

## Day8\_assignment\_Abhirami

**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 main() {

    int integer=10;

    increment( integer);

    printf("integer=%d\n",integer);

    return 0;

}

void increment(int a) {

    a++;

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

}
```

**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, int);

int main() {

    int a = 10, b= 20;

    printf("Before swap inside main function: a = %d, b = %d\n", a, b);
```

```

swap(a, b);

printf("After swap inside main function: a = %d, b = %d\n", a, b);

return 0;

}

void swap(int a, int b) {

printf("Before swap inside swap function: a = %d, b = %d\n", a, b);

a = a + b;

b = a - b;

a = a - b;

printf("After swap inside swap function: a=%d, b = %d\n", a, b);

}

```

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

```

int fact(int);

#include <stdio.h>

int main() {

printf("Enter a number: ");

int num;

scanf("%d", &num);

int res = fact(num);

printf("Factorial of %d is %d", num, res);

return 0;

}

int fact(int num) {

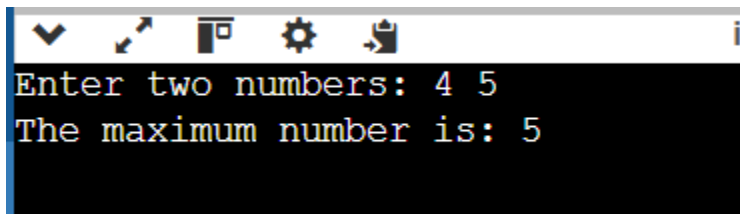
```

```
if (num == 0 || num == 1){  
    return 1;  
}  
else{  
    return num * fact(num-1);  
}  
}
```

**4. 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, int);  
  
int main() {  
    int num1, num2;  
  
    printf("Enter two numbers: ");  
  
    scanf("%d %d", &num1, &num2);  
  
    max(num1, num2);  
  
    return 0;  
}  
  
void max(int num1, int num2) {  
    int result;  
  
    if (num1 > num2) {  
        result = num1;
```

```
}  
  
else {  
  
result = num2;  
  
}  
  
printf("The maximum number is: %d\n", result);  
  
}
```



```
Enter two numbers: 4 5  
The maximum number is: 5
```

### 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, int);

int subtract(int, int);

int multiply(int, int);

float divide(int, int);

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));

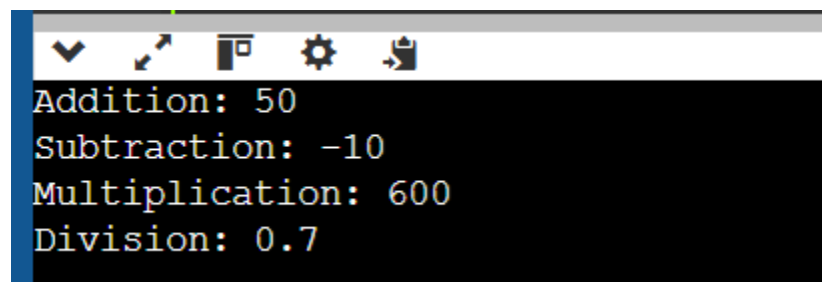
    printf("Division: %.1f\n", divide(num1, num2));

}

int add(int num1, int num2) {

    return num1 + num2;
```

```
}  
  
int subtract(int num1, int num2) {  
    return num1 - num2;  
}  
  
int multiply(int num1, int num2)  
{  
    return num1 * num2;  
}  
  
float divide(int num1, int num2){  
    if(0==num2) {  
        printf("Error: Division by zero!\n");  
    return 0.0;  
    }  
    else  
    {  
    return (float)num1/num2;  
    }  
}
```



```
✓ ↗ □ ⚙ 📄  
Addition: 50  
Subtraction: -10  
Multiplication: 600  
Division: 0.7
```

## 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 c_to_f(float); // Function prototype for Celsius to Fahrenheit
```

