

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.

```
#include <stdio.h>

void increment(int n) {

    n+= 1;

    printf("Value n incremented: %d\n", n);

}

int main() {

    int n = 5;

    printf("Initial value before function call: %d\n",n);

    increment(n);

    printf("value after function call: %d\n",n);

    return 0;

}
```

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.

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

```

void swap(int a,int b) {

    a=a+b;

    b=a-b;

    a=a-b;

    printf("Swapping a=%d, b=%d\n",a,b);

}

int main() {

    int a=10,b=20;

    printf("Initial value before function call: a=%d b=%d\n",a,b);

    swap(a,b);

    printf("value after function call: a=%d b=%d\n",a,b);

    return 0;

}

```

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

```

#include <stdio.h>

void factorial(int n) {

    int fact=1;

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

```

```

        fact=fact*i;

    }

    printf("Factorial=%d\n",fact);
}

int main() {

    int n=5;

    factorial(n);

    return 0;
}

```

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

```

#include <stdio.h>

void largest(int a, int b) {
    if(a>b){
        printf("a=%d is the largest",a);
    }
    else{
        printf("b=%d is the largest",b);
    }
}

int main() {
    int a=10,b=20;
    largest(a,b);
    return 0;
}

```

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>

void add(int a,int b){

    int sum=a+b;

    printf("%d",sum);

}

void subtract(int a,int b){

    int sub=a-b;

    printf("%d",sub);

}

void multiply(int a,int b){

    int mul=a*b;

    printf("%d",mul);

}

void divison(int a,int b){
```

```
        if(b!=0) {

            int div =a/b;

            printf("%d",div);

        }

        else{

            printf("Division by zero not allowed");

        }

    }

int main(){

    int a,b;

    printf("Enter first number::\n");

    scanf("%d",&a);

    printf("Enter second number::\n");

    scanf("%d",&b);

    printf("\nADDITION::");

    add(a,b);

    printf("\nSUBTRACTION::");

    subtract(a,b);

    printf("\nMULTIPLICATION::");

    multiply(a,b);

    printf("\nDIVISION::");

    divison(a,b);

    return 0;

}
```

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>

void FtoC(float f){

    float c=(5.0/9)*(f-32);

    printf("%.2f",c);

}

void CtoF(float c){

    float f=(c*(9.0/5))+32;

    printf("%.2f",f);

}

int main(){

    float f,c;

    printf("\nEnter Temperature in Fahrenheit::");
```

```

scanf("%f",&f);

printf("\nTemperature in Celsius::");

FtoC(f);

printf("\nEnter Temperature in Celsius ::");

scanf("%f",&c);

printf("\nTemperature in Fahrenheit::");

CtoF(c);

}

```

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).

Utilise arithmetic operators to compute simple interest using the formula

$$SI = P \times R \times T / 100$$

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>

void simpleintrest(int p, int r,int t){

```

```

        int si=(p*r*t)/100;

        printf("%d",si);

    }

int main() {

    int p,r,t;

    printf("\nEnter Principle Amount::");

    scanf("%d",&p);

    printf("\nEnter interest rate::");

    scanf("%d",&r);

    printf("\nEnter time period in years::");

    scanf("%d",&t);

    printf("\nSimple Interest is::");

    simpleinterest(p,r,t);

    return 0;

}

```

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

```

#include<stdio.h>
int main() {
    int a=10,b=20;
    int *ptr=NULL;
    printf("\nInitial Values:: a=%d b=%d",a,b);
    ptr=&a;
    *ptr=*ptr+b;
    b=*ptr-b;
    *ptr=*ptr-b;
    printf("\nAfter Swapping:: a=%d b=%d",a,b);
    return 0;
}

```


6.WAP to swap the number using swap function and follow the pass by reference method.

```
#include<stdio.h>
void swap(int *,int *);
int main(){
    int a=10,b=20;
    printf("\nBefore Swapping a=%d b=%d",a,b);
    swap(&a,&b);
    printf("\nAfter Swapping a=%d b=%d",a,b);
    return 0;
}
void swap(int *p,int *q ){
    int temp=*p;
    *p=*q;
    *q=temp;
}
```

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

```
#include <stdio.h>
int Cube(int *n) {
    int c = (*n) * (*n) * (*n);
    return c;
}

int main() {
    int n=6;
    int c=Cube(&n);
    printf("The cube of the number is: %d\n", c);

    return 0;
}
```

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

```
#include<stdio.h>
int simpleinterest(int *a,int *b,int *c){
    int si=((*a)*(*b)*(*c))/100;
    return si;
}
```

```
}  
int main(){  
    int p,r,t,si;  
    printf("\nEnter principle amount::");  
    scanf("%d",&p);  
    printf("\nEnter rate of interest::");  
    scanf("%d",&r);  
    printf("\nEnter time period::");  
    scanf("%d",&t);  
    si=simpleinterest(&p,&r,&t);  
    printf("\nSimple Interest::%d",si);  
}
```