/*Function Implelemtation
==========
1. Function Prototype (Function Declaration)
    which is defined before the main() function
    Syntax: return_type function_name(If you are passing paraemter, mention the datatypes of the parameter);
2. Function call
    This is implmented inside the main() function
    Syntax: function_name(pass the varible only the variable names);
3. Function Definition
    This is implemented after the main() Function
    Syntax:
    return_type function_name(If you are passing paraemter, mention the datatypes of the parameter){
        ========
        Function Body
        ========
    }
*/
//WAP to add two number using the add function by parameter and the function is not going to return
//any Data
#include <stdio.h>
void add_num(int , int);
int main(){
  int a = 10, b = 20;
  printf("001a = %p\n",&a);
  printf("001b = %p\n",&b);
  add_num(a,b);
  printf("The values of a and b is %d , %d",a,b);
  return 0;
}
void add_num(int a, int b){
  a = 40;
  b = 50;
  printf("002a = %p\n",&a);
  printf("002b = %p\n",&b);
  int sum =0;
  sum = a + b;
  printf("Sum = %d \n",sum);
---

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>
int increment(int);
int main(){
   int a = 10;
   int value=increment(a);
   printf("incremented value is %d\n",value);
   printf("original value is %d\n",a);
    return 0;
}

int increment(int c){

   c++;
   return c;
}
```

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,int);
int main(){
   int a=10,b=11;
   printf("before swapping a=%d b=%d\n",a,b);
   swap(a,b);
   printf("after swapping a=%d b=%d\n",a,b);

    return 0;
}
void swap(int a,int b){
  int temp=0;
  temp=a;
  a=b;
  b=temp;
  printf("in function a=%d b=%d\n",a,b);
}
```

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

```
#include <stdio.h>
int factorial(int a);
int main()
{
   int num;
   printf("enter the number");
   scanf("%d",&num);
   printf("factorial of %d is %d",num,factorial(num));
   return 0;
}
int factorial(int a){
    int fact=1;
   if(a==0||a==1){
      return 1;
   }
   else{
      for(int i=2;i<=a;i++){
```

```
    fact=fact*i;

    }
    return fact;
  }
}
```

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,int);
int main()
{
  int a=8,b=10;
  int max_value=max(a,b);
  return 0;
}
int max(int a,int b){
  if(a>b){
    printf("a is largest");
    return a;
  }
  else{
    printf("b is largest");
    return b;
  }
}
```

---

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

```c
#include <stdio.h>
int add(int,int);
int sub(int,int);
int mul(int,int);
float divi(int,int);
int main()
{  int num1,num2,sum,diff,pdt;
   float qua;
   printf("enter the 2 numbers\n");
   scanf("%d %d",&num1,&num2);
   sum=add(num1,num2);
   diff=sub(num1,num2);
   pdt=mul(num1,num2);
   qua=divi(num1,num2);
   printf("after addition %d\n",sum);
   printf("after subtraction %d\n",diff);
   printf("after multiplication %d\n",pdt);
   printf("after division %f\n",qua);
   return 0;
}
int add(int a,int b){
   return a+b;
}
int sub(int a,int b){
   return a-b;
}
int mul(int a,int b){
   return a*b;
}
float divi(int a,int b){
   return a/b;
}
```

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 );
float fahrenheitToCelsius(float);
int main() {
  float temp, result;
  int i;

  printf("1. Convert Celsius to Fahrenheit\n2. Convert Fahrenheit to Celsius\n");
  scanf("%d", &i);
  if (i == 1) {
    printf("Enter temperature in Celsius: ");
    scanf("%f", &temp);
    result = celsiusToFahrenheit(temp);
    printf("Temperature in Fahrenheit: %f\n", result);
  } else if (i == 2) {
    printf("Enter temperature in Fahrenheit: ");
    scanf("%f", &temp);
    result = fahrenheitToCelsius(temp);
    printf("Temperature in Celsius: %f\n", result);
  } else {
    printf("Invalid action!\n");
  }

  return 0;
}
float celsiusToFahrenheit(float celsius) {
  return (celsius * 9.0 / 5.0) + 32.0;
}
float fahrenheitToCelsius(float fahrenheit) {
  return (fahrenheit - 32.0) * 5.0 / 9.0;
}
```

Problem Statement 2: 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 SimpleInterest(float,float,float);
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 = SimpleInterest(principal, rate, time);

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

   return 0;
}
float SimpleInterest(float principal, float rate, float time) {
   return (principal * rate * time) / 100.0;
}
```

POINTER

```
#include <stdio.h>

void main() {
   char value = 100;
   printf("%p\n",&value);
   char *pvalue;
   pvalue=&value;
   char fetched_value=*pvalue;
   printf("%c\n",fetched_value);
   *pvalue=65;
   printf("value after write %p\n",pvalue);
   printf("%c\n",value);

}
```

```
#include <stdio.h>

void main() {
   int number =0;
   int *pnumber=NULL;
   number =10;
   printf("numbers address is %p\n",&number);
   printf("numbers value is %d\n",number);
   pnumber=&number;
   printf("pnumbers address is %p\n",&pnumber);
   printf("pnumbers address is %d\n",sizeof(pnumber));
   printf("pnumber value is %p\n",pnumber);
   printf("value pointed to is%d\n",*pnumber);
}
```

```
// swapping two number suing pointers
```

```
#include <stdio.h>
//swapping
void main() {
   int num1=10,num2=20;
   int *pnum1,*pnum2;
   pnum1=&num1;
   pnum2=&num2;
   int temp;
   printf("before swapping num1=%d num2=%d\n",*pnum1,*pnum2);
   temp=*pnum1;
   *pnum1=*pnum2;
   *pnum2=temp;
   printf("after swapping num1=%d num2=%d",*pnum1,*pnum2);
}
```

```
//swapping 2 numbers using pointers and call by reference
#include <stdio.h>
void swap(int*,int*);
void main() {
   int num1=10,num2=20;
   printf("before swapping num1=%d num2=%d\n",num1,num2);
   swap(&num1,&num2);
   printf("after swapping num1=%d num2=%d",num1,num2);
}
void swap(int *pnum1,int *pnum2){
   int temp;
   temp=*pnum1;
   *pnum1=*pnum2;
   *pnum2=temp;

}
```

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

```
// swapping two number susing pointers
#include <stdio.h>
int cube(int*);
void main() {
   int num=20;

   int ans=cube(&num);

   printf("cube of %d:%d",num,ans);
}
int cube(int *a){
   return *a* *a* *a;

}
```

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

```
 #include <stdio.h>
float SimpleInterest(float*,float*,float*);
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 = SimpleInterest(&principal, &rate, &time);

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

   return 0;
}
float SimpleInterest(float *p, float *r, float *t) {
   return (*p * *r * *t) / 100.0;
}
```
