**Batch:C2-2 Roll No.: 16010122109**

**Experiment / assignment / tutorial No. 7**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **TITLE: Using virtual labs to understand the concept of matrix multiplication, call by reference** |

**AIM:** Use of virtual labs to understand the concepts and theory with examples and verify the same with practice questions. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

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

**Books/ Journals/ Websites referred:**

1. Programming in ANSI C, E. Balagurusamy, 7 th Edition, 2016, McGraw-Hill Education, India.
2. Structured Programming Approach, Pradeep Dey and Manas Ghosh, 1 st Edition, 2016, Oxford University Press, India.
3. Let Us C, Yashwant Kanetkar, 15th Edition, 2016, BPB Publications, India.

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

**Problem Definition:**

Virtual Lab experiment on matrix multiplication

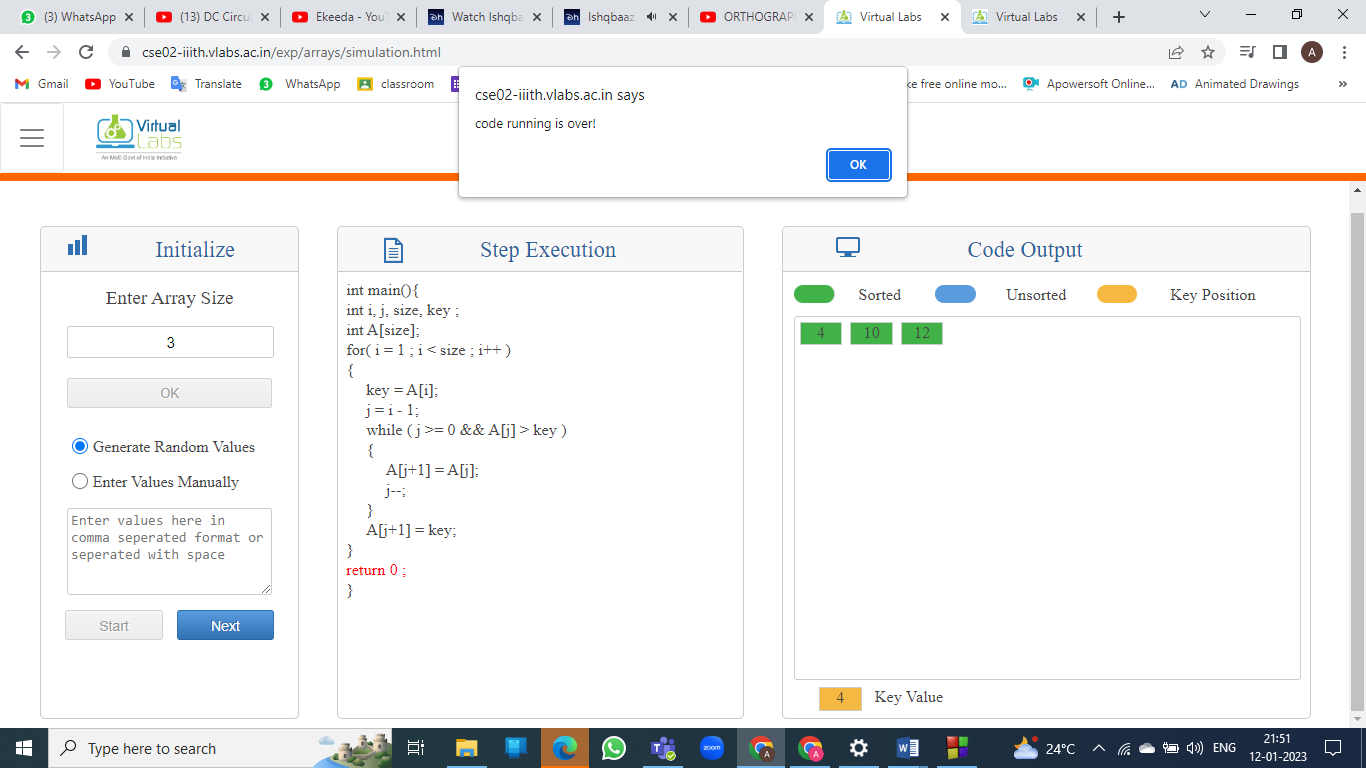
<https://cse02-iiith.vlabs.ac.in/exp/arrays/simulation.html>

