

| **TITLE:**  Virtual Lab experiment on Call by reference. |
| --- |

**AIM:** Virtual Lab experiment on Call by reference

<http://cse02-iiith.vlabs.ac.in/>  
<http://cse02-iiith.vlabs.ac.in/exp8/simulation/CallByReferencePointers/index.html>  
Program to swap two number without using third variable using Call by reference.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

**CO4:** Design modular programs using functions and demonstrate the concept of pointers and file handling.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Programming in C, second edition, Pradeep Dey and Manas Ghosh, Oxford University Press.
2. Programming in ANSI C, fifth edition, E Balagurusamy, Tata McGraw Hill.
3. Introduction to programming and problem solving , G. Michael Schneider ,Wiley India edition.
4. [**http://cse.iitkgp.ac.in/~rkumar/pds-vlab/**](http://cse.iitkgp.ac.in/~rkumar/pds-vlab/)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem Definition:**

The **call by reference** method of passing arguments to a function copies the address of an argument into the formal parameter. Inside the function, the address is used to access the actual argument used in the call.

In the program, a function called swap ( ) is used in which the address is used to access the actual argument.

**Algorithm:**

1: Start

2: Declare variable A and B as Integer data type and assign value 5 and 9 respectively.

3: Print A

4: Print B

5: Call function swap() and pass A and B as agreements

6: Function call. Local variable type pointer Pa and Pb get assigned the address of A and B respectively.

