

**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_integer(int);

int main(){
    int a=10;

    increment_integer(a);

    printf("value of a= %d\n",a);
}
```

```
void increment_integer(int a){

    a++;

    printf("value of a=%d\n",a);
}
```

Output

value of a=11

value of a= 10

**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_values(int,int);

int main(){
    int a=10;

    int b=20;

    swap_values(a,b);

    printf("value of a=%d\nvalue of b=%d\n",a,b);
}
```

```
    return 0;

}

void swap_values(int a,int b){
    int temp;
    printf("before swapping\na=%d\nb=%d\n",a,b);
    temp=a;
    a=b;
    b=temp;
    printf("After swapping\na=%d\nb=%d\n",a,b);
}
```

#### OUTPUT

before swapping

a=10

b=20

After swapping

a=20

b=10

value of a=10

value of b=20

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

```
#include<stdio.h>

void factorial(int);

int main(){
    int a=10;
    factorial(a);
    return 0;
}
```

```

void factorial(int n){
    int fact=1;
    for(int i=1;i<=5;i++){
        fact=fact*i;
    }
    printf(" factorial of %d= %d",n,fact);
}

```

#### OUTPUT

factorial of 10= 120

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

```

#include<stdio.h>
void calc_maximum(int,int);
int main(){
    int a=100,b=200;

    calc_maximum(a,b);
    return 0;
}
void calc_maximum(int num1,int num2){
    if(num1>num2){
        printf("%d is greater than %d",num1,num2);
    }else{
        printf("%d is greater than %d",num2,num1);
    }
}

```

#### OUTPUT

200 is greater than 100

## **5. Problem Statement : 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>

float addition(float a,float b){
    return a+b;
}

float subtraction(float a,float b){
    return a-b;
}

float multiplication(float a,float b){
    return a*b;
}

float division(float a,float b){
    if(b!=0){
```

```

        return a/b;
    }else{
        printf("cannot devide by zero\n");
        return 1;
    }
}

int main(){
    float num1,num2;
    printf("Enter First Number:\n");
    scanf("%f",&num1);
    printf("Enter Second Number:\n");
    scanf("%f",&num2);

    printf("Addition=%.0f\n",addition(num1,num2));
    printf("Subtraction=%.0f\n",subtraction(num1,num2));
    printf("Multiplication=%.0f\n",multiplication(num1,num2));
    if(num2!=0){
        printf("Division=%.2f\n",division(num1,num2));
    }
}

```

### OUTPUT

Enter First Number:

10

Enter Second Number:

5

Addition=15

Subtraction=5

Multiplication=50

Division=2.00

## **6.Problem Statement : 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 cels_farnht(float);
```

```
void farnht_cels(float);
```

```
void cels_farnht(float c_temp){
```

```
    float f_temp=(c_temp*9/5)+32;
```

```
    printf("%.2f celcius temperature in faranhit= %.2f\n",c_temp,f_temp);
```

```
}
```

```
void farnht_cels(float f_temp){
```

```
    float c_temp = (f_temp - 32)*5/9;
```

```
    printf("%.2f faranhit temparature in celcious= %.2f\n",f_temp,c_temp);
```

```
}
```

```

int main(){
    float celcius,faranhit;

    printf("Enter temperature in celcius: ");
    scanf("%f",&celcius);
    cels_farnht(celcius);

    printf("Enter temperature in faranhit: ");
    scanf("%f",&faranhit);
    farnht_cels(faranhit);

    return 0;
}

```

#### OUTPUT

```

Enter temperature in celcius: 25
25.00 celcius temperature in faranhit= 77.00
Enter temperature in faranhit: 77
77.00 faranhit temperature in celcius= 25.00

```

### **7.Problem Statement : 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>
```

```
void simple_interest(int,int,int);
```

```
void simple_interest(int p,int r,int t){
```

```
    float SI=p*r*t/100;
```

```
    printf("Simple interest = %.2f",SI);
```

```
}
```

```
int main(){
```

```
    int amount,interest,time;
```

```
    printf("Enter principal amount: ");
```

```
    scanf("%d",&amount);
```

```
    printf("Enter rate of interest: ");
```

```
    scanf("%d",&interest);
```

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