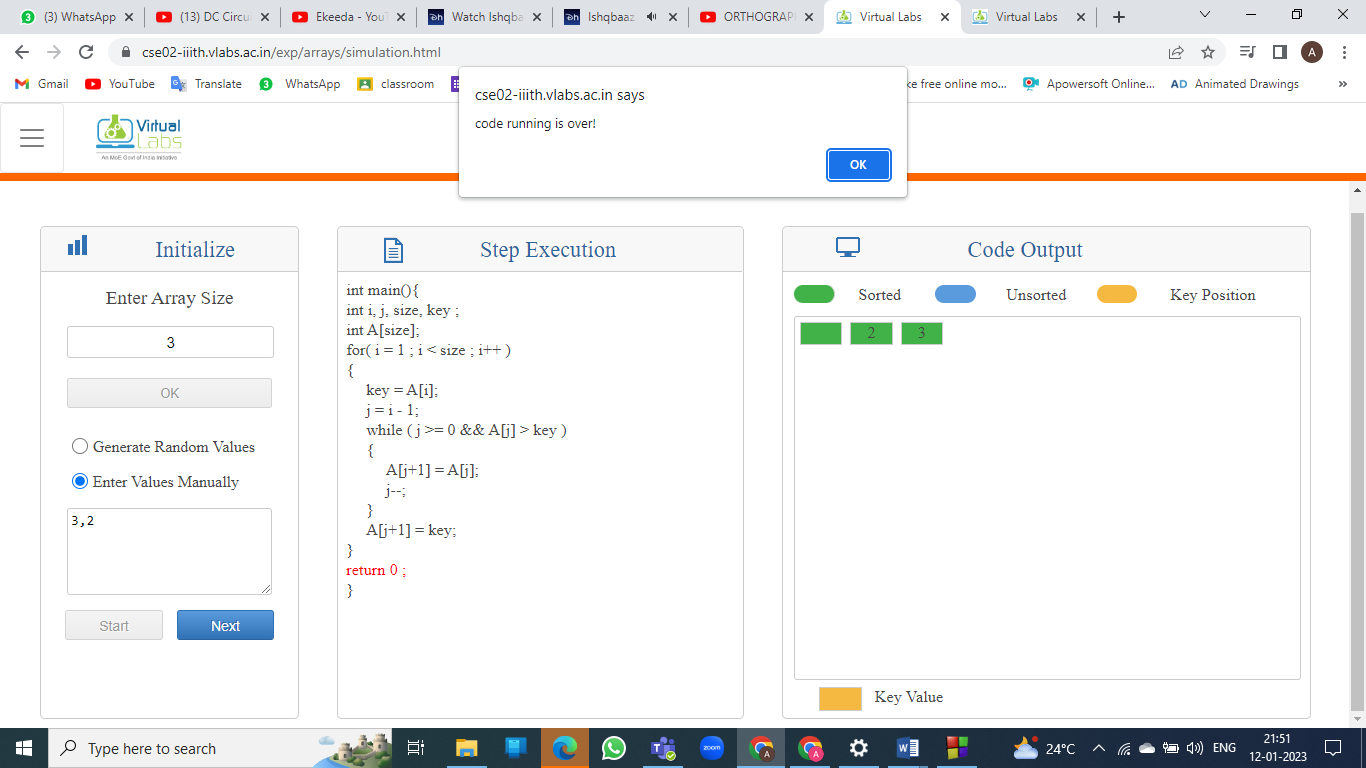
Virtual Lab experiment on Call by reference

<https://cse02-iiith.vlabs.ac.in/exp/pointers/procedure.html>

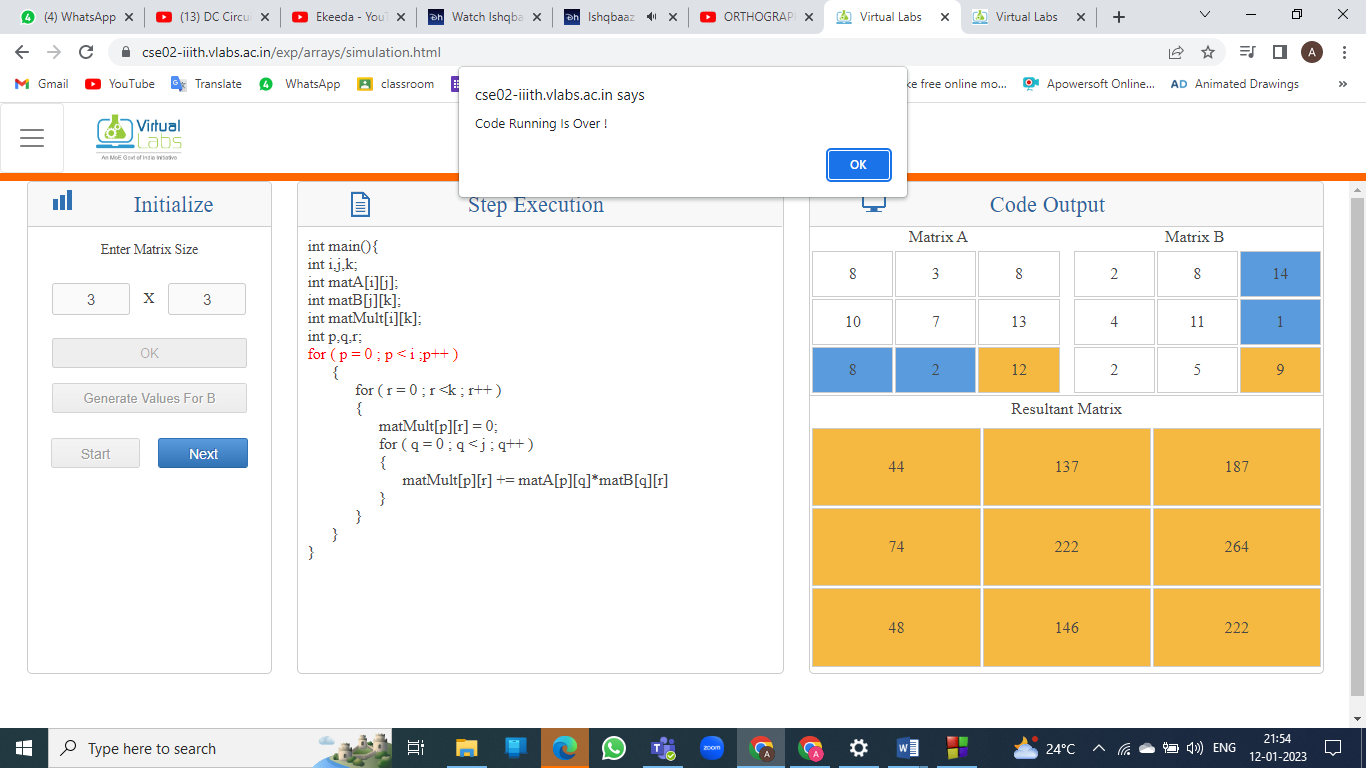
Program to swap two number without using third variable using Call by reference.

