DAY 8

1)

```
#include <stdio.h>

int main()
{
   int n=10;
   int arr[n]={2,3};
   for(int i=0;i<10;i++){
      scanf("%d",&arr[i]);
   }
   for(int j=0;j<10;j++){
       printf("%d",arr[j]);
   }
    return 0;
}
```

//not able to intialise values in var length array

2)//Variable length for 2-D array

```
#include <stdio.h>

int main()
{
   int i=2,j=3;
   int arr[i][j];
   for(int i=0;i<2;i++)
   {
       for(int j=0;j<3;j++){
         scanf("%d",&arr[i][j]);
       }
```

```
   }
   printf("\n");
   printf("<--Array--> \n");

  for(int i=0;i<2;i++)
   {
       for(int j=0;j<3;j++){
         printf("%d\t\t",arr[i][j]);
       }
      printf("\n");
   }
    return 0;
}
```

//not able to intialise values in var length array

Output

```
5
4
8
9
7
2

<--Array-->
5               4               8
9               7               2
```

FUNCTION

1)//wap to add two numbers using add function without passing any parameters

//and funvtion is not going to return any data

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

```
void add_num(void);//function definition
int main(){

     add_num();//function call

    return 0;
}

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

Output

```
sum=30
```

2)//wap to add two numbers using add function passing any parameters
//and funvtion is not going to return any data

```
#include<stdio.h>
void add_num(int,int);//function definition
int main(){
     int a=10,b=20;
     add_num(a,b);//function call

    return 0;
}

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

sum=30

4)//wap to add two numbers using add function passing any parameters
//and funvtion is not going to return any data

```
#include<stdio.h>
void add_num(int,int);//function definition
int main(){
     int a=10,b=20;
     printf("001a=%p\n",&a);
     printf("001b=%p\n",&b);
     add_num(a,b);//function call
     printf("values of a and b is %d ,%d",a,b);
    return 0;
}

void add_num(int a,int b){
     printf("002a=%p",&a);
     printf("002b=%p",&b);
    a=40,b=50;
    int sum=0;
    sum=a+b;
    printf("sum=%d\n",sum);
}
```

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>
```

```
void increment(int);//function definition
int main(){
    int a=1;
    increment(a);
    printf("value of a1=%d\n",a);
    return 0;
}

void increment(int a){
   a=2;
   a=a+1;
   printf("value of a2=%d\n",a);

}
```

Output
[?2004l
value of a2=3
value of a1=1
[?2004h

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);//function definition
int main(){
    int a=1,b=2;
    swap(a,b);
    printf("value of a1=%d,b1=%d\n",a,b);
    return 0;
}
```

```
void swap(int a,int b){
    a=5;
    b=6;
    int temp;
    temp=a;
    a=b;
    b=temp;

    printf("value of a2=%d,b2=%d\n",a,b);

}
```

```
[?2004l
value of a2=6,b2=5
value of a1=1,b1=2
[?2004h
```

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

```
#include<stdio.h>
int factorial(int);//function definition
int main(){
    int a=6;
    int result=factorial(a);
    printf("factorial of a2=%d\n",result);
    return 0;
}

int factorial(int a){
    int fact=1;
    for(int i=1;i<=a;i++)
    {
```

```
        fact=fact*i;

    }
    return fact;

}
```

4)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,int);//function definition
int main(){
    int a=6,b=8;
    int result=max(a,b);
    printf("\nmax=%d\n",result);
    return 0;
}

int max(int a,int b){
    int max=a;
    if(max<b){
        max=b;


    }
    else{
        max=a;


    }
    return max;
}
```

Output
max=8

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 divison(int,int);
int multiply(int,int);
int main(){
    int a,b,result;
    printf("enter first number:");
    scanf("%d",&a);
    printf("Enter second number:");
    scanf("%d",&b);


    result=add(a,b);
    int x=sub(a,b);
    int y=divison(a,b);
    int z=multiply(a,b);
    printf("Addition:%d\n",result);
    printf("Substraction:%d\n",x);
    if(a>b){
      printf("Divison:%d\n",y);
    }

    else{
        printf("please enter higher value for %d\n",a);
    }
    printf("Multiplication:%d",z);
    return 0;
}

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

```
int divison(int a,int b){
    int divison=a/b;
    return divison;
}
int multiply(int a,int b){
    int multiply=a*b;
    return multiply;
}
```

Output

