Experiment 1: Write a C Program to swap two numbers without using the 3rd variable using pointers.

Objective:

- To swap two numbers using pointer manipulation, without using a third (temporary) variable.

Problem Analysis:

In this program, the goal is to swap the values of two numbers without using a third (temporary) variable. This is achieved using pointer variables and simple arithmetic operations. The user enters two numbers. Then, pointers are used to access and manipulate the values directly in memory using addition and subtraction. The swapped values are then displayed.

Input variable	Pointer variable	Processing variable	Output variable	Header file
a,b(int)	X,Y → Pointing to a and b	Swap values using arithmetic and pointers: X =X+Y, Y=X-Y, X=X-Y	Swapped values of a and b	<stdio.h>

Algorithm:

Step1: Start

Step2: Declare integers a, b

Step3: Take input for a and b

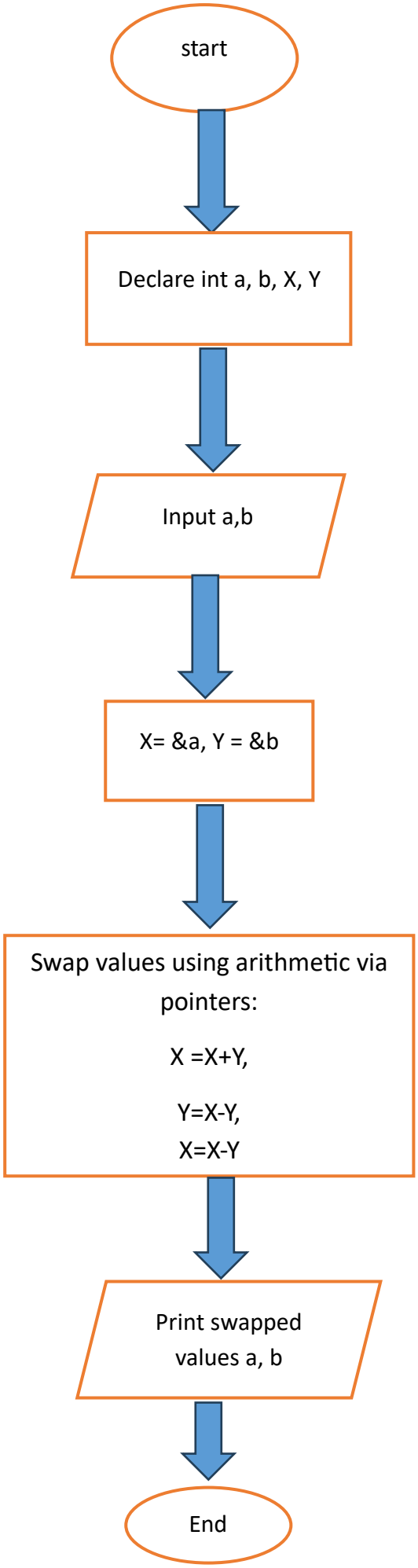
Step4: Declare two pointers X and Y, point to a and b

Step5: Swap values using arithmetic and pointers

Steps6: Print swapped values

Step7: End

Flowchart:


start
Declare int a, b, X, Y
Input a,b
X= &a, Y = &b
Swap values using arithmetic via pointers:
X =X+Y,
Y=X-Y,
X=X-Y
Print swapped values a, b
End

Source code:

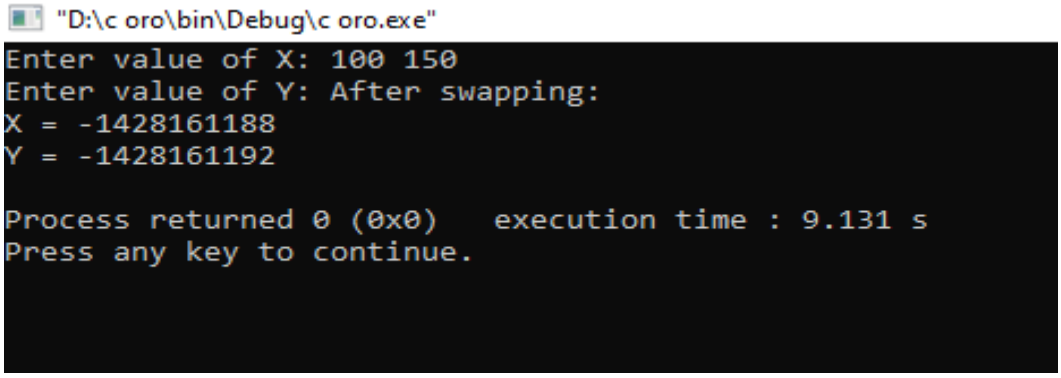
```c
#include <stdio.h>
int main() {
    int a, b;
    int *X, *Y;
    printf("Enter value of X: ");
    scanf("%d", &X);
    printf("Enter value of Y: ");
    scanf("%d", &Y);

    X = &a;
    Y = &b;

    *X = *X + *Y;
    *Y = *X - *Y;
    *X = *X - *Y;

    printf("After swapping:\n");
    printf("X = %d\n", X);
    printf("Y = %d\n", Y);

    return 0;
}
```



Discussion & Conclusion: This program swaps two numbers without using a third variable, by using pointer arithmetic. It accesses and modifies the values directly using their addresses. This method is memory-efficient and works well for small numbers. But with very large values, the addition step might cause overflow, so it should be used carefully.