
ASSIGNMENT NO:- 04

PROGRAM:

```
1#include <stdio.h>
2
3// Function to swap values by value
4void swapByValue(int a, int b) {
5    int temp = a;
6    a = b;
7    b = temp;
8}
9
10// Function to swap values by reference
11void swapByReference(int *a, int *b) {
12    int temp = *a;
13    *a = *b;
14    *b = temp;
15}
16
17int main() {
18    int num1, num2;
19
20    // Input the two numbers
21    printf("Enter the first number: ");
22    scanf("%d", &num1);
23    printf("Enter the second number: ");
24    scanf("%d", &num2);
25
26    // Display the original values
27    printf("Original values: num1 = %d, num2 = %d\n", num1, num2);
28
29    // Swap values by value
30    swapByValue(num1, num2);
31
32    // Display the swapped values
33    printf("After swapping by value: num1 = %d, num2 = %d\n", num1, num2);
34
35    // Swap values by reference
36    swapByReference(&num1, &num2);
37
38    // Display the swapped values
39    printf("After swapping by reference: num1 = %d, num2 = %d\n", num1, num2);
40
41    return 0;
42}
```

OUTPUT:

```
student@cc-91:~$ gcc swap.c
student@cc-91:~$ ./a.out
Enter the first number: 22
Enter the second number: 44
Original values: num1 = 22, num2 = 44
After swapping by value: num1 = 22, num2 = 44
After swapping by reference: num1 = 44, num2 = 22
```

```
student@cc-91:~$ cat swap.s
.file    "swap.c"
.text
.globl   swapByValue
.type    swapByValue, @function
swapByValue:
.LFB0:
.cfi_startproc
endbr64
pushq    %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq     %rsp, %rbp
.cfi_def_cfa_register 6
movl     %edi, -20(%rbp)
movl     %esi, -24(%rbp)
movl     -20(%rbp), %eax
movl     %eax, -4(%rbp)
movl     -24(%rbp), %eax
movl     %eax, -20(%rbp)
movl     -4(%rbp), %eax
movl     %eax, -24(%rbp)
nop
popq     %rbp
.cfi_def_cfa 7, 8
ret
.cfi_endproc
.LFE0:
.size    swapByValue, .-swapByValue
.globl   swapByReference
.type    swapByReference, @function
```

```
swapByReference:
.LFB1:
    .cfi_startproc
    endbr64
    pushq   %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq    %rsp, %rbp
    .cfi_def_cfa_register 6
    movq    %rdi, -24(%rbp)
    movq    %rsi, -32(%rbp)
    movq    -24(%rbp), %rax
    movl    (%rax), %eax
    movl    %eax, -4(%rbp)
    movq    -32(%rbp), %rax
    movl    (%rax), %edx
    movq    -24(%rbp), %rax
    movl    %edx, (%rax)
    movq    -32(%rbp), %rax
    movl    -4(%rbp), %edx
    movl    %edx, (%rax)
    nop
    popq    %rbp
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
```

DIFFERENCE BETWEEN CALL BY VALUE AND CALL BY REFERENCE:

1. "swapByValue" receives two 32-bit integers as arguments and stores them in local stack variables using movl instructions.
2. "swapByReference" receives two 64-bit pointers as arguments and stores them in local stack variables using movq instructions.
3. In "swapByValue," the integers are moved into and out of local stack variables, while in "swapByReference," pointer dereferencing is used to load and store values in memory.