**Implementation details/** **Simulation screenshots and outputs:**

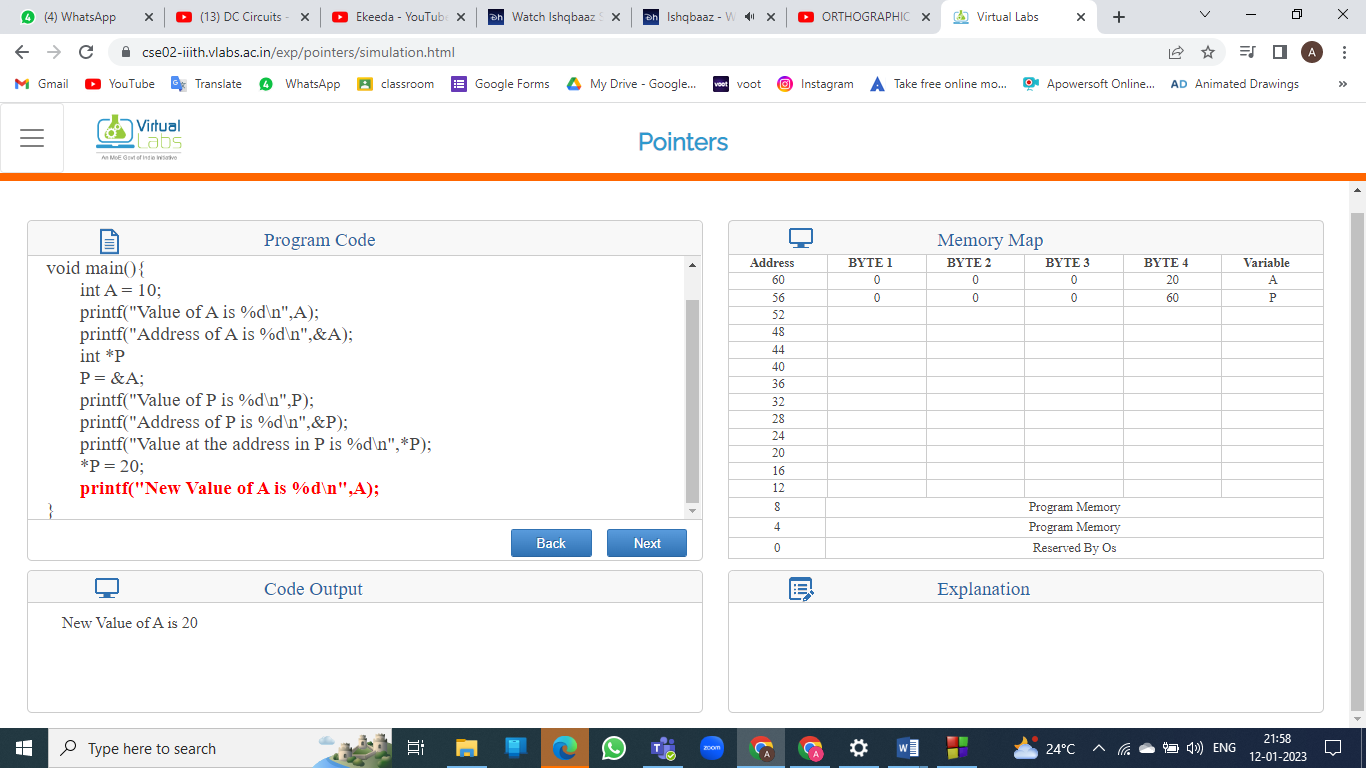




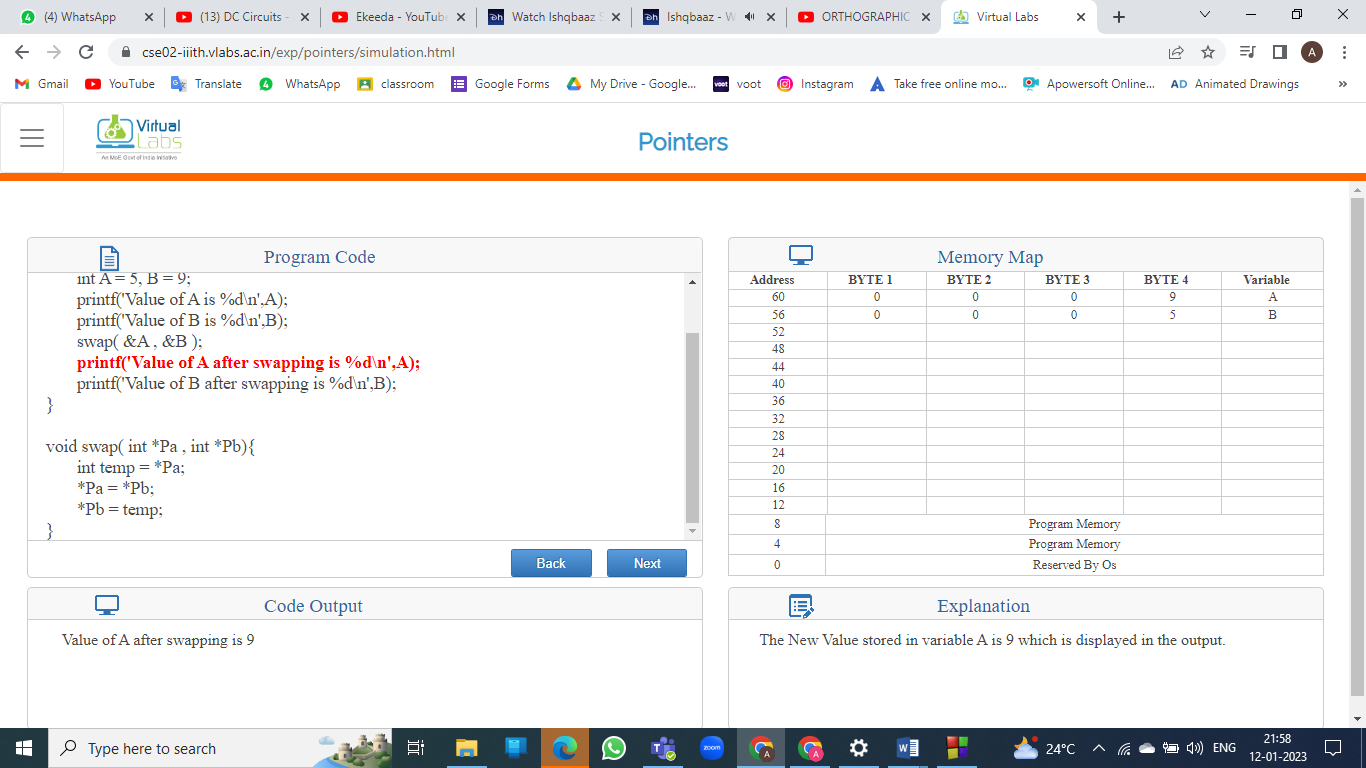
2D MATRIX:

