VARIABLE LENGTH ARRAY

```
#include<stdio.h>
int main(){
   int m=3;
   int n=2;
   int arr[m][n];
   for(int i=0;i<m;i++){
      for(int j=0;j<n;j++)
      scanf("%d",&arr[i][j]);
      printf("\n");
   }
   for(int i=0;i<m;i++){
      for(int j=0;j<n;j++){
         printf("%d ",arr[i][j]);
      }
      printf("\n");

   }
}
```

FUNCTION

```
//WAP to add two numbers using function without passing any parameters and not return anything
#include<stdio.h>
void add_num(void); //not passing and not returning

int main(){
   add_num();

   return 0;
}
void add_num(){
```

```
    int a=10,b=20,sum=0;
    sum=a+b;
    printf("Sum = %d",sum);
}
//WAP to add two numbers using function by passing parameters and not return anything
#include<stdio.h>
void add_num(int,int); // not returning

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

    return 0;
}
void add_num(int a,int b){
    int sum=0;
    sum=a+b;
    printf("Sum = %d",sum);
}
```

Printing the addresses of formal and actual parameters

```
#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);
}
```

WAP add two numbers by passing parameters and returning value

```
#include <stdio.h>

int add_num(int , int);
int main(){
    int a = 10, b = 20;
    int sum=0;
    sum=add_num(a,b);
    printf("Sum = %d \n",sum);
    return 0;

}

int add_num(int a, int b){
```

```
    int sum=a+b;
    return sum;
}
```

ASSIGNMENTS ON FUNCTION

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 inc_num(int);
int main(){
    int a = 10;
    printf("Value of a is %d\n",a);
    inc_num(a);
    printf("Value of a is %d\n",a);
    return 0;

}

void inc_num(int a){
    a++;
    printf("Value of a when incremented by 1 is %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_num(int,int);
int main(){
    int a = 10,b=15;
    printf("Value of a is %d\n",a);
    printf("Value of b is %d\n",b);
    swap_num(a,b);
    printf("Value of a is %d\n",a);
    printf("Value of b is %d\n",b);
    return 0;

}

void swap_num(int a,int b){
    int temp;
    temp=a;
    a=b;
    b=temp;
```

```
    printf("Value of a after swapping is %d\n",a);
    printf("Value of b after swapping is %d\n",b);
}
```

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

```
#include <stdio.h>

int fact_num(int);
int main(){
    int a;
    printf("Enter a number: ");
    scanf("%d",&a);
    int fact=fact_num(a);
    printf("Factorial of %d is %d",a,fact);
    return 0;

}

int fact_num(int a){
    int fact=1;
    for(int i=1;i<a;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 max_num(int,int);
int main(){
    int a,b;
    printf("Enter two numbers: ");
    scanf("%d %d",&a,&b);
    int max=max_num(a,b);
    printf("Maximum number among %d and %d is %d",a,b,max);
    return 0;

}

int max_num(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_num(int,int);
int sub_num(int,int);
int mul_num(int,int);
int div_num(int,int);
int main(){
    int a,b;
    printf("Enter two numbers: ");
    scanf("%d %d",&a,&b);
    int sum=add_num(a,b);
    int subtract=sub_num(a,b);
    int product=mul_num(a,b);
    float division=div_num(a,b);
    printf("Addition: %d\n",sum);
    printf("Subtraction: %d\n",subtract);
    printf("Multiplication: %d\n",product);
    printf("Division: %f\n",division);
    return 0;

}

int add_num(int a,int b){
    int sum=a+b;
    return sum;
```

```
}
int sub_num(int a,int b){
   int sub=a-b;
   return sub;
}
int mul_num(int a,int b){
   int mul=a*b;
   return mul;
}
int div_num(int a,int b){
   int division=a/b;
   return division;
}
```

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 CtoF(float);
float FtoC(float);
int main(){
   float c,f;
   printf("Enter temperature in Celcius: ");
   scanf("%f",&c);
   float fahren=CtoF(c);
   printf("Temperature in Fahrenheit: %.2f\n",fahren);
   printf("Enter temperature in Fahrenheit: ");
   scanf("%f",&f);
```

```
   float cel=FtoC(f);
   printf("Temperature in Celcius: %.2f",cel);

   return 0;

}

float CtoF(float c){
   float fahrenheit=(c*9/5)+32;
   return fahrenheit;
}
float FtoC(float f){
   float celcius=(f-32)*5/9;
   return celcius;
}
```

