MIT-WORLD PEACE UNIVERSITY

F. Y. B. Tech

Trimester: I/II/III Subject: Programming and Problem Solving

Name: Krishnaraj Thadesar **Division:** 9

Roll No.: 109054 **Batch:** I3

Experiment No.: 8

Name of the Experiment: Write a C Program to swap numbers using pointers.

Performed on: 17rd February 2022

Submitted on: 24st February 2022

AIM: Write an algorithm and draw a flowchart to Make C program to Swap 2 numbers using pointers.

OBJECTIVE:

1. To study the concept of pointers and function call—call by reference in C language

THEORY:

Pointers:

Pointers are variables which hold addresses of other variables. A function can be called either by value or by reference.

Syntax:

```
data-type *pointer_name;
```

Data type of a pointer must be same as the data type of a variable to which the pointer variable is pointing.

The two types of function calls—call by value and call by reference.

Arguments can generally be passed to functions in one of the two ways:

Sending the values of the arguments

In call by value method, the 'value' of each of the actual arguments in the calling function is copied into corresponding formal arguments of the called function. With this method the changes made to the formal arguments in the called function have no effect on the values of actual arguments in the calling function.

Sending the addresses of the arguments

In this method (call by reference) the addresses of actual arguments in the calling function are copied into formal arguments of the called function. This means that using these addresses we would have an access to the actual arguments and hence we would be able to manipulate them

PLATFORM: *Arch Linux 64 Bit with gcc/g++ compiler.*

ALGORITHM:

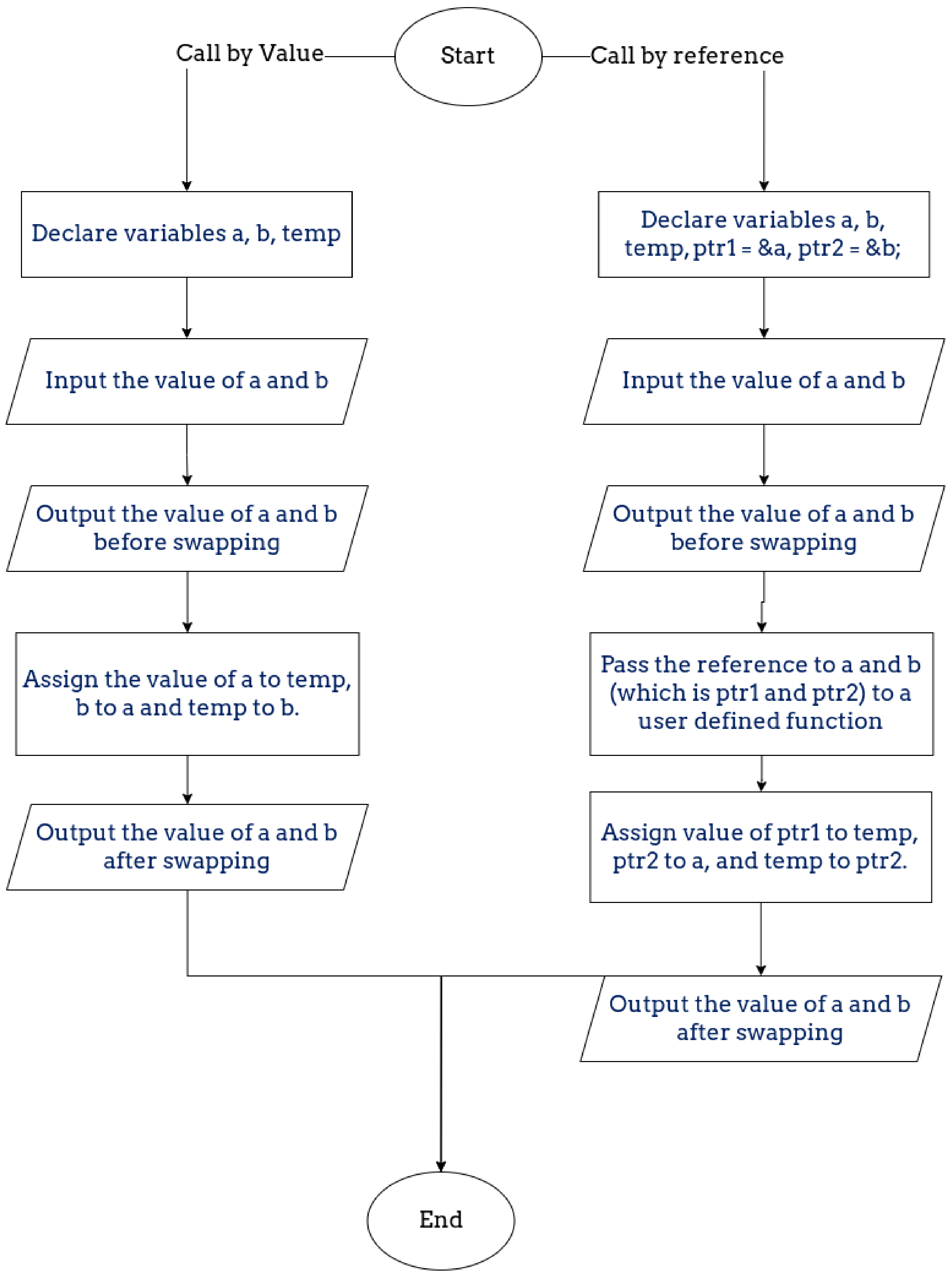
Call by value

```
Step 1: Start
Step 2: Declare variables a, b, temp
Step 3: Input the value of a and b
Step 4: Output the value of a and b before swapping
Step 5: Assign the value of a to temp, b to a and temp to b.
Step 6: Output the values of a and b after swapping.
```