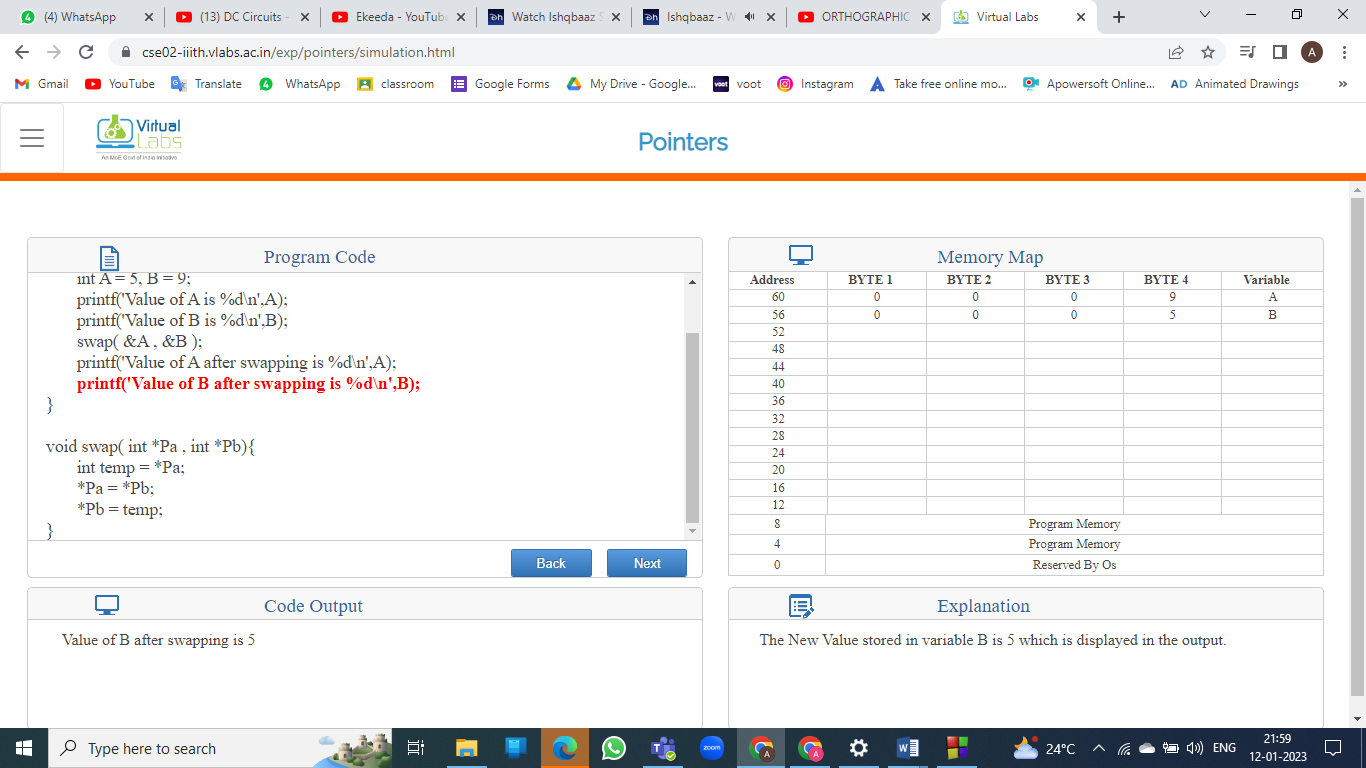


CALL BY VALUE:



CALL BY REFERENCE :





SWAP TWO NUMBERS :

#include<stdio.h>

int main()

{

int x, y;

