int num = 5;
   int result = factorial(num);
   printf("The factorial of %d is: %d\n", num, result);
   return 0;
}
```

OUTPUT

The factorial of 5 is: 120

6. Create a C program that defines a function to find the maximum of two numbers using call by value.

```
#include <stdio.h>

int max(int a, int b) {
   if (a > b) {
      return a;
   } else {
```

```
    return b;
  }
}

int main() {
  int num1 = 10, num2 = 20;
  int result = max(num1, num2);
  printf("The maximum of %d and %d is: %d\n", num1, num2, result);
  return 0;
}
```

OUTPUT

The maximum of 10 and 20 is: 20

7. Problem Statement 1: Arithmetic Operations Calculator Description: Write a C program that performs basic arithmetic operations (addition, subtraction, multiplication, and division) on two numbers provided by the user. The program should use functions to perform each operation and demonstrate call by value. Requirements: Create separate functions for addition, subtraction, multiplication, and division. Each function should take two parameters (the numbers) and return the result. Use appropriate data types for the variables. Use operators for arithmetic calculations. Example Input/Output: Enter first number: 10 Enter second number: 5 Addition: 15 Subtraction: 5 Multiplication: 50 Division: 2.0

```
#include <stdio.h>

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) {
    if (b != 0) {
        return (float)a / b;
    } else {
        printf("Error! Division by zero.\n");
        return 0;
    }
}

int main() {
    int num1, num2;

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

    printf("Enter second number: ");
    scanf("%d", &num2);

    printf("Addition: %d\n", add(num1, num2));
    printf("Subtraction: %d\n", subtract(num1, num2));
    printf("Multiplication: %d\n", multiply(num1, num2));
    printf("Division: %.2f\n", divide(num1, num2));
```

```
    return 0;
}
```

OUTPUT

Enter first number: 10

Enter second number: 5

Addition: 15

Subtraction: 5

Multiplication: 50

Division: 2.00

8. Problem Statement 2: Temperature Conversion Description: Develop a C program that converts temperatures between Celsius and Fahrenheit. The program should use functions to handle the conversions and demonstrate call by value. Requirements: Create two functions: one for converting Celsius to Fahrenheit and another for converting Fahrenheit to Celsius. Each function should accept a temperature value as an argument and return the converted temperature. Use appropriate data types for temperature values. Use arithmetic operators to perform the conversion calculations. Example Input/Output: Enter temperature in Celsius: 25 Temperature in Fahrenheit: 77.0 Enter temperature in Fahrenheit: 77 Temperature in Celsius: 25.0

```
#include <stdio.h>

float celsiusToFahrenheit(float celsius) {
    return (celsius * 9/5) + 32;
}

float fahrenheitToCelsius(float fahrenheit) {
    return (fahrenheit - 32) * 5/9;
}
```

```
int main() {
   float celsius, fahrenheit;

   printf("Enter temperature in Celsius: ");
   scanf("%f", &celsius);
   printf("Temperature in Fahrenheit: %.1f\n", celsiusToFahrenheit(celsius));

   printf("Enter temperature in Fahrenheit: ");
   scanf("%f", &fahrenheit);
   printf("Temperature in Celsius: %.1f\n", fahrenheitToCelsius(fahrenheit));

   return 0;
}
```

OUTPUT

Enter temperature in Celsius: 25

Temperature in Fahrenheit: 77.0

Enter temperature in Fahrenheit: 77

Temperature in Celsius: 25.0

9. Problem Statement 3: Simple Interest Calculator Description: Develop a C program that calculates simple interest based on user input for principal amount, rate of interest, and time period. The program should use a function to compute interest and demonstrate call by value. Requirements: Implement a function that takes three parameters (principal, rate, time) and returns the calculated simple interest. Use appropriate data types for financial calculations (e.g., float or double). Utilize arithmetic operators to compute simple interest using the formula SI = P×R×T/100 Example Input/Output: Enter principal amount: 1000 Enter rate of interest: 5 Enter time period (in years): 3 Simple Interest is: 150.0

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

