

/\*Function Implemmentation

=====

1. Function Prototype (Function Declaration)

which is defined before the main() function

Syntax: return\_type function\_name(If you are passing parameter, mention the datatypes of the parameter);

2. Function call

This is implemented inside the main() function

Syntax: function\_name(pass the variable only the variable names);

3. Function Definition

This is implemented after the main() Function

Syntax:

return\_type function\_name(If you are passing parameter, mention the datatypes of the parameter){

=====

Function Body

=====

}

\*/

**1.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);
}
```

**2. 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 inc_value(int num);

int main()
{
    int num;

    printf("Enter a number:");

    scanf("%d",&num);

    inc_value(num);

    printf("After function call value of num is %d\n",num);
}

void inc_value(int num)
{
    num += 1;

    printf("Incremented value is %d\n",num);
}
```

**3. 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 num1,num2;

    printf("Enter two numbers:");
```

```

scanf("%d %d",&num1,&num2);

printf("Before swapping num1= %d\nnum2 =%d\n",num1,num2);

swap(num1,num2);

printf("After function call value of \n num1= %d\nnum2 = %d\n",num1,num2);
}

void swap(int num1,int num2)
{
    int temp = num1;

    num1 = num2;

    num2 = temp;

    printf("After swapping num1= %d\nnum2 =%d\n",num1,num2);
}

```

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

```

#include <stdio.h>

void factorial(int num);

int main()
{
    int num;

    printf("Enter number:");

    scanf("%d",&num);

    factorial(num);
}

void factorial(int num)
{
    int fact =1;

    for(int i=1;i<=num;i++)

```

```
{  
    fact *= i;  
}  
  
printf("factorial is %d",fact);  
}
```

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

```
#include <stdio.h>  
  
void max(int num1,int num2);  
  
int main()  
{  
    int num1,num2;  
  
    printf("Enter two numbers:");  
  
    scanf("%d %d",&num1,&num2);  
  
  
    max(num1,num2);  
  
}  
  
void max(int num1,int num2)  
{  
    if(num1 > num2)  
    {  
        printf("%d is maximum",num1);  
    }  
  
    else if(num1 < num2)  
    {  
        printf("%d is maximum",num2);  
    }  
}
```

```
}  
  
else  
  
{  
  
    printf("Both are equal");  
  
}  
  
}
```

### **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 sum(float num1,float num2);
```

```
int sub(float num1,float num2);
```

```

float mul(float num1,float num2);

float divi(float num1,float num2);

int main()
{
    float num1,num2;

    printf("Enter 1st number:");

    scanf("%f",&num1);

    printf("Enter 2nd number:");

    scanf("%f",&num2);

    int sum_a = sum(num1,num2);

    int sub_a = sub(num1,num2);

    float mul_a = mul(num1,num2);

    float divi_a = divi(num1,num2);

    printf("Addition:%d\n",sum_a);

    printf("Subtraction:%d\n",sub_a);

    printf("Multiplication:%.1f\n",mul_a);

    printf("Division:%.1f\n",divi_a);


}

int sum(float num1,float num2)

{

    int sum = num1+num2;

    return sum;

}

int sub(float num1,float num2)

```

```
{  
  
    int sub = num1-num2;  
  
    return sub;  
  
}
```

```
float mul(float num1,float num2)
```

```
{  
  
    float mul = num1*num2;  
  
    return mul;  
  
}
```

```
float divi(float num1,float num2)
```

```
{  
  
    float divi = num1/num2;  
  
    return divi;  
  
}
```

## **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 convert_fahren(float temp);
```

```
float convert_celcius(float temp);
```

```
int main()
```

```
{
```

```
    float temp1,temp2;
```

```
    printf("Enter temperature in Celsius:");
```

```
    scanf("%f",&temp1);
```

```
    float fahren_temp = convert_fahren(temp1);
```

```
    printf("\nTemperature in Fahrenheit:%.1f",fahren_temp);
```

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

```
    scanf("%f",&temp2);
```

```
    float celcius_temp = convert_celcius(temp2);
```

```
    printf("\nTemperature in Celcius:%.1f",celcius_temp);
```

```
}
```

```
float convert_fahren(float temp)
```

```
{
```

```
    float fahren_temp = (temp * (9.0/5.0)) + 32.0;
```

```
    return fahren_temp;
```



```

}

float convert_celcius(float temp)

{

    float celcius_temp = (temp-32.0) * (5.0/9.0);

    return celcius_temp;

}

```

### **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

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

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>
```

```

float calculateSimpleInterest(float principal, float rate, float time)

{

    return (principal * rate * time) / 100;

}

```

```

}

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

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

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

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

    interest = calculateSimpleInterest(principal, rate, time);

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

    return 0;
}

```

#### **Problem statement 4**

- 1) 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

```
int main()

{

    char var = 100;

    printf("Address of var = %p\n",&var);

    char *pVar = &var;

    char read_data = *pVar;

    printf("The data obtained from read operation is %c\n",read_data);

    *pVar=65;

    printf("Updated value is %c\n",var);

    return 0;

}
```

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

```
#include <stdio.h>

int main()

{

    int num1=10,num2=20;

    int *p_num1 = &num1;

    int *p_num2 = &num2;

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

    int temp = *p_num1;
```

```
*p_num1 = *p_num2;

*p_num2 = temp;

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

return 0;

}
```

### **Example:**

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

```
int main()

{

    long num1 = 0;

    long num2 = 0;

    long *pNum = NULL;


    pNum = &num1;

    *pNum = 2;

    ++num2;

    num2 += *pNum;


    pNum = &num2;

    ++*pNum;


    printf("num1: %ld, num2: %ld, *pNum: %ld, *pNum + num2 = %ld\n" , num1,
num2, *pNum, *pNum + num2 );

    return 0;

}
```

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

```
#include<stdio.h>

int swap(int *,int *);

int main()
{
    int a=20,b=30;

    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;
}

int swap(int *p,int *q)
{
    int temp = *p;

    *p=*q;

    *q=temp;

    return 0;
}
```

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

```
#include <stdio.h>

void cube(int *ptr)
{
    *ptr=(*ptr) * (*ptr) * (*ptr);
}
```

```
}  
  
int main()  
  
{  
  
    int num;  
  
    printf("Enter a number: ");  
  
    scanf("%d", &num);  
  
    cube(&num);  
  
    printf("Cube is %d", num);  
  
}
```

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

```
#include <stdio.h>  
  
void calculateSimpleInterest(float *principal, float *rate, float *time, float *interest)  
  
{
```

```
    *interest = (*principal * *rate * *time) / 100;
}

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);

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

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

    return 0;
}
```