7: A local variable is declared temp and assigned the value at Address Pa

8: Variable Pa is assigned the value at Address Pb

9: Variable Pb is assigned the value at temp

10: Print A

11: Print B

12: Stop

**Implementation details:**

**#include<stdio.h>**

**void main(){**

**int A = 5, B = 9;**

**printf('Value of A is %d\n',A);**

**printf('Value of B is %d\n',B);**

**swap( &A , &B );**

**printf('Value of A after swapping is %d\n',A);**

**printf('Value of B after swapping is %d\n',B);**

**}**

**void swap( int \*Pa , int \*Pb){**

**int temp = \*Pa;**

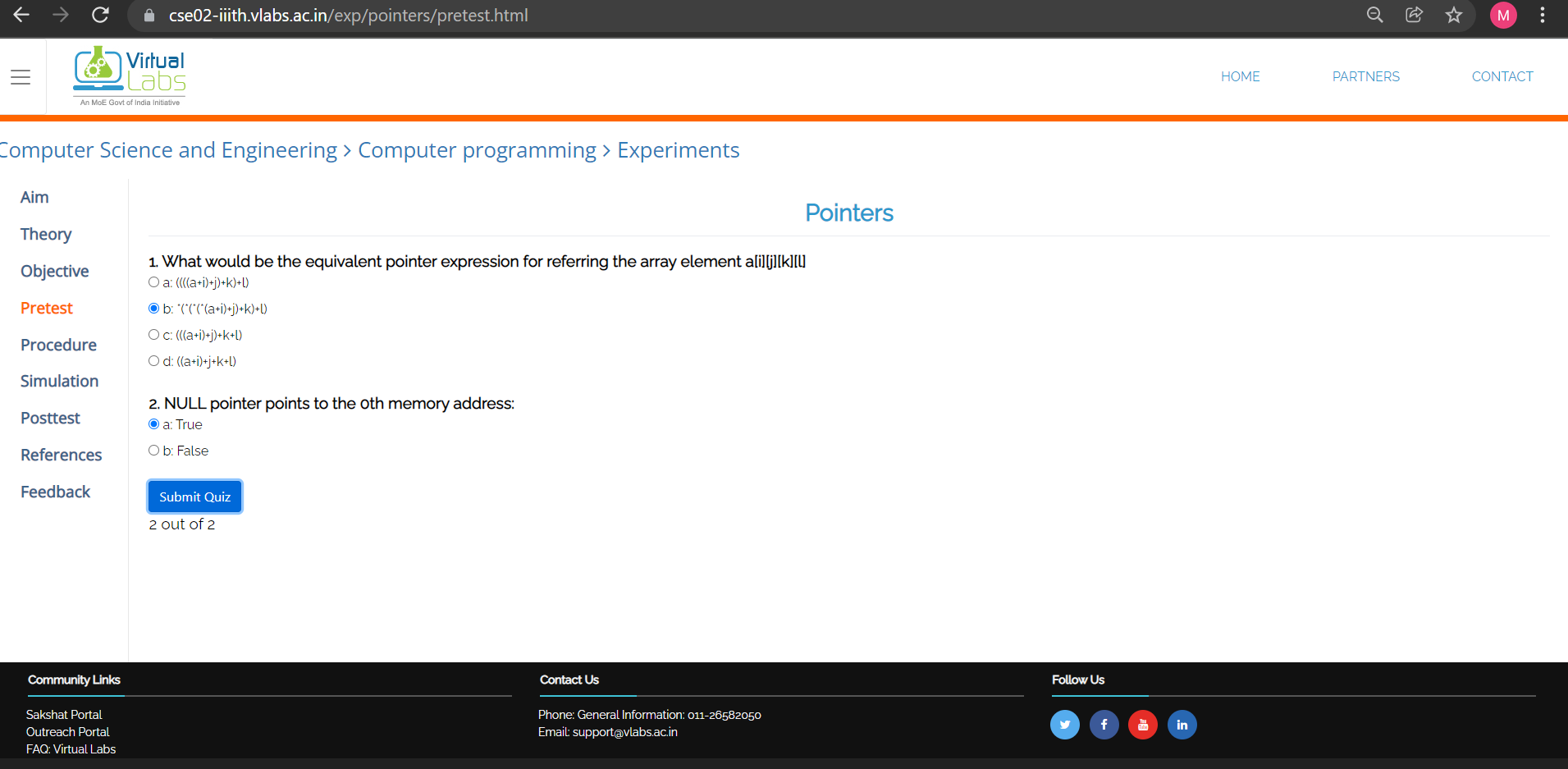
**\*Pa = \*Pb;**

**\*Pb = temp;**

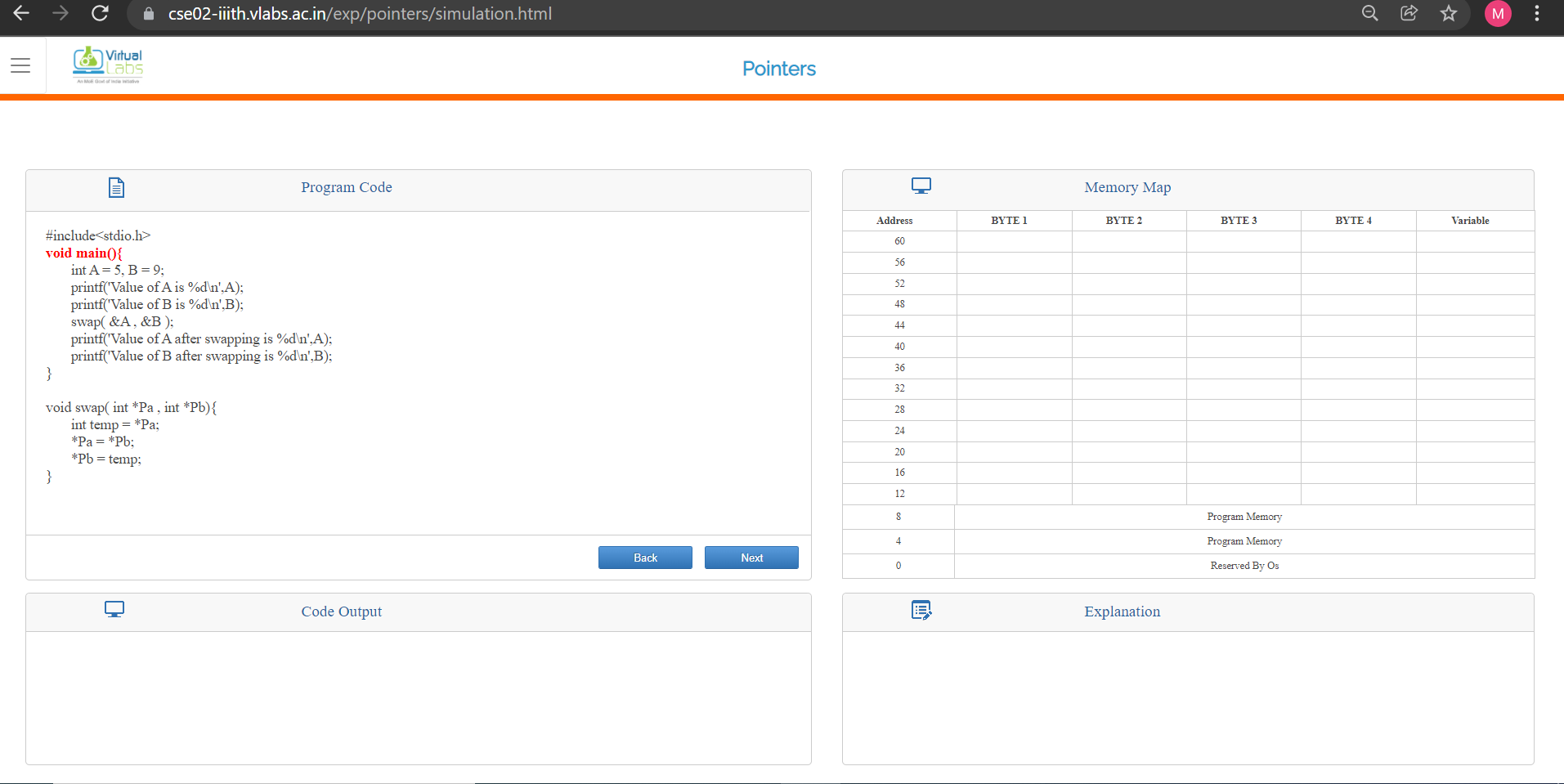
**}**

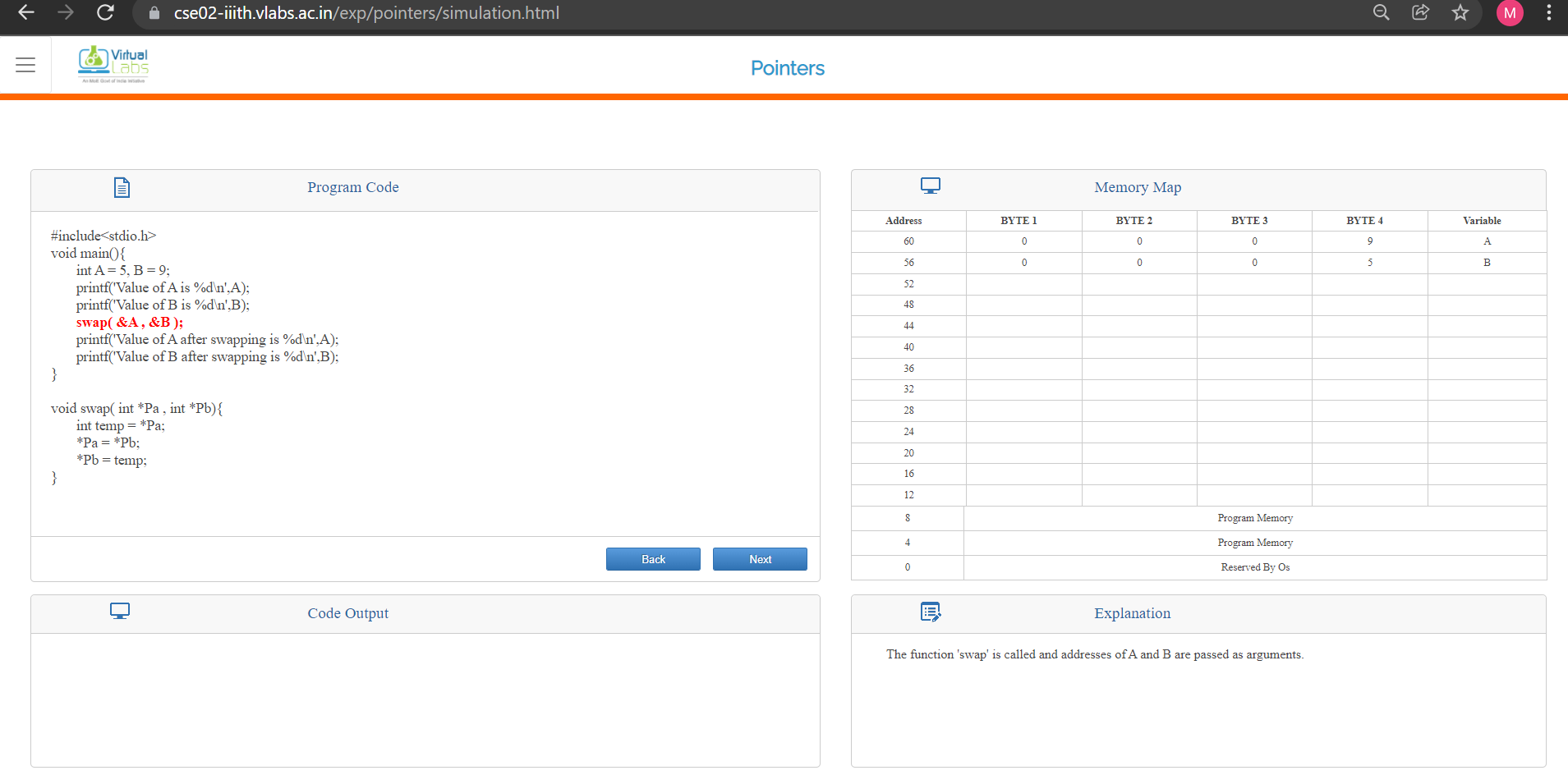
**Output(s):**

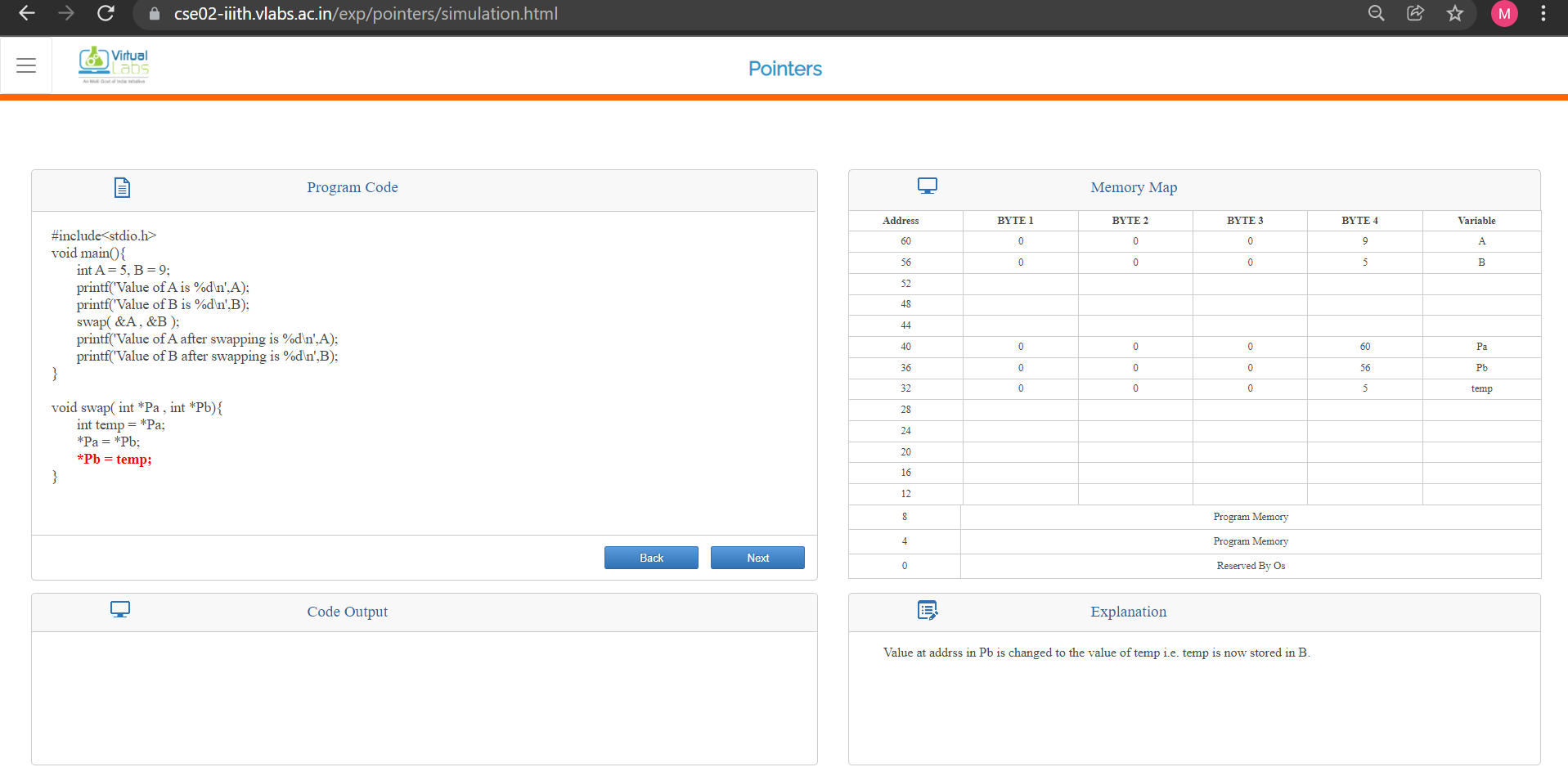
**Pretest:-**

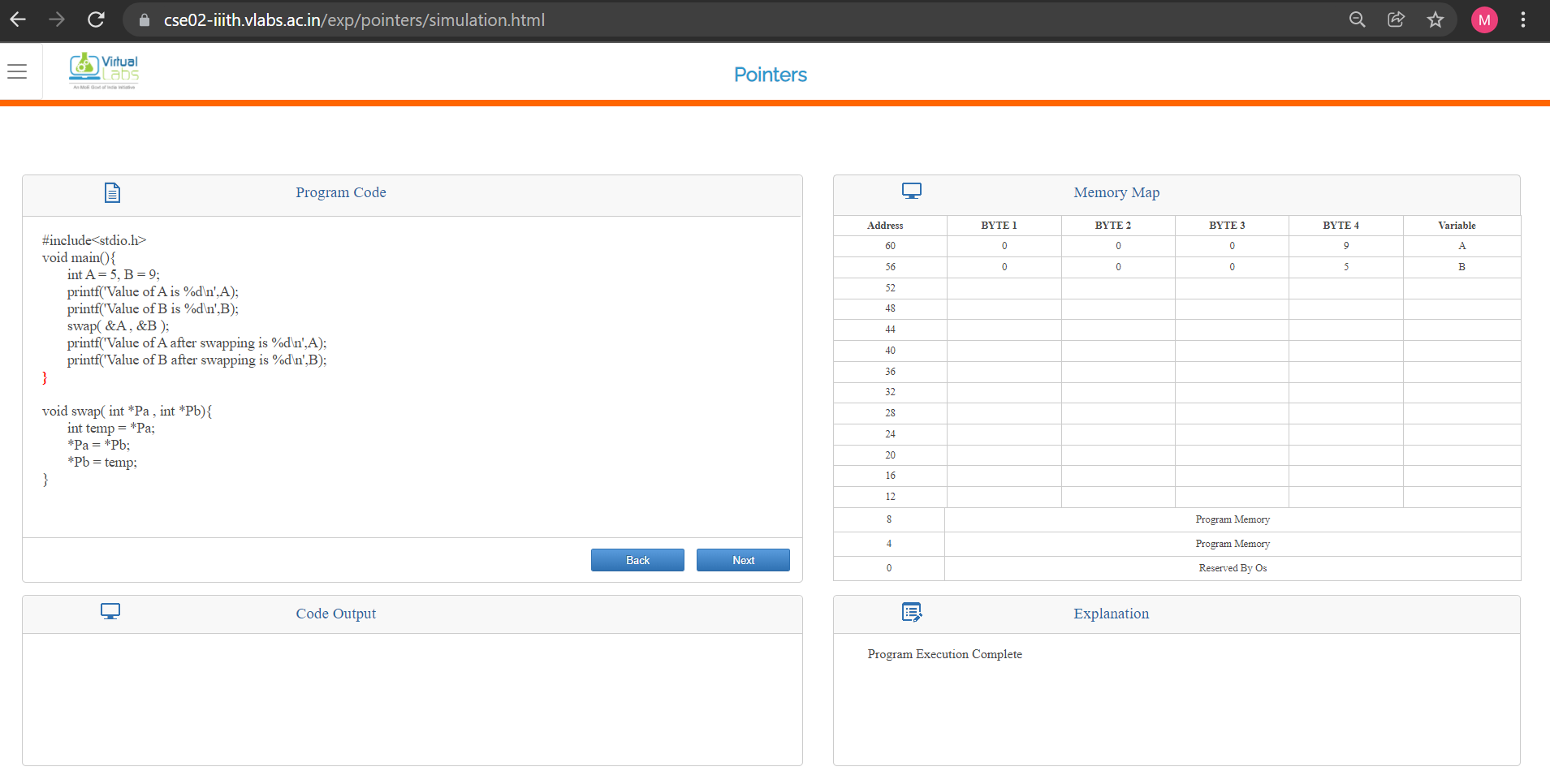
