

## DAY 8

### 1.//VLA

```
#include<stdio.h>

int main()
{
    int n=10;

    //declaring vla
    int ar[n];//={1,2,3}; // in vla intialization and declaration not together
    printf("enter elements \n");

    //intializing
    for(int i=0;i<10;i++)
    {
        scanf("%d",&ar[i]);
    }
    for(int i=0;i<10;i++)
    {
        printf("%d -->",ar[i]);
    }

}
```

```

PS D:\learning c\output> & .\'day8_1.exe'
enter elements
1
2
3
4
5
6
7
8
9
0
1 -->2 -->3 -->4 -->5 -->6 -->7 -->8 -->9 -->0 -->
PS D:\learning c\output>

```

2.

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

```
int main()
```

```
{ // two dimensional vla
```

```
int n=2,a=4;;
```

```
int ar[n][a];
```

```
printf("enter elements \n");
```

```
for(int i=0;i<2;i++)
```

```
{
```

```
for(int j=0;j<4;j++)
```

```
{
```

```
scanf("%d",&ar[i][j]);
```

```
}
```

```
}
```

```
for(int i=0;i<2;i++)
```

```
{
```

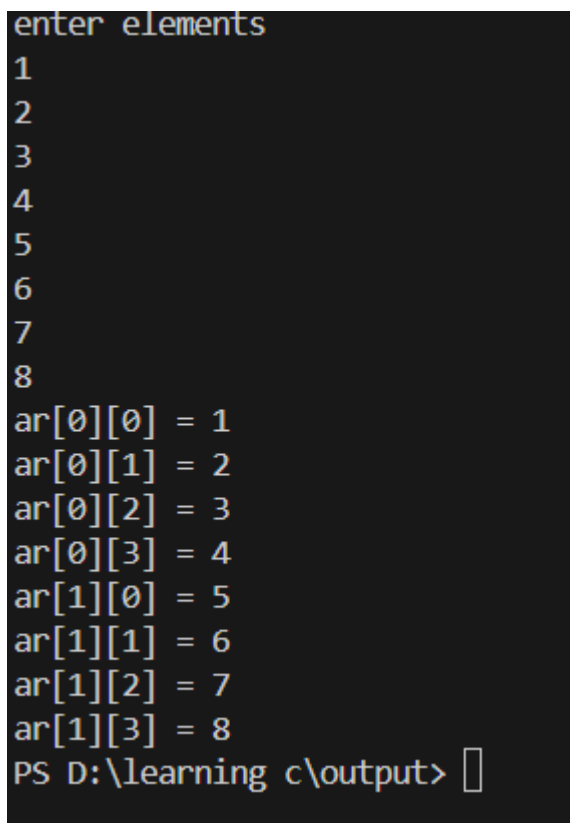
```
for(int j=0;j<4;j++)
```

```

    {
        printf("ar[%d][%d] = %d \n",i,j,ar[i][j]);
    }
}

}

```



```

enter elements
1
2
3
4
5
6
7
8
ar[0][0] = 1
ar[0][1] = 2
ar[0][2] = 3
ar[0][3] = 4
ar[1][0] = 5
ar[1][1] = 6
ar[1][2] = 7
ar[1][3] = 8
PS D:\learning c\output>

```

3. //function

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

```
int sum( int a,int b)//if i didn't given int a,int b instead a, b it is by default considered as int
```

```

{
    printf("sum is %d",a+b);
}

```

```
int main()
```

```

{ int a,b;

    printf("enter a number");

```

```

scanf("%d",&a);

printf("enter num2");

scanf("%d",&b);

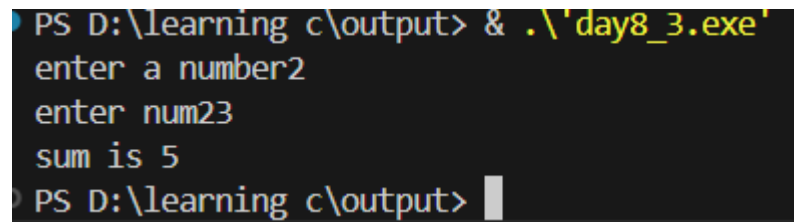
//int add=sum(a,b);

// printf("sum is %d",add);

sum(a,b);

}

