DAY 8

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)
{
   num = num + 1;
   printf("Value inside the function after increment: %d\n", num);
}

int main()
{
   int value ;
   printf("Enter a value: ");
   scanf("%d", &value);
   printf("Original value before function call: %d\n", value);

   increment(value);

   printf("Original value after function call: %d\n", value);

   return 0;
}
```

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 temp = a;
   a = b;
   b = temp;
   printf("Values inside the function after swap: a = %d, b = %d\n", a, b);
}

int main()
{
   int num1, num2;
   printf("Enter two numbers: ");
   scanf("%d%d", &num1, &num2);

   printf("Original values before function call: num1 = %d, num2 = %d\n", num1, num2);

   swap(num1, num2);

   printf("Original values after function call: num1 = %d, num2 = %d\n", num1, num2);
```

```
    return 0;
}
```

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

```
#include <stdio.h>

int factorial(int num)
{
    int fact=1;
    while(num)
    {
        fact=fact*num--;
    }
    return fact;
}

int main()
{
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    int res=factorial(num);

    printf("Factorial of %d is %d\n", num, res);
    return 0;
}
```

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(int a, int b)
{
   if(a > b)
   return a;
   else
   return b;
}

int main()
{
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d%d", &num1, &num2);
    int res=max(num1, num2);

    printf("The maximum given of two numbers is : %d\n",res);
    return 0;
}
```

5. 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 divi(int, int);

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", sub(num1, num2));
    printf("Multiplication: %d\n", mult(num1, num2));
    printf("Division: %.1f\n", divi(num1, num2));

    return 0;
}
```

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

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

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

float divi(int a, int b)
{
   return ((float)a/b);
}
```

6. 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 c)
{
   return (c * 9.0 / 5.0) + 32.0;
}
```

```
float FtoC(float f)
{
   return (f - 32.0) * 5.0 / 9.0;
}

int main()
{
   float c, f;

   printf("Enter temperature in Celsius: ");
   scanf("%f", &c);
   f = CtoF(c);
   printf("Temperature in Fahrenheit: %.1f\n", f);


   printf("Enter temperature in Fahrenheit: ");
   scanf("%f", &f);
   c = FtoC(f);
   printf("Temperature in Celsius: %.2f\n", c);

   return 0;
}
```

7. 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 cal_simple_interest(int p, int r, int t)
```

```
{
   return (p*t*r)/100.0;
}
int main()
{
   int prin_amt, rate, period;
   printf("Enter principal amount: ");
   scanf("%d", &prin_amt);
   printf("Enter rate of interest: ");
   scanf("%d", &rate);
   printf("Enter time period (in years): ");
   scanf("%d", &period);

   float simple_interest=cal_simple_interest(prin_amt, rate, period);
   printf("Simple interest is: %.1f", simple_interest);
}
```

8. 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("Address of var: %p\n", &num);


   char *ptr = &num;

   char data = *ptr;
   printf("Data read from pointer: %d\n", data);

   *ptr = 65;

   printf("Updated value of var: %d\n", num);

   return 0;
}
```

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

```
#include <stdio.h>
int main()
{
   int num1, num2;
   printf("Enter two numbers: ");
   scanf("%d%d", &num1, &num2);
   printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);
   int *p1 = &num1;
   int *p2 = &num2;
   int temp = *p1;
   *p1 = *p2;
   *p2 = temp;
   printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);

}
```

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

```
#include <stdio.h>
void swap(int *p1, int *p2)
{
   int temp = *p1;
   *p1 = *p2;
   *p2 = temp;
}
int main()
{
   int num1, num2;
   printf("Enter two numbers: ");
   scanf("%d%d", &num1, &num2);
   printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);
   swap(&num1, &num2);
   printf("After swapping: num1 = %d, num2 = %d", num1, num2);

}
```

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

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

   cube(&num);
```

```
    printf("Cube is %d", num);

}
```

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

```
#include <stdio.h>

void calculateSimpleInterest(float principal, float rate, float time, float *interest)
{
    *interest = (principal * rate * time) / 100;
}

int main()
{
    float principal, rate, time, interest;

    printf("Enter Principal amount: ");
    scanf("%f", &principal);
    printf("Enter Rate of interest: ");
    scanf("%f", &rate);
    printf("Enter Time (in years): ");
    scanf("%f", &time);

    calculateSimpleInterest(principal, rate, time, &interest);

    printf("The Simple Interest is: %.2f\n", interest);

    return 0;
}
```