****

Call by Reference:-

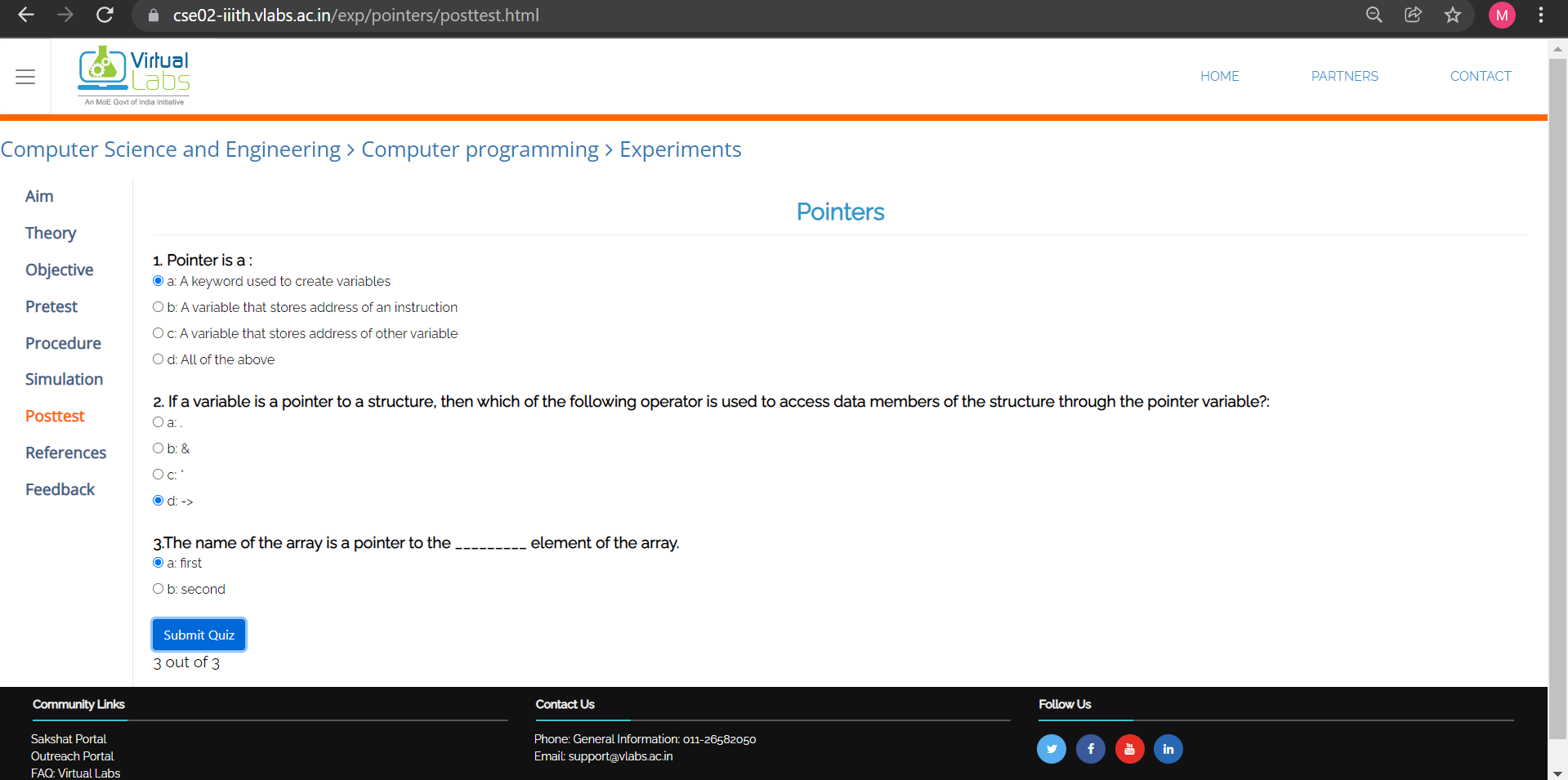








Posttest:-



**Conclusion:**

We learnt the execution of the call by reference program. We were also able to design modular programs using functions and demonstrate the concept of pointers

**Post Lab Descriptive Questions**

1. **Difference between call by value and call by reference in C.**

**Ans:-**

| **Call By Value** | **Call By Reference** |
| --- | --- |
| While calling a function, we pass values of variables to it. Such functions are known as Call By Values. | While calling a function, we pass address of variables to the function known as “Call By References |
| In this method, the value of each variable in calling function is copied into corresponding dummy variables of the called function. | In this method, the address of actual variables in the calling function are copied into the dummy variables of the called function. |
| Changes made in a copy of variable never modify the value of variable outside the function | Change in the variable also affects the value of the variable outside the function |
| In call by values we cannot alter the values of actual variables through function calls. | In call by reference we can alter the values of variables through function calls. |

1. **What would be the output of the following program:**

main( )

{

float a = 13.5 ;

float \*b, \*c ;

b = &a ; /\* suppose address of a is 1006 \*/

c = b ;

printf ( "\n%u %u %u", &a, b, c ) ;

printf ( "\n%f %f %f %f %f", a, \*(&a), \*&a, \*b, \*c ) ;

}

Ans:-

**Output is:**

1006 1006 1006

13.500000 13.500000 13.500000 13.500000 13.500000

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**