```



```

PS D:\learning c\output> & .\'day8_3.exe\'
enter a number2
enter num23
sum is 5
PS D:\learning c\output>

```

4. /\*

1. function prototype/declaration :before main()

2.function call:implemented inside main()

3.function definition:implemented after main()

\*/

//wap to add two numbers using add fn without passing any parameter , function is not doing return any data

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

```
void add_num(void);//since not passing para (void) and since not returning anything void fnname
```

```
int main()
```

```
{
```

```
    add_num();
```

```
    return 0;
```

```
}
```

```
void add_num()
```

```
{
```

```
    int a=10;
```

```
    int b=20;
```

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

```
    printf("%d",sum);
```

```
}
```

```
PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day8_4.exe'
30
PS D:\learning c\output>
```

5. //wap to add two numbers using add fn by passing any parameter , function is not doing return any data

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

```
void add_num(int , int);
```

```
int main()
```

```
{
```

```
    int a=10,b=20;
```

```
    add_num(a,b);
```

```
    return 0;
```

```
}
```

```
//void add_num(int a,int b)
```

```
void add_num(int c,int d)
```

```
{
```

```
    int sum;
```

```
    //sum=a+b;
```

```
    sum=c+d;
```

```
    printf("sum is %d",sum);
```

```
}
```

```
PS D:\learning c\output> & .\'day8_5.exe'
sum is 30
PS D:\learning c\output>
```

6. #include<stdio.h>

```
void add_num(int , int);
```

```
int main()
```

```
{
```

```

int a=10,b=20;

add_num(a,b);

printf("the values of a and b is %d \n",a,b);

return 0;

```

```

}

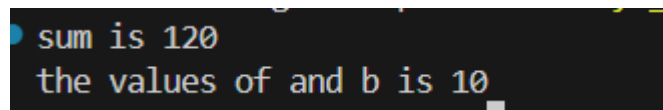
//void add_num(int a,int b)
void add_num(int a,int b)
{
    a=50;
    b=70;
    int sum;
    //sum=a+b;
    sum=a+b;
    printf("sum is %d \n",sum);
}

```

//after executing fn when it goes back to main function why is value of a and b not changed

//here pass by value happens

// int a=10 and int b=20 then a copy of values are passed to fn



```

sum is 120
the values of a and b is 10

```

7. //wap to add two number using add fn by passing parameter and

// function is going to return sum

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

```
int add_num(int,int);
```

```
int main()
```

```
{ int a=10;
```

```
    int b=20;
```

```
    int sum;
```

```
    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;
}

```

```

PS D:\learning c\output> & .\'day8_7.exe'
sum=30
PS D:\learning c\output>

```

8. #include<stdio.h>

```

int main()
{
    int a;
    int *p;
    p=&a;
    *p=20;
    printf("a=%d \n",a);
    printf("adress of a =%p \n",&a);
    printf("adress of *p =%p \n",&p);
    printf("*p=%p \n",p);
}

```

```

a=20
adress of a =0061FF1C
adress of *p =0061FF18
*p=0061FF1C
PS D:\learning c\output>

```

10. #include<stdio.h>

```

int main()
{
    int a=10;
    printf("001 a =%d \n",a);
}

```

```

int *p=&a;

a=*p+5;

printf("002 a =%d \n",a);

}

```

```

PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day8_10.exe'
001 a =10
002 a =15

```

11. #include<stdio.h>

```

int main()
{
    int count=10,x;
    int *pcount=&count;
    x=*pcount;
    printf("count =%d,x=%d ",count,x);

}

```

```

PS D:\learning c\output> cd 'd:\learning c\output'
PS D:\learning c\output> & .\'day8_11.exe'
count =10,x=10

