

DAY 8 ASSIGNMENT

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 n=5,val=0;
    printf("the value od v is %d\n",n);
    val= increment(n);
    printf("the incremented value is %d\n",val);
    return 0;
}
int increment (int v){
    v=v+1;
    return v;
}
```

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>

int swap(int,int);
int main() {
    int n = 5, m = 4;

    printf("Before swap function call: n = %d, m = %d\n", n, m);
    swap(n, m);
    printf("After swap function call: n = %d, m = %d\n", n, m);

    return 0;
}
int swap(int x, int y) {
    int temp = x;
    x = y;
    y = temp;
}
```

```

    printf("Inside swap function: x = %d, y = %d\n", x, y);
}

```

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

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

```

int swap(int);
int main() {
    int result;
    int n = 5;
    result = swap(n);
    printf("factorial of %d is %d",n,result);

```

```

    return 0;
}
int swap(int n) {
    int fact =1;
    for(int i =n;i>0;i--){
        fact =fact*i;
    }
    return fact;
}

```

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

```

#include<stdio.h>
int maxvalue (int ,int);
int main(){
    int n=3,m=7,max;
    max = maxvalue(n,m);
    printf("the heighest value is %d",max);

}
int maxvalue (int a,int b){
    if(a>b){

```

```

        return a;
    }else{
        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

```

```

#include<stdio.h>
int add(int,int);
int sub(int,int);
int mult(int,int);
float div(float,float);
int main(){
    int n,m ,result;
    float result1;
    printf("enter the number");
    scanf("%d",&n);
    printf("enter the second number");
    scanf("%d",&m);
    result = add(n,m);
    printf("Addition %d\n",result);
    result = sub(n,m);

```

```

    printf("subtraction %d\n",result);
    result = mult(n,m);
    printf("multiplication %d\n",result);
    result1 = div(n,m);
    printf("divition %.2f\n",result1);

}
int add (int a,int b){
    int result =a+b;
    return result;

}
int sub (int a,int b){
    int result =a-b;
    return result;

}
int mult (int a,int b){
    int result =a*b;
    return result;

}
float div (float a,float b){

    float result =a/b;
    return result;

}

```

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 fahrenheit(float);
int main(){
    float n,m,result;
    printf("enter the temperature in celsius");
    scanf("%f",&n);
    result= celsiustofahrenheit(n);
    printf("converting celsius in to fahrenheit %.2f\n",result);
    printf("enter the temperature in Fahrenheit");
    scanf("%f",&m);
    result = fahrenheit(m);
    printf("converting  fahrenheit in to celsius  %.2f",result);
}
float celsiustofahrenheit(float a){
    float result ;
    result =(a*9/5)+32;
}
float fahrenheit(float a){
    float result ;
    result =(a-32)*5/9;
}
```

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 interest(int,float,int);
int main(){
    int n,p;
    float final,m;
    printf("enter the principle amount");
    scanf("%d",&n);
    printf("enter the rate of interest");
    scanf("%f",&m);
    printf("enter the time period in an year");
    scanf("%d",&p);
    final = interest(n,m,p);
    printf("the simple interest is %.2f",final);
}
float interest (int a,float b,int c){
    float result ;
    result = (a*b*c)/100;
    return result;
}
```

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

```
#include<stdio.h>
int main(){
    int num1,num2,temp;
    int *ptr1;
    int *ptr2;
    num1=20;
```

```

num2=30;
printf("the values before swaping num1 :%d num2:%d\n",num1,num2);
ptr1 = &num1;
ptr2 = &num2;

temp=*ptr1;
*ptr1=*ptr2;
*ptr2=temp;

printf("the value after swaping num1 :%d num2: %d\n",num1,num2);

}

```

```

#include<stdio.h>
int swap(int *, int *);
int main(){
    int a=30,b=40,new;
    printf("the value before swapping a:%d, b:%d\n",a,b);
    new=swap(&a,&b);
    printf("the swaped value a: %d,b: %d\n",a,b);
}
int swap(int *a,int *b){
    int temp;
    temp=*a;
    *a=*b;
    *b=temp;
}

```

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

```

#include<stdio.h>
int cube(int*);
int main(){
    int n=3,result;
    result = cube(&n);
    printf("the cube of the value %d is %d",n,result);
}

```

```

}
int cube(int *a){
    int v;
    v= (*a)*(*a)*(*a);
    return v;
}

```

SIMPLE INTEREST CALCULATOR

```

#include<stdio.h>
float simple(int *,int *,float *);
int main(){
    int salary=1000,year=5;
    float result ,interest=5.5;
    result=simple(&salary,&year,&interest);
    printf("the simple interest accodinding to %d salary %d year %.2f
interest===%.2f",salary,year,interest,result);

}
float simple(int *s,int *y,float*i){
    float inter;
    inter =(s)*(y)*(i)/100;
    return inter;

}