```
[?2004l
enter first number:5
Enter second number:6
Addition:11
Substraction:-1
please enter higher value for 5
Multiplication:30[?2004h
```

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>
void celcius_farenheit(float);
void fareheit_celcius(float);
int main(){
    float temp;
    celcius_farenheit(temp);
    fareheit_celcius(temp);



    return 0;
}

void celcius_farenheit(float temp){
    printf("enter temperature in celcius:");
    scanf("%f",&temp);
    float farenheit=(temp * 9.0 / 5.0) + 32;
    printf("Temperature in fareheit:%.2f\n",farenheit);
```

```
}

void fareheit_celcius(float temp){
    printf("enter temperature in farenheit:");
    scanf("%f",&temp);
    float celcius=(temp - 32)*5 / 9;;
    printf("Temperature in celcius:%.2f\n",celcius);


}
```

```
[?2004l
enter temperature in celcius:25
Temperature in fareheit:77.00
enter temperature in farenheit:77
Temperature in celcius:25.00
[?2004h
```

3)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>
void simple_intrest(float,int,int);
int main(){
    float principal_amt;
    int rate;
    int time;
    simple_intrest(principal_amt,rate,time);

    return 0;
}

void simple_intrest(float principal_amt,int rate,int time){

    printf("Enter Principal Amount:");
    scanf("%f",&principal_amt);
    printf("enter rate of intrest:");
    scanf("%d",&rate);
    printf("enter time period(in years):");
    scanf("%d",&time);
    float intrest=(principal_amt*rate*time)/100;
    printf("Simple Intrest:%.1f",intrest);

}
```

```
Enter Principal Amount:1000
enter rate of intrest:5
```

enter time period(in years):3
Simple Intrest:150.0

POINTERS

```
1)#include<stdio.h>
int main(){
    int a;
    int *p;
    p=&a;
    *p=20;

    printf("%d\n",a);
    printf("address of a=%p\n",&a);
    printf("address of *p=%p\n",&p);
    printf("value of *p=%p",p);



    return 0;
}
```

[?2004l
20
address of a=0x7ffcb721446c
address of *p=0x7ffcb7214470
value of *p=0x7ffcb721446c[?2004h

3)
Exercise

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
Te

```
#include<stdio.h>
int main(){
   char a=100;
   printf("address of a=%p\n",&a);
   char *p=&a;
   char x=*p;
   printf("value of x=%d\n",x);
   *p=65;
   printf("value of a=%d",a);

    return 0;
}
```

```
[?2004l
address of a=0x7ffe863bbf4e
value of x=100
value of a=65[?2004h
```

```
5)#include<stdio.h>
int main(){
  int number=0;
  int *p=NULL;
  number=10;
```

```
  printf("number address:%p\n",&number);
  printf("number value=%d\n",number);


  p=&number;


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

    return 0;
}



[?2004l
number address:0x7ffe465549dc
number value=10
p address:0x7ffe465549e0
p size=8
p value=0x7ffe465549dc
value pointed to 10
[?2004h
```

6)SWAPPING

```
#include <stdio.h>

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

```
}

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

    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;
}
```

7)

```
#include <stdio.h>


int main() {

    long num1=0;
    long num2=0;
    long *p=NULL;

    p=&num1;
    *p=2;
    ++num2;
    num2=num2+*p;


    p=&num2;
    ++*p;


    printf("num1=%ld num2=%ld   *p=%ld
*p+num2=%ld\n",num1,num2,*p,*p+num2);
```

```
    return 0;
}
```

8)//testing null

```
#include <stdio.h>


int main() {

   int a;
   int *p;
   if(p!=0){
       *p=5;
   }
 printf("value of a=%d",a);
    return 0;
}
```

9)//call by reference

```
#include <stdio.h>
int addnum(int *,int *);

int main() {

  int a=20,b=30;
  printf("001a=%d  001b=%d",a,b);
  int sum=addnum(&a,&b);
  printf("\n002a=%d  002b=%d",a,b);
  printf("\nsum=%d\n",sum);
    return 0;
}
int addnum(int *a,int *b){
    *a=30;
```

```
    *b=70;
    int s=*a+*b;
    return s;


}
```

10)//call by reference swap

```
#include <stdio.h>
int swap(int *,int *);

int main() {

  int a=20,b=30;
  printf("Value before swap a=%d,b=%d\n",a,b);
  int x=swap(&a,&b);
  printf("Value after swap a=%d,b=%d",a,b);
    return 0;
}
int swap(int *x,int *y){
    int temp=*x;
    *x=*y;
    *y=temp;
    return *x,*y;

}
```