printf("Input value for x & y: \n");

scanf("%d%d",&x,&y);

printf("Before swapping the value of x & y: %d %d",x,y);

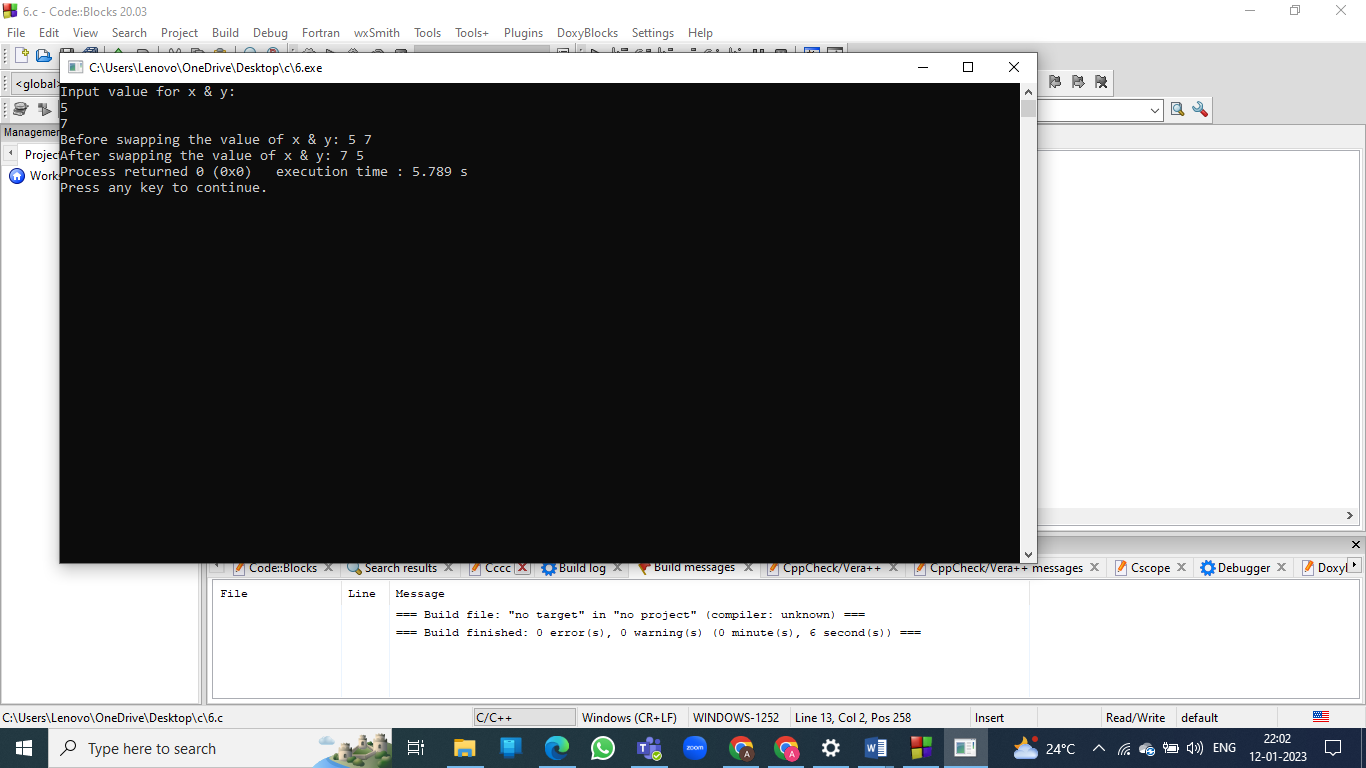
x=x+y;