Call by Reference

```
Step 1: Start
Step 2: Declare variables a, b, temp, ptr1 = &a, ptr2 = &b;
Step 3: Input the value of a and b.
Step 4: Output the value of a and b before swapping
Step 5: Pass the reference to a and b (which is ptr1 and ptr2) to a user defined function
Step 6: Assign value of ptr1 to temp, ptr2 to a, and temp to ptr2.
Step 7: Output the values of a and b after swapping.
```

Flowchart:


Start
Call by Value
Call by reference
Declare variables a, b, temp
Input the value of a and b
Output the value of a and b before swapping
Assign the value of a to temp, b to a and temp to b.
Output the value of a and b after swapping
Declare variables a, b, temp, ptr1 = &a, ptr2 = &b;
Input the value of a and b
Output the value of a and b before swapping
Pass the reference to a and b (which is ptr1 and ptr2) to a user defined function
Assign value of ptr1 to temp, ptr2 to a, and temp to ptr2.
Output the value of a and b after swapping
End

CODE:

```c
// Program to swap 2 numbers using pointers and functions

#include <stdio.h>
void swap(int *a, int *b)
{
    int temp = 0;
    temp = *a;
    *a = *b;
    *b = temp;
}

int main()
{
    int a, b;
    int *p1 = &a, *p2 = &b;
    printf("Enter the value of a: ");
    scanf("%d", p1);
    printf("Enter the value of b: ");
    scanf("%d", p2);
    printf("The value of a and b before swapping is %d and %d", a, b);
    swap(p1, p2);
    printf("\nThe value of a and b after swapping is %d and %d", a, b);
    return 0;
}
```

OUTPUT

Addition

```
Enter the value of a: 1
Enter the value of b: 2
The value of a and b before swapping is 1 and 2
The value of a and b after swapping is 2 and 1
```

CONCLUSION:

Understand the concept of pointers and function call—call by reference in C language.

FAQs:

Q1. What is a pointer? Write the syntax for pointer declaration and initialization?

Pointers are variables which hold addresses of other variables. A function can be called either by value or by reference.
Syntax for declaration:
data-type *pointer_name;

Syntax for Initialization:
data-type *pointer_name = &data_type_variable

Data type of a pointer must be same as the data type of a variable to which the pointer variable is pointing.

Q2. What do you mean be call by value and call by reference?

Call By Value: In this parameter passing method, values of actual parameters are copied to function's formal parameters and the two types of parameters are stored in different memory locations. So any changes made inside functions are not reflected in actual parameters of the caller.

Call by Reference: Both the actual and formal parameters refer to the same locations, so any changes made inside the function are actually reflected in actual parameters of the caller.

Q3. Explain difference between call by value and call by reference?

Call by value in C

1. In call by value method, the value of the actual parameters is copied into the formal parameters. In other words, we can say that the value of the variable is used in the function call in the call by value method.

2. In call by value method, we can not modify the value of the actual parameter by the formal parameter.

3. In call by value, different memory is allocated for actual and formal parameters since the value of the actual parameter is copied into the formal parameter.

4. The actual parameter is the argument which is used in the function call whereas formal parameter is the argument which is used in the function definition.

Call by reference in C

1. In call by reference, the address of the variable is passed into the function call as the actual parameter.

2. The value of the actual parameters can be modified by changing the formal parameters since the address of the actual parameters is passed.

3. In call by reference, the memory allocation is similar for both formal parameters and actual parameters. All the operations in the function are performed on the value stored at the address of the actual parameters, and the modified value gets stored at the same address.