```

12. #include<stdio.h>

```

int main()
{
    int count=10,x;
    int *pcount=&count;
    x=*pcount;
    printf("count =%d,x=%d ",count,x);
    printf("address of pcount =%p \n",(void*)&pcount);
    printf("size of pcount=%d \n",sizeof(pcount));

}

```



```

● PS D:\learning c\output> & .\'day8_12.exe'
count =10,x=10 address of pcount =0061FF14
size of pcount=4
● PS D:\learning c\output>

```

13. #include<stdio.h>

int main(void)

```

{
    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("pnumbers size :%zd bytes \n",sizeof(pnumber));
    printf("pnumber's value :%p \n",pnumber);
    printf("value pointed to:%d \n",*pnumber);
}

```

```

● PS D:\learning c\output> & .\'day8_13.exe'
● number's address :0061FF1C
  number's value:10

  pnumber's address:0061FF18
  pnumbers size :4 bytes
  pnumber's value :0061FF1C
  value pointed to:10

```

14. #include<stdio.h>

int main(void)

```

{
    long num1=0L;
    long num2=0L;
    long *pnum=NULL;

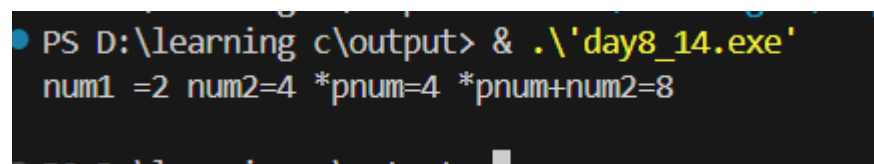
```

```

pnum=&num1;
*pnum=2L;
++num2;
num2+=*pnum;
pnum=&num2;
++*pnum;

printf("num1 =%ld num2=%ld *pnum=%ld *pnum+num2=%ld \n ",
num1,num2,*pnum,*pnum+num2);
}

```



```

PS D:\learning c\output> & .\'day8_14.exe'
num1 =2 num2=4 *pnum=4 *pnum+num2=8

```

15. //call 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("002 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;
}

```

```

PS D:\learning c\output> & .\'day8_15.exe'
001 a = 20    b = 30
002 a = 30    b = 50
sum = 80

```

## 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.

/\*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_num(int);
```

```
int main()
```

```
{
```

```
    int n=10;
```

```
    increment_num(n);
```

```
    printf(" n is %d \n",n);
```

```
    return 0;
```

```
}
```

```
void increment_num(int n)
```

```
{
```

```
    n=11;
```

```
    n=n+1;
```

```
    printf("n after incrementing is %d \n",n);
```

```
}
```

```

PS D:\learning c\output> & .\'day8_ass1.exe'
n after incrementing is 12
n is 10

```

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.

/\*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;
    int b=4;
    swap_num(a,b);
    printf("the values of a and b are: %d ,%d",a,b);
    return 0;
}

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

}
```

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

/\*Develop a C program that calculates the factorial of a number using call by value.\*/

```
#include<stdio.h>

int factorial(int);

int main() {
    int n = 4;
    int fact = factorial(n);
```

```

printf("The value of n is %d\n", n);

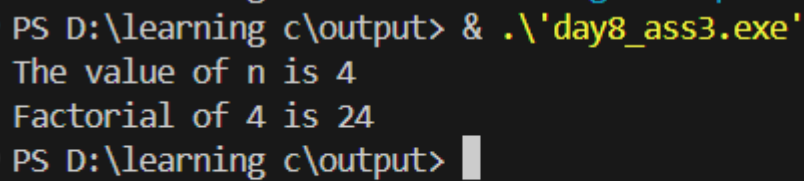
printf("Factorial of %d is %d\n", n, fact);

return 0;
}

int factorial(int n) {
    if (n <= 1) return 1;

    return n * factorial(n - 1);
}

```



```

PS D:\learning c\output> & .\'day8_ass3.exe\'
The value of n is 4
Factorial of 4 is 24
PS D:\learning c\output>

```

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

/\*Create a C program that defines a function to find the maximum of two numbers using call by value.\*/

