Assignment

1.Vla using 2D array

```
#include<stdio.h>
int main()
{
   int n=2,m=5;
   int arr[n][m];
   printf("Enter 10 elements:\n");
   for(int i=0;i<n;i++){
      for(int j=0;j<m;j++){
         scanf("%d",&arr[i][j]);
         printf("\n");
      }
   }
   for(int i=0;i<n;i++)
   {
      for(int j=0;j<m;j++)
      {
        printf("%d ->",arr[i][j]);
      }
   }
   return 0;
}
```

Output:

```
Enter 10 elements:
10

20

30
```

40

50

60

70

80

90

100

10 ->20 ->30 ->40 ->50 ->60 ->70 ->80 ->90 ->100 ->

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 num);
int main()
{
   int num=5;
   printf("Original number:%d\n",num);
   increment(num);
   printf("Number after incrementing:%d\n",num);
   return 0;
}
void increment(int n)
{
   n=n+1;
```

```
    printf("Increment num:%d\n",n);
}
```

Output:

Original number:5

Increment num:6

Number after incrementing:5

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 a,int b);
int main()
{
    int x=10,y=20;
    printf("Before swap: x=%d, y=%d\n",x,y);
    swap(x,y);
    printf("After swap:x=%d, y=%d\n",x,y);
    return 0;
}
void swap(int a,int b)
{
    int temp=a;
    a=b;
    b=temp;
    printf("swap:a=%d, b=%d\n",a,b);

}
```

Output:

Before swap: x=10, y=20

swap:a=20, b=10

After swap:x=10, y=20

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

```
#include<stdio.h>
int factorial(int n);
int main()
{
   int n=5;
   int res=factorial(n);
   printf("Factorial of %d is:%d\n",n,res);
   return 0;
}
int factorial(int num)
{
   int fact=1;
   for(int i=1;i<=num;i++){
      fact*=i;
   }
   return fact;
}
```

Output:

Factorial of 5 is:120

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

```
#include<stdio.h>
int max(int a,int b);
int main()
{
```

```
    int n1=15,n2=25;
    int maximum=max(n1,n2);
    printf("Max of %d and %d is:%d\n",n1,n2,maximum);
    return 0;
}
int max(int a,int b)
{
    return (a > b) ? a : b;
}
```

Output:

Max of 15 and 25 is:25

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 a,int b);
int subtract(int a,int b);
int multiply(int a,int b);
int divide(int a,int b);
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));
   if(num2!=0){
     printf("Division:%d\n",divide(num1,num2));
   }
   else{
   printf("Can't Divide");
   }
   return 0;
}
int add(int a,int b){
   return a+b;
}
int subtract(int a,int b){
   return a-b;
}
int multiply(int a,int b){
```

```
    return a*b;
}
int divide(int a,int b){
    return a/b;
}
```

Output:

Enter first number:50

Enter second number:10

Addition:60

subtraction:40

Multiplication:500

Division:5

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 cel_to_fahren(float celsius);
float fahren_to_cel(float fahrenheit);
int main()
{
   float cel_temp,fahren_temp;
   printf("Enter temperature in celsius:");
   scanf("%f",&cel_temp);
   printf("Temperature in fahrenheit:%.1f\n",cel_to_fahren(cel_temp));
   printf("Enter temperature in fahrenheit:");
   scanf("%f",&fahren_temp);
   printf("Temperature in fahrenheit:%.1f\n",fahren_to_cel(fahren_temp));
   return 0;
}
float cel_to_fahren(float celsius){
   return (celsius*9/5)+32;
}
float fahren_to_cel(float fahrenheit){
   return(fahrenheit-32)*5/9;
}
```

Output:

Enter temperature in celsius:25

Temperature in fahrenheit:77.0

Enter temperature in fahrenheit:77

Temperature in fahrenheit:25.0

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×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 simple_interest(float principal,float rate,float time);
int main(){
   float principal,rate,time;
   printf("Enter principal amount:");
   scanf("%f",&principal);
   printf("Enter rate of interest:");
   scanf("%f",&rate);
   printf("Enter time period (in years):");
   scanf("%f",&time);
   printf("Simple interest is:%.1f\n",simple_interest(principal,rate,time));
   return 0;
}
float simple_interest(float principal,float rate,float time){
   return (principal*rate*time)/100;
```

