PHASE 1 DAY 8

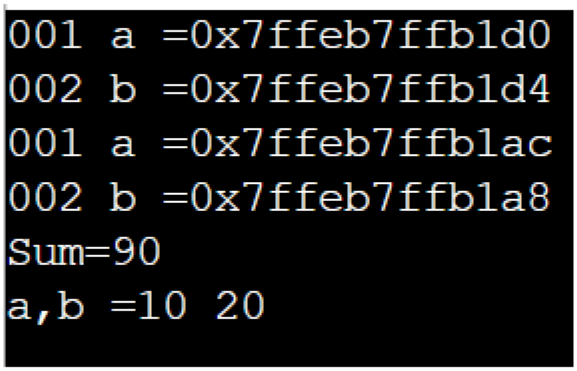
1.Variable-length arrays

```
#include <stdio.h>

int main()
{
    int n=10;
    int a[n];

    for(int i=0;i<10;i++)
    {
        scanf("%d",&a[i]);

    }
    for(int j=0;j<10;j++)
    {
        printf("%d->",a[j]);

    }

    return 0;
}
```

FUNCTIONS

1.

```
#include <stdio.h>
void add(int,int);
int main()
{
    int a=10,b=20;
    printf("001 a =%p\n",&a);
    printf("002 b =%p\n",&b);

    add(a,b);

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

    return 0;
}
void add(int a,int b)
{
```



```
001 a =0x7ffeb7ffb1d0
002 b =0x7ffeb7ffb1d4
001 a =0x7ffeb7ffb1ac
002 b =0x7ffeb7ffb1a8
Sum=90
a,b =10 20
```

```
    a=40;
    b=50;
    printf("001 a =%p\n",&a);
    printf("002 b =%p\n",&b);
    int sum=0;
    sum=a+b;
    printf("Sum=%d\n",sum);

}
```

2.Sum of numbers

```
#include <stdio.h>
int add(int,int);
int main()
{
    int a=10,b=20;

    int sum=add(a,b);

    printf("Sum=%d\n",sum);

    return 0;
}
int add(int a,int b)
{
    int sum=a+b;
    return sum;

}
```

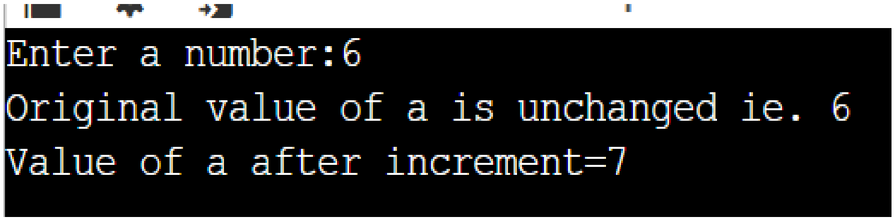
Output:

```
Sum=30
```

<u>Assignments</u>

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 a;

    printf("Enter a number:");

    scanf("%d",&a);

    int b=increment(a);

    printf("Original value of a is unchanged ie. %d\n",a);

    printf("Value of a after increment=%d",b);

    return 0;

}

int increment(int a)

{

    return ++a;


}
```



```
Enter a number:6
Original value of a is unchanged ie. 6
Value of a after increment=7
```

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 num1 = 5, num2 = 10;

        printf("Before swap: num1 = %d, num2 = %d\n", num1, num2);
        swap(num1, num2);

        printf("After swap: num1 = %d, num2 = %d\n", num1, num2);

        return 0;

}

void swap(int a, int b)

{

    int temp;

    temp = a;

    a = b;

    b = temp;

}
```

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

```
 #include <stdio.h>

int factorial(int);

int main()

{

    int a;

    printf("Enter a numbers:");

    scanf("%d",&a);


    int fact=factorial(a);

    printf("Factorial =%d\n",fact);
    return 0;

}

int factorial(int num)

{

    return num *factorial(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>
int largest(int,int);
int main()
{
    int a,b;
    printf("Enter 2 numbers:");
    scanf("%d %d",&a,&b);

    int large=largest(a,b);
    printf( "Largest=%d\n",large);

    return 0;
}
int largest(int num1,int num2)
{
    if(num1>num2)
    return num1;
    else
    return num2;

}
```

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.