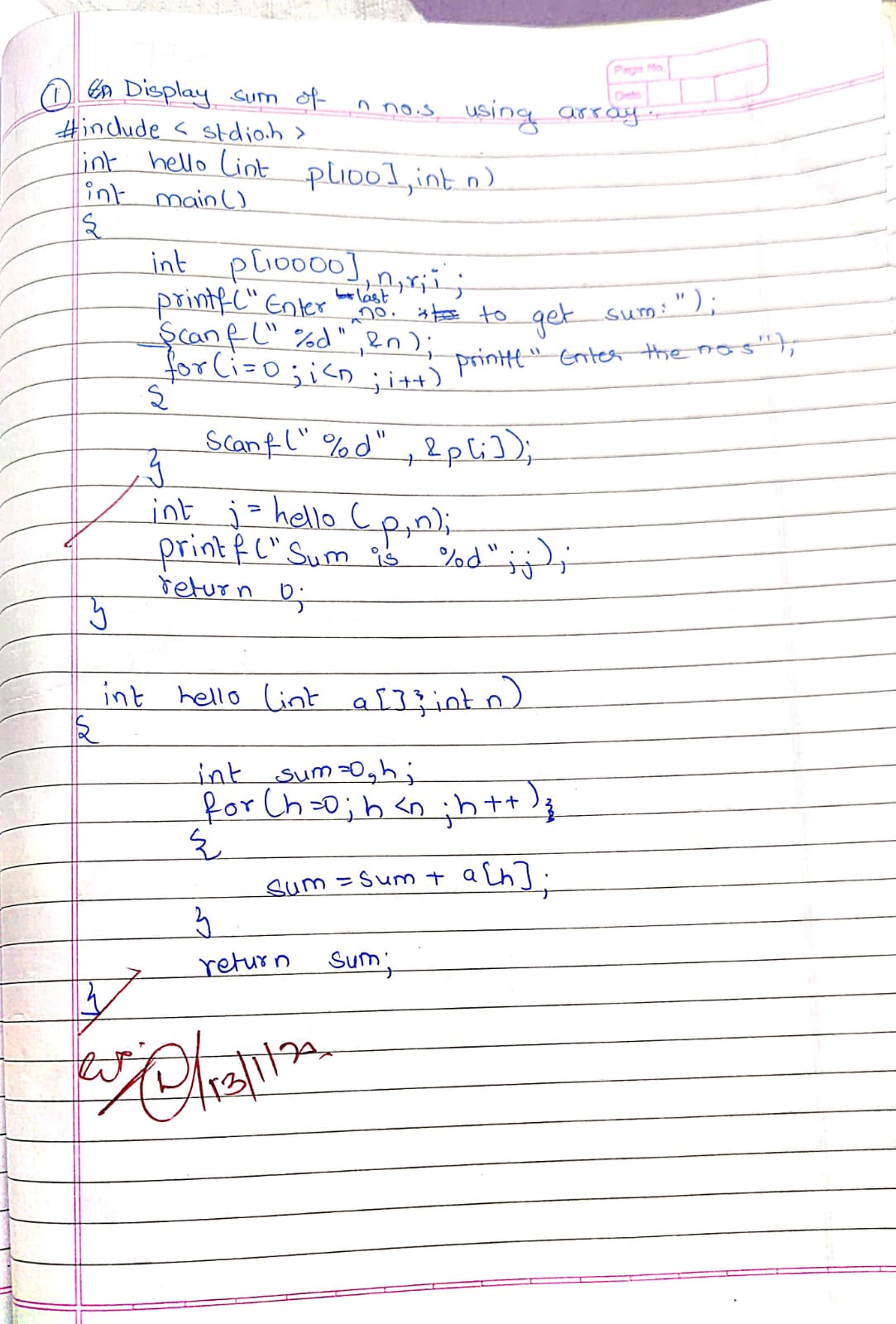
y=x-y;