```

#include <stdio.h>
int findMax(int ,int);

```

```

int main() {
    int num1, num2, max;

```

```

    printf("Enter the first number: ");
    scanf("%d", &num1);
    printf("Enter the second number: ");
    scanf("%d", &num2);

```

```

    // Call the function to find the maximum
    max = findMax(num1, num2);

```

```

    printf("The maximum of %d and %d is %d\n", num1, num2, max);

```

```

    return 0;
}

```

```

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

```

}

```
PS D:\learning c\output> & .\'day8_ass4.exe'  
Enter the first number: 2  
Enter the second number: 3  
The maximum of 2 and 3 is 3  
PS D:\learning c\output> █
```

### 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
```

/\*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.\*/

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

```
void add(int,int);
```

```
void subtract(int,int);
```

```
void multiply(int,int);
```

```
void division(int,int);
```

```
int main()
```

```
{
```

```
    int num1,num2;
```

```
    printf("enter the first number \n");
```

```
    scanf("%d",&num1);
```

```
    printf("enter the second number \n");
```

```
    scanf("%d",&num2);
```

```
    add(num1,num2);
```

```
    subtract(num1,num2);
```

```
    multiply(num1,num2);
```

```
    division(num1,num2);
```

```
    return 0;
```

```
}
```

```
void add(int num1, int num2)
```

```
{
```

```
    printf("\n sum is %d",num1+num2);
```

```
}
```

```
void subtract(int num1,int num2)
```

```
{
```

```
    printf("\n after subtraction:%d",num1-num2);
```

```
}
```

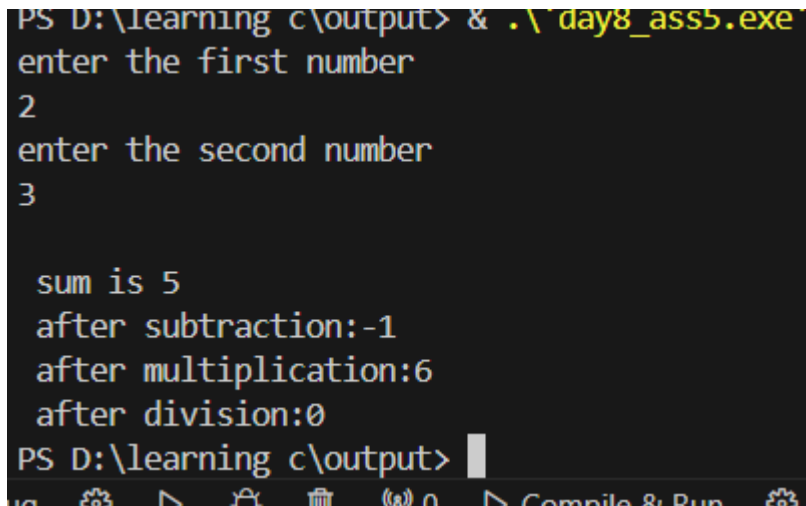
```
void multiply(int num1,int num2)
```

```
{
```

```

    printf("\n after multiplication:%d",num1*num2);
}
void division(int num1,int num2)
{
    printf("\n after division:%d",num1/num2);
}

```



```

PS D:\learning c\output> & .\day8_ass5.exe
enter the first number
2
enter the second number
3

sum is 5
after subtraction:-1
after multiplication:6
after division:0
PS D:\learning c\output>

```

## 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

/\* 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.\*/

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

```
float celsius(float);
```

```
float fahrenheit(float);
```

```
int main()
```

```
{
```

```
    float c,f;
```

```
    printf("enter temperature in celsius \n");
```

```
    scanf("%f",&c);
```

```
    float covert_to_f=celsius(c);
```

```
    printf("celsius to fahrenheit : %f \n",covert_to_f);
```

```
    printf("enter temperature in fahrenheit \n ");
```

```
    scanf("%f",&f);
```

```
    float covert_to_c=fahrenheit(f);
```

```
    printf("fahreinheit to celsius : %f \n",covert_to_c);
```

```
    return 0;
```

```
}
```

```
float celsius(float c)
```

```
{
```

```
    return (c*9/5)+32;
```

```
}
```

```
float fahrenheit(float f)
```

```
{
```

```
    return (f-32)*5/9;
```

```
}
```

```

enter temperature in celsius
27
celsius to fahrenheit : 80.60
enter temperature in fahrenheit
80.60
fahrenheit to celsius : 27.00
PS D:\learning c\output>

