9. Write a program that uses a function to swap two numbers. The function takes as input the pointers to the two variables that contain the values to be swapped.

Description:

This program demonstrates how to **swap two numbers using a function with pointers** in C.

- The user enters two numbers.

- A function swap() is used to **swap their values**.

- The function takes **pointers** to the two numbers as arguments.

- Using **pointer dereferencing**, the values are swapped inside the function.

- After returning from the function, the swapped values are displayed.

Example:

Enter two numbers: 5 10

After swapping: 10 5

Algorithm:

```
Step 1: Start

Step 2: Declare two integer variables a and b to store the user-input values.

Step 3: Prompt the user to enter two numbers and store them in a and b.

Step 4: Define a function swap(int *x, int *y) that:
    • Takes pointers to a and b as arguments.
    • Uses a temporary variable to swap their values.
    • Updates the values in the original memory locations using pointer dereferencing.

Step 5: Call swap(&a, &b) to pass the addresses of a and b.

Step 6: Print the values of a and b after swapping.

Step 7: Stop.
```

Source code:

```
#include <stdio.h>

void swap(int *p, int *q);

int main()
{
        int x, y;

        printf("Enter the value of x:");
        scanf("%d", &x);

        printf("Enter the value of y:");
        scanf("%d",&y);

        printf("Before Swapping\nx = %d\ny = %d\n", x,

        y); swap(&x, &y);

        printf("After Swapping\nx = %d\ny = %d\n", x, y);

        return 0;
}

void swap(int *p, int *q)
{
    int temp;

    temp = *p;
    *p = *q;
    *q = temp;
}
```

Sample Output:

```
Enter the value of x:201
Enter the value of y:123
Before Swapping
x = 201
y = 123
After Swapping
x = 123
y = 201
```