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×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 interest(float,float,int);
int main(){
   float p,r;
   int t;
   printf("Enter principal amount: ");
   scanf("%f",&p);
   printf("Enter rate of interest: ");
   scanf("%f",&r);
   printf("Enter time (in years): ");
   scanf("%d",&t);
```

```
   float i=interest(p,r,t);
   printf("Simple interest: %.2f",i);
   return 0;

}

float interest(float p,float r,int t){
   float i=p*r*t/100;
   return i;
}
```

POINTERS

```
#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);
}
```

Dereferencing:

```
#include<stdio.h>
int main(){
   int count=10,x;
   int *pCount=&count;
   x=*pCount;
   printf("count = %d, x = %d",count,x);
   return 0;
}
```

Output:

count = 10, x = 10

Char pointer

```
#include<stdio.h>
int main(){
   char var=100;
   printf("The address of var is %p\n",&var);
   char *pVar=&var;
   *pVar=var;
   printf("The data obtained from read operation is %c\n",*pVar);
   *pVar=65;
   printf("The value of variable now is %c\n",var);
   return 0;}
```

NULL pointers

```
#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\n",number);
   pnumber=&number;
   printf("pnumber's address:%p\n",(void*)&pnumber);
   printf("pnumber's size:%zd bytes\n",sizeof(pnumber));
   printf("pnumber's value:%p\n",pnumber);
   printf("value pointed to:%d\n",*pnumber);
   return 0;
}
```

Output:

```
number's address:0x7ffd4a38082c
number's value:10

pnumber's address:0x7ffd4a380830
pnumber's size:8 bytes
pnumber's value:0x7ffd4a38082c
value pointed to:10
```

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

```
#include<stdio.h>
int main(){
   int number1=10;
   int number2=15;
   int *pnumber1=&number1;
   int *pnumber2=&number2;
   printf("Value of number1 is %d\n",number1);
   printf("Value of number2 is %d\n",number2);
   int temp=*pnumber1;
   *pnumber1=*pnumber2;
   *pnumber2=temp;
   printf("Value of number1 after swapping is %d\n",*pnumber1);
   printf("Value of number2 after swapping is %d\n",*pnumber2);
   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: %ld, num2: %ld, *pNum: %ld, *pNum + num2 = %ld\n" , num1, num2,
*pNum, *pNum + num2 );
   return 0;
}
```

Pass by reference

```
#include<stdio.h>
int addnum(int *,int *);
int main(){
   int a=20,b=30;
   printf("001 a = %d b = %d\n",a,b);
   int sum=addnum(&a,&b);
   printf("001 a = %d b = %d\n",a,b);
   printf("sum = %d\n",sum);
   return 0;
}
int addnum(int *p,int *q){
   *p=30;
   *q=50;
   int s = *p+*q;
   return s;
}
```

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("a = %d b = %d\n",a,b);
   swap(&a,&b);
   printf("Values after swapping: \n");
   printf("a = %d b = %d\n",a,b);
   return 0;
}
int swap(int *p,int *q){
```

```
   int temp=*p;
   *p=*q;
   *q=temp;

}
```

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

```
#include<stdio.h>
int cube(int *);
int main(){
   int a;
   printf("Enter a number: ");
   scanf("%d",&a);
   int c=cube(&a);
   printf("Cube of %d is %d \n",a,c);
   return 0;
}
int cube(int *p){
   int c=(*p)*(*p)*(*p);
   return c;

}
```

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

```
#include <stdio.h>

float interest(float *,float *,int *);
int main(){
   float p,r;
   int t;
   printf("Enter principal amount: ");
   scanf("%f",&p);
   printf("Enter rate of interest: ");
   scanf("%f",&r);
   printf("Enter time (in years): ");
   scanf("%d",&t);
   float i=interest(&p,&r,&t);
   printf("Simple interest: %.2f",i);
   return 0;

}

float interest(float *p,float *r,int *t){
   float i=(*p)*(*r)*(*t)/100;
   return i;
}
```