1. 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.

ANSWER:

```
#include<stdio.h>

void incr(int num);

int main() {
   int value;
   printf("Enter the number: ");
   scanf("%d", &value);

   printf("The value before incrementing is %d\n", value);
   incr(value);
   printf("The value after incrementing (in main) is %d\n", value); // Original value remains unchanged

   return 0;
}

void incr(int num) {
   num = num + 1;
   printf("The value inside incr function after incrementing is %d\n", num); // Shows incremented value inside function
}
```

2. 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.

ANSWER:

```
#include<stdio.h>

void swap(int a,int b);

int main() {
   int a,b;
   printf("Enter the value of A: ");
   scanf("%d", &a);
   printf("Enter the value of B: ");
   scanf("%d", &b);
   printf("The value before swapping is %d,%d\n", a,b);
   swap(a,b);
   printf("The value after swapping  (in main) is %d,%d\n",a,b); // Original value remains unchanged

   return 0;
}

void swap(int a,int b) {
   int temp;
```

```
    temp=a;
    a=b;
    b=temp;
    printf("The value inside swap function after swapping is %d,%d\n", a,b); // Shows incremented value inside function
}
```

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

ANSWER:

```
#include <stdio.h>

int factorial(int n);

int main() {
    int number = 5;
    printf("Factorial of %d is %d\n", number, factorial(number));
    return 0;
}

int factorial(int n) {
    if (n == 0 || n == 1) {
        return 1;
    }
    return n * factorial(n - 1);

}
```

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

ANSWER:

```
#include <stdio.h>


int find_max(int a, int b);


int main() {

    int num1 = 15, num2 = 20;

    printf("Maximum of %d and %d is %d\n", num1, num2, find_max(num1, num2));

    return 0;

}


int find_max(int a, int b) {

    return (a > b) ? a : b;

}
```

5.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
```

ANSWER:

```
#include <stdio.h>

int add(int a, int b);

int subtract(int a, int b);

int multiply(int a, int b);

float divide(int a, int b);


int main() {
    int n1, n2;

    // Input from the user
    printf("Enter first number: ");
    scanf("%d", &n1);
    printf("Enter second number: ");
    scanf("%d", &n2);

    // Display results using call by value
    printf("Addition: %d\n", add(n1, n2));
    printf("Subtraction: %d\n", subtract(n1, n2));
    printf("Multiplication: %d\n", multiply(n1, n2));
    if (n2 != 0) {
        printf("Division: %.2f\n", divide(n1, n2));
    } else {
        printf("Division: Undefined (division by zero)\n");
    }

    return 0;
}

// Function definitions
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;
}
```

6.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
```

ANSWER:

```
#include <stdio.h>
float celsius_to_fahrenheit(float celsius);
float fahrenheit_to_celsius(float fahrenheit);

int main() {
   float tempC, tempF;
   printf("Enter temperature in Celsius: ");
   scanf("%f", &tempC);
   printf("Temperature in Fahrenheit: %.2f\n", celsius_to_fahrenheit(tempC));
   printf("Enter temperature in Fahrenheit: ");
   scanf("%f", &tempF);
   printf("Temperature in Celsius: %.2f\n", fahrenheit_to_celsius(tempF));

   return 0;
}
float celsius_to_fahrenheit(float celsius) {
   return (celsius * 9 / 5) + 32;
}
```

```
float fahrenheit_to_celsius(float fahrenheit) {

   return (fahrenheit - 32) * 5 / 9;

}
```

7. 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 脳 R 脳 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
```

ANSWER;

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

```
float calculate_simple_interest(float principal, float rate, float time);


int main() {
    float principal, rate, time;
    printf("Enter principal amount: ");
    scanf("%f", &principal);
    printf("Enter rate of interest: ");
    scanf("%f", &rate);
    printf("Enter time period (in years): ");
    scanf("%f", &time);
    printf("Simple Interest is: %.2f\n", calculate_simple_interest(principal, rate, time));


    return 0;
}



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

8.Write a C program which follows the following recquirements

A) Create a char type variable and initialize it to value 100

B) Print the address of the above variable.

C) Create a pointer variable and store the address of the above variable

D) Perform read operation on the pointer variable to fetch 1 byte of data from the pointer

E) Print the data obtained from the read operation on the pointer.

F) Perform write operation on the pointer to store the value 65

G) Print the value of the variable defined in step 1

ANSWER:

```
#include <stdio.h>


int main() {
    char ch = 100;
    printf("Address of variable ch: %p\n", (void *)&ch);
    char *ptr = &ch;
    char data = *ptr;
    printf("Data fetched from the pointer: %d\n", data);
   *ptr = 65;
    printf("Updated value of variable ch: %d\n", ch);


    return 0;
}
```

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

ANSWER:

```
#include<stdio.h>
void swap(int *a,int *b){
   int temp=*a;
   *a=*b;
   *b=temp;
}
int main(){
   int a,b;
   printf("Enter value of A:");
   scanf("%d",&a);
   printf("\nEnter value of B:");
   scanf("%d",&b);
   printf("\nBefore Swapping\nA=%d \nB=%d \n",a,b);
   swap(&a,&b);
   printf("After Swapping \nA=%d \nB=%d\n",a,b);
   return 0;

}
```

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

ANSWER:

```
#include <stdio.h>
void calculateCube(int *num) {
    *num = (*num) * (*num) * (*num);
}

int main() {
    int number;
    printf("Enter a number: ");
    scanf("%d", &number);
    calculateCube(&number);
    printf("The cube of the number is: %d\n", number);

    return 0;
}
```

11.Calculate the simple interest with the help of a function and call by reference method.

ANSWER:

```
#include <stdio.h>
void SimpleInterest(float *principal, float *rate, float *time, float *interest) {
    *interest = (*principal) * (*rate) * (*time) / 100; // Simple Interest formula
}

int main() {
    float principal, rate, time, interest;
    printf("Enter the principal amount: ");
    scanf("%f", &principal);
    printf("Enter the rate of interest: ");
    scanf("%f", &rate);
    printf("Enter the time period (in years): ");
    scanf("%f", &time);
    SimpleInterest(&principal, &rate, &time, &interest);
    printf("The simple interest is: %.2f\n", interest);

    return 0;
}
```