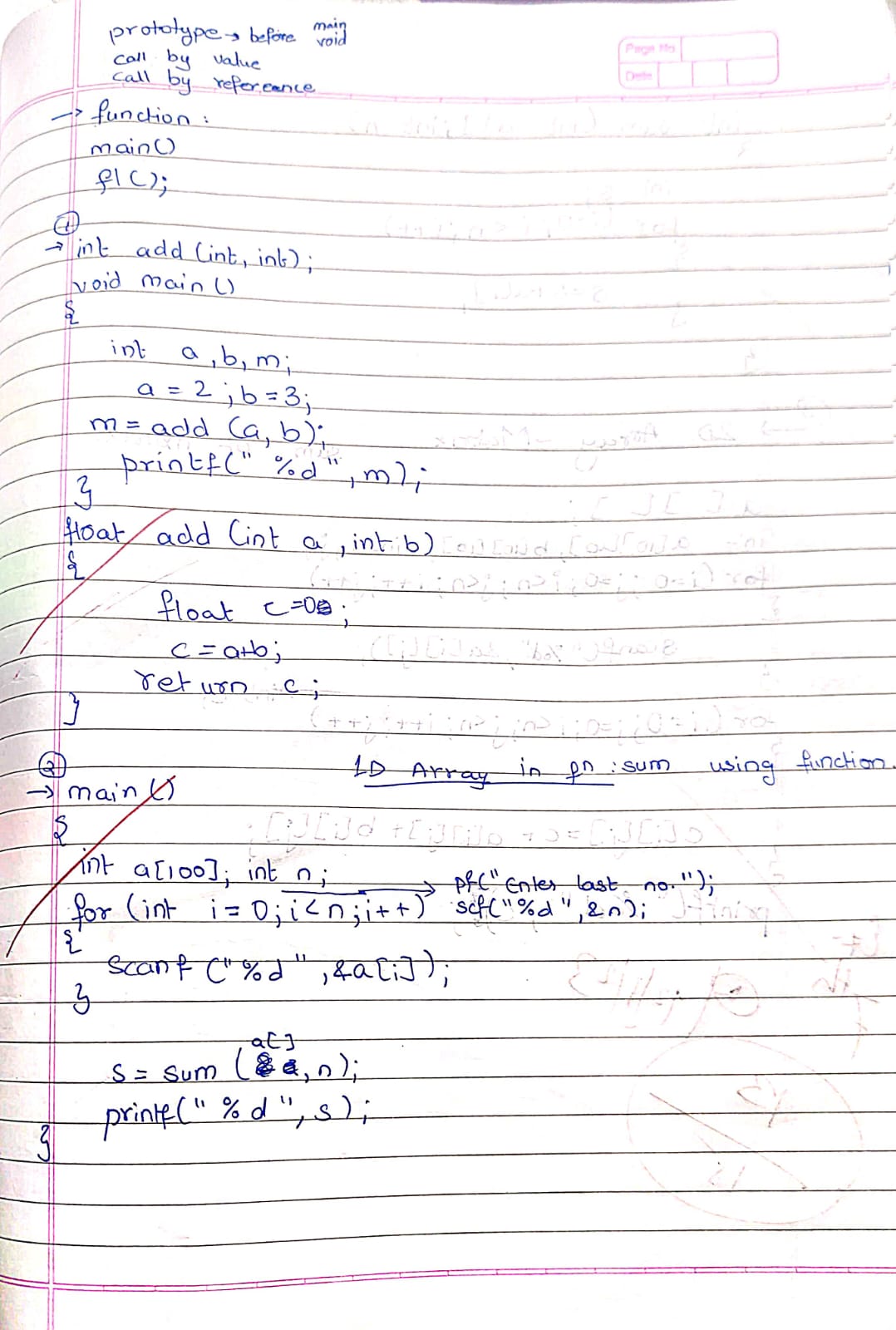
x=x-y;