```

## 7.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

$$SI = P \times R \times T / 100$$

Example Input/Output:

Enter principal amount: 1000

Enter rate of interest: 5

Enter time period (in years): 3

Simple Interest is: 150.0

/\* 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.\*/

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

```
float simple_interest(float,float,float);
```

```
int main()
```

```
{
```

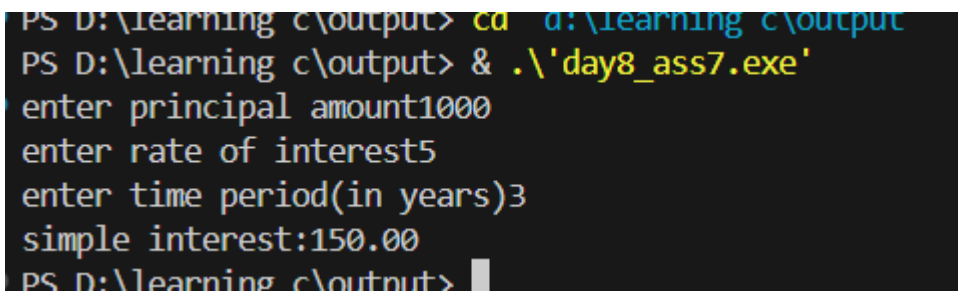
```
    float p,r,t;
```

```
    printf("enter principal amount");
```

```

scanf("%f",&p);
printf("enter rate of interest");
scanf("%f",&r);
printf("enter time period(in years)");
scanf("%f",&t);
float si=simple_interest(p,r,t);
printf("simple interest:%f",si);
return 0;
}
float simple_interest(float p,float r, float t)
{
    return (p*r*t)/100;
}

```



```

PS D:\learning c\output> cd d:\learning c\output
PS D:\learning c\output> & .\'day8_ass7.exe'
enter principal amount1000
enter rate of interest5
enter time period(in years)3
simple interest:150.00
PS D:\learning c\output>

```

8.

## 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

```
#include<stdio.h>

int main()
{
    char var=100 ,x;

    printf("adress of var is %p \n",&var);

    char *p=&var;

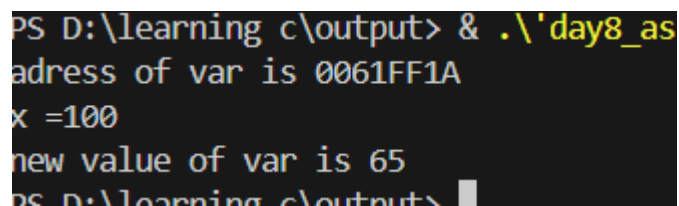
    x=*p;

    printf("x =%d \n",x);

    *p=65;

    printf("new value of var is %d",var);

}
```



```
PS D:\learning c\output> & .\day8_as
adress of var is 0061FF1A
x =100
new value of var is 65
PS D:\learning c\output>
```

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

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

```
int main()
```

```
{
```

```

int a=10,b=20,temp;

int*p1=&a;
int*p2=&b;

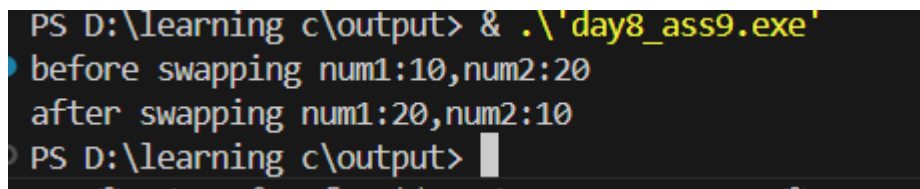
printf("before swapping num1:%d,num2:%d \n",a,b);

temp=*p1;
*p1=*p2;
*p2=temp;

printf("after swapping num1:%d,num2:%d \n",a,b);

}