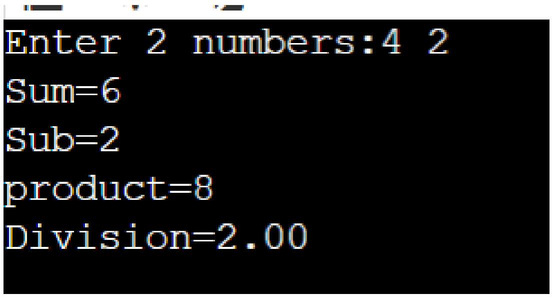
Example Input/Output:

```
Enter first number: 10
 Enter second number: 5
 Addition: 15
 Subtraction: 5
 Multiplication: 50
 Division: 2.0
```

ANSWER:

```
 #include <stdio.h>

int addition(int,int);

int subtraction(int,int);

int multiplication(int,int);

float division(int,int);



int main()
```

```
{
    int a,b;
    int sum,sub,product;
    float divi;

    printf("Enter 2 numbers:");
    scanf("%d %d",&a,&b);

    sum=addition(a,b);
    sub=subtraction(a,b);
    product=multiplication(a,b);
    divi=division(a,b);

    printf( "Sum=%d\nSub=%d\nproduct=%d\nDivision=%.2f\n",sum,sub,product,divi);
    return 0;
}

int addition(int num1,int num2)
{
    return num1+num2;
}
int subtraction(int num1,int num2)
{
    return num1-num2;
```



```
Enter 2 numbers:4 2
Sum=6
Sub=2
product=8
Division=2.00
```

```
}
int multiplication(int num1,int num2)
{
    return num1*num2;
}
float division(int num1,int num2)
{
    return num1/num2;
}
```

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.

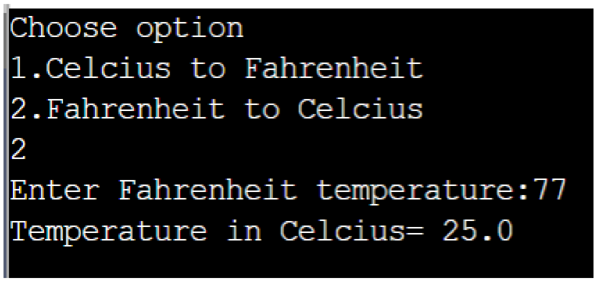
Example Input/Output:

```
 Enter temperature in Celsius: 25
 Temperature in Fahrenheit: 77.0



Enter temperature in Fahrenheit: 77
 Temperature in Celsius: 25.0
```

ANSWER:

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

```
float C2F(float);
float F2C(float);

int main()
{

    int option,celcius;
    float Fahrenheit;

    printf("Choose option\n1.Celcius to Fahrenheit\n2.Fahrenheit to Celcius\n");
    scanf("%d",&option);

    switch(option)
    {
        case 1:
        {
            printf("Enter Celcius temperature:");
            scanf("%d",&celcius);
            C2F(celcius);
            break;
        }
        case 2:
        {
            printf("Enter Fahrenheit temperature:");
```



```
Choose option
1.Celcius to Fahrenheit
2.Fahrenheit to Celcius
2
Enter Fahrenheit temperature:77
Temperature in Celcius= 25.0
```

```
      scanf("%f",&Fahrenheit);
      F2C(Fahrenheit);
      break;
    }
    default:
    printf("Enter valid option");
  }
}
float C2F(float C)
{
  printf("Temperature in Fahrenheit= %.1f",(C*9/5) +32);
}
float F2C(float F)
{
  printf("Temperature in Celcius= %.1f",(F-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.

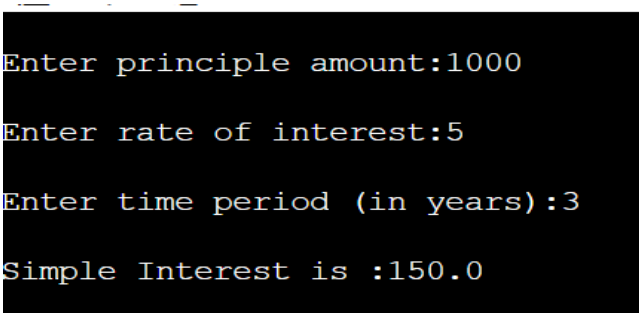
Requirements:
Implement a function that takes three parameters (principal, rate, time) and returns the calculated simple interest.
Use appropriate data types for financial calculations (e.g., float or double).
Utilize arithmetic operators to compute simple interest using the formula
SI = P×R×T/100

Example Input/Output:

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

ANSWER:

```
 #include <stdio.h>
float simple_intrest_calculator(int,int,int);

int main()
{

   int P,R,T;
   float SI;

   printf("\nEnter principle amount:");
   scanf("%d",&P);

   printf("\nEnter rate of interest:");
   scanf("%d",&R);

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

   SI=simple_intrest_calculator(P,R,T);
   printf("\nSimple Interest is :%.1f",SI);
}

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



```
Enter principle amount:1000

Enter rate of interest:5

Enter time period (in years):3

Simple Interest is :150.0
```

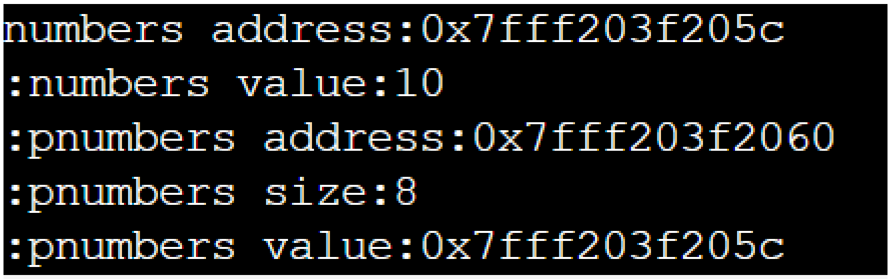
POINTERS

1.

```
 #include <stdio.h>

int main()
{
   char ch=100;
   printf("address of ch =%p\n",&ch);
   char *ptr=&ch;
```

```
    char new=*ptr;
    printf("new=%d\n",new);
    *ptr=65;
    printf("value:%d\n",new);

    return 0;
}
```

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

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

    pnumber=&number;

    printf("pnumbers address:%p\n:",&pnumber);
    printf("pnumbers size:%zd\n:",sizeof(pnumber));
    printf("pnumbers value:%p\n:",pnumber);
    printf("value pointed to%d\n:",*pnumber);

    return 0;
}
```



```
numbers address:0x7fff203f205c
:numbers value:10
:pnumbers address:0x7fff203f2060
:pnumbers size:8
:pnumbers value:0x7fff203f205c
```

3.Write a C program that swaps the values of two integers using pointers.(pass by reference)

```
 #include <stdio.h>
void swap(int*,int*);
int main()
{
    int a=10,b=20;
    printf("Before swapping:\na:%d\nb:%d\n",a,b);
    swap(&a,&b);
    printf("After swapping:\na:%d\nb:%d\n",a,b);

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

4.Pass by value

```
 #include <stdio.h>
void swap(int,int);
int main()
{
    int a=10,b=20;
    printf("Before swapping:\na:%d\nb:%d\n",a,b);
    swap(a,b);

    return 0;
}
    void swap(int  x,int  y)
    {
        int temp = x;
        x =y;
        y=temp;
            printf("After swapping:\na:%d\nb:%d\n",x,y);

    }
```

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

```
#include <stdio.h>
int cube(int*);
int main()
{
    int num;
    printf("Enter a number:");
    scanf("%d",&num);
    int result=cube(&num);
    printf("result=%d\n",result);

    return 0;
}
    int cube(int * x)
    {
        return *x * *x * *x;
    }
```

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

```
#include <stdio.h>

float simple_intrest_calculator(int *,int *,int *);

int main()

{

    int P,R,T;

    float SI;

     printf("\nEnter principle amount:");

    scanf("%d",&P);

    printf("\nEnter rate of interest:");

    scanf("%d",&R);

        printf("\nEnter time period (in years):");

    scanf("%d",&T);

        SI=simple_intrest_calculator(&P,&R,&T);

    printf("\nSimple Interest is :%.1f",SI);

}

float simple_intrest_calculator(int *p,int *r,int *t)

{

    return (*p)*(*r)*(*t)/100;

}
```