```

CLASS WORK

```

variables
#include<stdio.h>
int main(){
    int n=10;
    int ar[n];
    for(int i=0;i<10;i++){
        scanf("%d",&ar[i]);
    }
    for(int i=0;i<10;i++){
        printf("%d",ar[i]);
    }
}
-----

```



```

#include<stdio.h>
int main(){
    int n=3;
    int ar[n][n];
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            scanf("%d",&ar[i][j]);
        }
    }
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            printf("%d->",ar[i][j]);
        }
    }
}

```

function

```

#include<stdio.h>
int add(int,int);
int main(){
    int n,m,total;
    printf("enter a number");
    scanf("%d",&n);
    printf("enter a number");
    scanf("%d",&m);
    total= add(n,m);
    printf("%d",total);
    return 0;
}
int add(int a,int b){
    int v;
    v=a+b;
    return v;
}

```

```

#include<stdio.h>
void add (void);
int main(){

```

```

    add();
    return 0;
}
void add(){
    int a=10,b=20,sum=0;
    sum=a+b;
    printf("%d",sum);
}

```

```

#include<stdio.h>
void add (int,int);
int main(){
    int a=10,b=20;
    add(a,b);
    return 0;
}
void add(int c,int d){
    int sum =0;
    sum=c+d;
    printf("%d",sum);
}

```

```

#include<stdio.h>
void add (int,int);
int main(){
    int a=10,b=20;
    printf("001a=%p\n",&a);
    printf("001b=%p\n",&b);
    add(a,b);
    printf("the value of a and b is %d,%d",a,b);
    return 0;
}
void add(int a,int b){
    a=50;
    b=70;
    printf("002a=%p\n",&a);
    printf("001b=%p\n",&b);
    int sum =0;
    sum=a+b;
    printf("%d",sum);
}

```

```
}
```

```
-----  
  
#include<stdio.h>  
int add (int,int);  
int main(){  
    int a=10,b=20,sum;  
    sum = add(a,b);  
    printf("the sum is %d",sum);  
    return 0;  
}  
int add(int a,int b){  
    int sum;  
    sum= a+b;  
    return sum;  
  
}
```

POINTER

```
-----  
  
#include<stdio.h>  
int main(){  
    int a;  
    int *p;  
    p=&a;  
    *p = 20;  
    printf("a=%d\n",a);  
    printf("address of a = %p\n",&a);  
    printf("address of *p=%p\n",&p);  
    printf("**p = %p\n",p);  
  
    return 0;  
}
```

```
-----  
  
#include<stdio.h>  
int main(){  
    int A=10;  
    printf("001A=%d\n",A);  
    int *PA = &A;
```

```
A=*PA + 5;  
printf("002A=%d",A);
```

```
}
```

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

```
int main(){  
    int count =10,x;  
    int *pcount = &count;  
    x=*pcount;  
  
    printf("count =%d, x=%d",count,x);  
    printf("address of pcount =%p",&pcount);  
    printf("size of pcount %d",sizeof(pcount));
```

```
}
```

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

```
int main(){  
    char ch =100,ch2;  
    char *pch;  
    pch = &ch;  
  
    printf("%p\n",ch);  
    printf("%d\n",*pch);  
    ch2 = *pch;  
    printf("%d\n",ch2);  
    *pch =65;  
    printf("%d",ch);
```

```
}
```

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

```
int main(){  
    int number =0;  
    int *pnumber=NULL;  
    number=10;  
    printf("numbers address :%p\n",&number);  
    printf("number value :%d\n",number);  
    pnumber=&number;  
    printf("pnumbers address:%p\n",&pnumber);  
    printf("pnumber size :%d bytes\n",sizeof(pnumber));  
    printf("pnumber value:%p\n",pnumber);
```

```
    printf("value pointer to :%d\n",*pnumber);  
    return 0;  
}
```

```
-----  
#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=%1d num2=%1d *pnum=%1d *pnum +num2=%1d\n",num1,num2,*pnum,*pnum  
+num2);  
}
```

```
-----  
// rereference
```

```
#include<stdio.h>  
int main(){  
    int a;  
  
    int *p; //=&a  
    if(p!=0){  
        *p=5;  
  
    }  
    printf("a=%d",a);  
}
```

```
-----  
// pass by reference
```

```
#include<stdio.h>  
int add(int *,int*);  
int main(){  
    int a=20,b=30;  
    printf("001 a=%d,b=%d\n",a,b);  
    int sum=add(&a,&b); //call by reference
```

```
printf("002 a=%d,b=%d\n",a,b);  
printf("the sum =%d\n",sum);  
return 0;  
}  
int add(int *a,int *b){  
    *a=30;  
    *b=40;  
    int s=*a+*b; //passing the values  
    return s;  
}
```