```
float calculateSimpleInterest(float principal, float rate, float time) {
   return (principal * rate * time) / 100;
}

int main() {
   float principal, rate, time, simpleInterest;

   printf("Enter principal amount: ");
   scanf("%f", &principal);

   printf("Enter rate of interest: ");
   scanf("%f", &rate);

   printf("Enter time period (in years): ");
   scanf("%f", &time);

   simpleInterest = calculateSimpleInterest(principal, rate, time);

   printf("Simple Interest is: %.2f\n", simpleInterest);

   return 0;
}
```

OUTPUT

```
Enter principal amount: 1000
Enter rate of interest: 5
Enter time period (in years): 3
Simple Interest is: 150.00
```

10. Create a char type variable and initialize it to value 100 2) Print the address of the above variable. 3) Create a pointer variable and store the address of the above variable 4) Perform read operation on the pointer variable to fetch 1 byte of data from the pointer 5) Print the data obtained from the read operation on the pointer. 6) Perform write operation on the pointer to store the value 65 7) Print the value of the variable defined in step 1

```
#include <stdio.h>


int main() {
   char var = 100;
   printf("Address of var: %p\n", (void*)&var);


   char *ptr = &var;
   char data = *ptr;


   printf("Data read from pointer: %d\n", data);
   *ptr = 65;


   printf("Value of var after write operation: %d\n", var);


   return 0;
}
```

OUTPUT

Address of var: 0x7ffee9c1a23b

Data read from pointer: 100

Value of var after write operation: 65

11. Write a C program that swaps the values of two integers using pointers

```c
#include <stdio.h>

void swap(int *a, int *b) {
   int temp = *a;
   *a = *b;
   *b = temp;
}

int main() {
   int num1, num2;

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

   printf("Enter second integer: ");
   scanf("%d", &num2);

   printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);

   swap(&num1, &num2);

   printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);

   return 0;
}
```

OUTPUT

```
Enter first integer: 5
```

Enter second integer: 10

Before swapping: num1 = 5, num2 = 10

After swapping: num1 = 10, num2 = 5

12. Write a C program that swaps the values of two integers using pointers (pass by refernce).

```
#include <stdio.h>

void swap(int *a, int *b) {
   int temp = *a;
   *a = *b;
   *b = temp;
}

int main() {
   int num1, num2;

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

   printf("Enter second integer: ");
   scanf("%d", &num2);

   printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);

   swap(&num1, &num2);

   printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);
```

```
    return 0;
}
```

OUTPUT

```
Enter first integer: 5
Enter second integer: 10
Before swapping: num1 = 5, num2 = 10
After swapping: num1 = 10, num2 = 5
```

13. WAP for Finding the Cube of a Number Using Pass by Reference

```
#include <stdio.h>

void cube(int *n) {
    *n = (*n) * (*n) * (*n);
}

int main() {
    int num;

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

    cube(&num);

    printf("The cube of the number is: %d\n", num);

    return 0;
}
```

OUTPUT

Enter a number: 3

The cube of the number is: 27

14. WAP to calculate the simple interest with the help of a function and pass call by reference method.

```
#include <stdio.h>

void calculateSimpleInterest(float *SI, float principal, float rate, float time) {
   *SI = (principal * rate * time) / 100;
}

int main() {
   float principal, rate, time, simpleInterest;

   printf("Enter principal amount: ");
   scanf("%f", &principal);

   printf("Enter rate of interest: ");
   scanf("%f", &rate);

   printf("Enter time period (in years): ");
   scanf("%f", &time);

   calculateSimpleInterest(&simpleInterest, principal, rate, time);

   printf("Simple Interest is: %.2f\n", simpleInterest);
```

```
    return 0;
}
```

OUTPUT

```
Enter principal amount: 1000
Enter rate of interest: 5
Enter time period (in years): 3
Simple Interest is: 150.00
```