```
}
```

Output:

```
Enter principal amount:1000
Enter rate of interest:5
Enter time period (in years):3
Simple interest is:150.0
```

```
//Exercise
//1.create a char type variable and initialize it to value 100
//2.print the adress 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 ch=100;
   printf("Address of ch:%p\n",&ch);
   char *ptr=&ch;
   char data=*ptr;
   printf("Value of ch using pointer:%d\n",data);
   *ptr=65;
   printf("New value of ch:%d\n",ch);
   return 0;

}
```

Output:

```
Address of ch:0x7ffdeb27b346
```

Value of ch using pointer:100

New value of ch:65

```
#include<stdio.h>
int main(void)
{
   int number=0;//A variable of type int initialized to 0
   int *pnumber=NULL;//A pointer that can point to type int
   number=10;
   printf("number's address:%p\n",&number);//output the address
   printf("number's value:%d\n\n",number);//output the value
   pnumber=&number;//store the address of number in pnumber
   printf("pnumber's address:%p\n",(void*)&pnumber);//output the address
   printf("pnumber's size:%zd bytes\n",sizeof(pnumber));//output the size
   printf("pnumber's value:%p\n",pnumber);//output the value(an address)
   printf("value pointed to:%d\n",*pnumber);//value at the address
   return 0;
}
```

Output:

number's address:0x7ffc02e7a86c

number's value:10

pnumber's address:0x7ffc02e7a860

pnumber's size:8 bytes

pnumber's value:0x7ffc02e7a86c

value pointed to:10

```
//Write a C program that swaps the values of two integers using pointers.
#include<stdio.h>
```

```
int main()
{
   int a,b;
   int*p1,*p2;
   printf("Enter the first number:");
   scanf("%d",&a);
   printf("Enter the second number:");
   scanf("%d",&b);
   p1=&a;
   p2=&b;
   printf("Before swapping:a=%d,b=%d\n",a,b);
   int temp;
   temp=*p1;
   *p1=*p2;
   *p2=temp;
   printf("After swaping:a=%d,b=%d\n",a,b);
   return 0;
}
```

Output:

```
Enter the first number:30
Enter the second number:25
Before swapping:a=30,b=25
After swaping:a=25,b=30
```

//WAP to swap the numbers using swap function and follow the pass by reference method

```
#include<stdio.h>
int swap(int *,int *);
int main()
{
   int a,b;
   printf("Enter the first number:");
```

```
    scanf("%d",&a);
    printf("Enter the second number:");
    scanf("%d",&b);
    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;
    temp=*p;
    *p=*q;
    *q=temp;
}
```

Output:

```
Enter the first number:100
Enter the second number:150
Before swapping a=100,b=150
After swapping a=150,b=100
```

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

```
#include<stdio.h>
int find_cube(int *);
int main()
{
    int num;
    printf("Enter the number:");
    scanf("%d",&num);
    find_cube(&num);
```

```
    printf("Cube of number is:%d\n",num);
    return 0;
}
int find_cube(int *n)
{
    *n=(*n)*(*n)*(*n);
}
```

Output:

```
Enter the number:3
Cube of number is:27
```

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

```
 #include<stdio.h>
float calculate_simple_interest(float *,float *,float *,float *);
int main()
{
    float principal,rate,time,simple_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);
    calculate_simple_interest(&principal,&rate,&time,&simple_interest);
    printf("Simple interest is:%.2f\n",simple_interest);
    return 0;

}
float calculate_simple_interest(float *principal,float *rate,float *time,float *simple_interest)
{
```

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

Output:

```
Enter principal amount:2000
Enter rate of interest:5
Enter time period (in years):3
Simple interest is:300.00
```