```



```

PS D:\learning c\output> & .\'day8_ass9.exe\'
before swapping num1:10,num2:20
after swapping num1:20,num2:10
PS D:\learning c\output>

```

10. //wap to swap the numbers using swap\_function and follow the pass by reference method

```

#include<stdio.h>

void swap(int *,int *);

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

    printf("before swapping num1:%d,num2:%d \n",a,b);

    swap(&a,&b);

    printf("after swapping num1:%d,num2:%d \n",a,b);
}

void swap(int*p,int*q)
{
    *p=10;
    *q=20;

    int temp=*p;

```

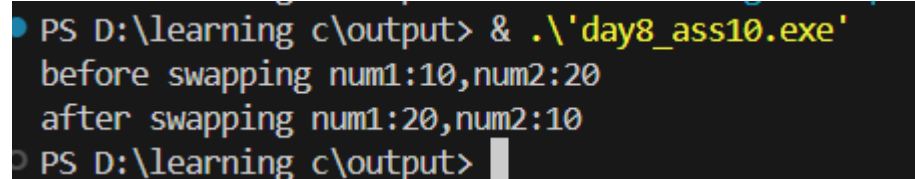
```

    *p=*q;

    *q=temp;

}

```



```

PS D:\learning c\output> & .\'day8_ass10.exe'
before swapping num1:10,num2:20
after swapping num1:20,num2:10
PS D:\learning c\output>

```

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

```

#include<stdio.h>

int cube(int *);

int main()
{
    int n=2;

    int c=cube(&n);

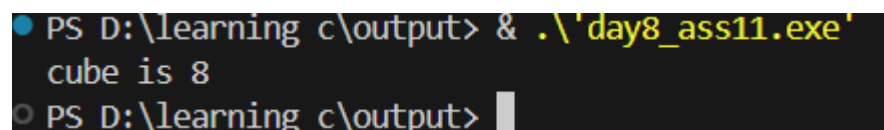
    printf("cube is %d \n",c);

    return 0;
}

int cube(int *p)
{
    *p=2;

    return *p * *p * *p;
}

```



```

PS D:\learning c\output> & .\'day8_ass11.exe'
cube is 8
PS D:\learning c\output>

```

12. //wap to find SI using pass by reference

```

#include<stdio.h>

float simple_interest(int *,int *,int *);

int main()
{

```

```

int p,r,t;

printf("enter principal amount");

scanf("%d",&p);

printf("enter rate of interest");

scanf("%d",&r);

printf("enter time period(in years)");

scanf("%d",&t);

int si=simple_interest(&p,&r,&t);

printf("simple interest:%d",si);

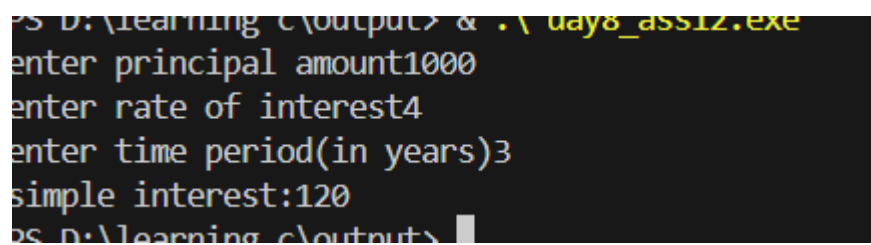
return 0;
}

float simple_interest(int *pr,int *prate,int *ptime)
{

return ((*pr)*(*prate)*(*ptime))/100;

}

```



```

PS D:\learning c\output> & .\ day8_ass12.exe
enter principal amount1000
enter rate of interest4
enter time period(in years)3
simple interest:120
PS D:\learning c\output>

```