```
float f_to_c(float); // Function prototype for Fahrenheit to Celsius
```

```
int main() {
```

```
    float c, f;
```

```
    // Celsius to Fahrenheit conversion
```

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

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

float res1 = c_to_f(c); // Corrected syntax

printf("Temperature in Fahrenheit: %.1f\n", res1);

printf("\n");

// Fahrenheit to Celsius conversion

printf("Enter temperature in Fahrenheit: ");

scanf("%f", &f);

float res2 = f_to_c(f); // Corrected syntax

printf("Temperature in Celsius: %.1f\n", res2);


return 0;
}


float c_to_f(float c) {

    return (c * 9 / 5) + 32; // Added multiplication operator
}


float f_to_c(float f) {

    return (f - 32) * 5 / 9; // Corrected formula
}
```



## 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

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

### 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 simple_interest(float, float, int); // Function prototype
```

```
int main() {  
    float P, R;  
    int T;
```

```
    // Input principal, rate of interest, and time
```

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

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

```
    printf("Enter time period (in years): ");
```

```

scanf("%d", &T);

// Calculate simple interest
float SI = simple_interest(P, R, T); // Corrected syntax
printf("Simple Interest is: %.1f\n", SI);

return 0;
}

// Function to calculate simple interest
float simple_interest(float P, float R, int T) {
    float SI = (P * R * T) / 100; // Corrected formula
    return SI;
}

```

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

```

#include <stdio.h>

int main() {
    char A = 100; // Declare and initialize variable A
    printf("Address of A is %p\n", (void *)&A); // Print the address of A

    char *pA = &A; // Pointer pA points to the address of A
    printf("Value of A is %c\n", A); // Print the current value of A

    char B = *pA; // Assign the value pointed to by pA (i.e., value of A) to B
    printf("Value of B is %c\n", B); // Print the value of B

    printf("New value of A is %c\n", A); // Print the current value of A (unchanged at this
    point)

    *pA = 65; // Change the value at the address pointed to by pA to 65 (ASCII value of
    'A')
    printf("Updated value of A is %c\n", A); // Print the updated value of A

    return 0;
}

```

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

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

```
// Function to swap two numbers using pointers
```

```
void swap(int *p, int *q) {  
    printf("Before swap inside swap function: a = %d, b = %d\n", *p, *q);  
  
    // Swapping logic  
    int temp = *p; // Store value of *p in temp  
    *p = *q;       // Assign value of *q to *p  
    *q = temp;     // Assign value of temp to *q  
  
    printf("After swap inside swap function: a = %d, b = %d\n", *p, *q);  
}
```

```
int main() {  
    int a = 10, b = 20;  
  
    printf("Before swap inside main function: a = %d, b = %d\n", a, b);  
    swap(&a, &b); // Passing addresses of a and b to the swap function  
    printf("After swap inside main function: a = %d, b = %d\n", a, b);  
  
    return 0;  
}
```

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

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

```
// Function to calculate the cube using pass by reference
```

```
int cube(int *i) {  
    return (*i) * (*i) * (*i); // Dereference the pointer to access the value  
}  
  
int main() {  
    int num;  
  
    printf("Enter a number: ");
```

```

scanf("%d", &num); // Read the number

int res = cube(&num); // Pass the address of `num` to the function
printf("The cube of %d is: %d\n", num, res); // Print the result

return 0;
}

```

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

```

#include <stdio.h>

// Function to calculate simple interest using call by reference
float simple_interest(float *a, float *b, int *c) {
    return ((*a) * (*b) * (*c)) / 100; // Calculate simple interest
}

int main() {
    float P, R;
    int T;

    // Input values
    printf("Enter principal amount: ");
    scanf("%f", &P);

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

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

    // Calculate simple interest
    float SI = simple_interest(&P, &R, &T);

    // Output result
    printf("Simple Interest is: %.1f\n", SI);

    return 0;
}

```