```
    scanf("%d",&time);
```

```
    simple_interest(amount,interest,time);
```

```
    return 0;
```

```
}
```



## OUTPUT

Enter principal amount: 1000

Enter rate of interest: 5

Enter time period (in years): 3

Simple interest = 150.00

## **8. EXERCISE**

**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.**

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

```
int main(){
```

```
    char NUM=100;
```

```
    printf("Adress of variable NUM = %p\n",&NUM);
```

```
    char *pNUM = &NUM;
```

```
    printf("1 Byte data in the pointer *pNUM = %p\n",pNUM);
```

```
    *pNUM=*pNUM-35;
```

```
    printf("Value of Num= %d\n",NUM);
```

```
}
```

## OUTPUT

Adress of variable NUM = 0061FF1B

1 Byte data in the pointer \*pNUM = 0061FF1B

Value of Num= 65

## **9. POINTER EXERCISE**

```
#include<stdio.h>

int main(){
    int number = 0;
    int *pnumber = NULL;

    number = 10;
    printf("number's address: %p\n",&number);
    printf("number's value: %d\n",number);

    pnumber=&number;
    printf("pnumber's address: %p\n",&pnumber);
    printf("pnumber's size: %d bytes\n",sizeof(pnumber));
    printf("pnumber's value: %p\n",pnumber);
    printf("value pointed to: %d",*pnumber);

    return 0;
}
```

### **OUTPUT**

```
number's address: 0061FF1C
number's value: 10
pnumber's address: 0061FF18
pnumber's size: 4 bytes
pnumber's value: 0061FF1C
value pointed to: 10
```

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

```
#include<stdio.h>

int main(){
    int a=10;
    int b=20;

    printf("value before swapping:\na = %d\nb = %d\n",a,b);

    int *pa=&a;
    int *pb=&b;

    *pa=*pa+*pb; //30
    *pb=*pa-*pb; //10
    *pa=*pa-*pb;

    printf("value after  swapping using pointers:\na = %d\nb = %d\n",a,b);
    return 0;

}
```

**OUTPUT**

value before swapping:

a = 10

b = 20

value after swapping using pointers:

a = 20

b = 10

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

```
#include<stdio.h>

int find_cube(int *);

int main(){
    int num=3;
    find_cube(&num);
    printf("cube=%d",find_cube(&num));
    return 0;
}

int find_cube(int *a){
    return *a * *a * *a;
}
```

#### OUTPUT

cube=27

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

```
#include<stdio.h>

void swap_numbers(int *,int *);

int main(){
    int num1=20;
    int num2=30;

    printf("Before swapping:\nnum1 = %d\nnum2 = %d\n ",num1,num2);
    swap_numbers(&num1,&num2);
    printf("After swapping:\nnum1 = %d\nnum2 = %d\n ",num1,num2);

}
```

```

void swap_numbers(int *a,int *b){

    *a=*a+*b;

    *b=*a-*b;

    *a=*a-*b;

}

```

### OUTPUT

Before swapping:

num1 = 20

num2 = 30

After swapping:

num1 = 30

num2 = 20

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

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

```
float simple_interest(float *,float *, float *);
```

```

int main(){

    float amount,interest,time,SI;

    printf("Enter principal amount: ");

    scanf("%f",&amount);

    printf("Enter rate of interest: ");

    scanf("%f",&interest);

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

    scanf("%f",&time);


    simple_interest(&amount,&interest,&time);

    return 0;
}

```

```
}
```

```
float simple_interest(float *p, float *r, float *t){
```

```
    float si=(*p * *r * *t)/100;
```

```
    printf("Simple interest = %.2f", si);
```

```
}
```

#### OUTPUT

Enter principal amount: 1000

Enter rate of interest: 5

Enter time period (in years): 3

Simple interest = 150.00