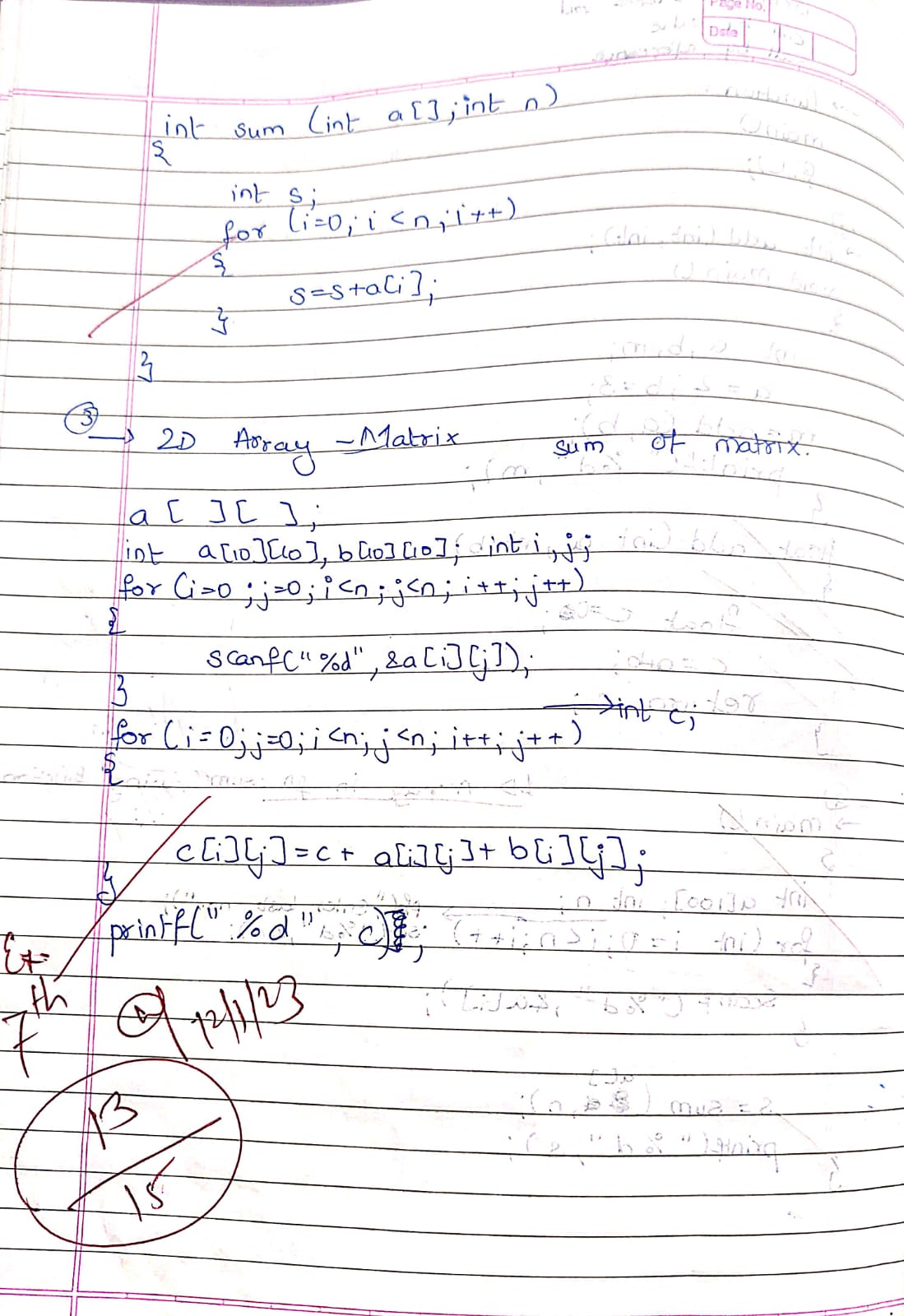
printf("\nAfter swapping the value of x & y: %d %d",x,y);

}







****

**Conclusion and your take away after performing the virtual lab experiment: -**

We learnt how step by step sorting is done in 1-D array and how to step by step perform matrix multiplication.

**Post Lab Descriptive Questions**

1. **Differentiate between Call by Value and Call by Reference.**
2. **Try to understand the working of pointers by Running the following code and noting down the output.**

main( )

{

int i = 3 ;

int \*j ;

j = &i ;

printf ( "\nAddress of i = %u", &i ) ;

printf ( "\nAddress of i = %u", j ) ;

printf ( "\nAddress of j = %u", &j ) ;

printf ( "\nValue of j = %u", j ) ;

printf ( "\nValue of i = %d", i ) ;

printf ( "\nValue of i = %d", \*( &i ) ) ;

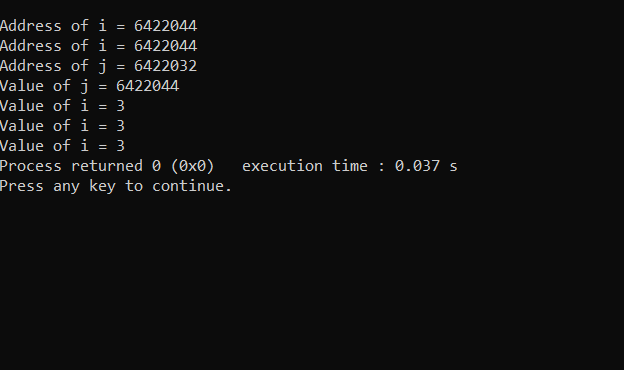
printf ( "\nValue of i = %d", \*j ) ;

}

1 ]

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Call by value** | **Call by reference** |
| Definition | While calling a function, when you pass values by copying variables, it is known as “Call By Values.” | While calling a function, in programming language instead of copying the values of variables, the address of the variables is used it is known as “Call By References. |
| Arguments | In this method, a copy of the variable is passed. | In this method, a variable itself is passed. |
| Effect | 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. |
| Alteration of value | Does not allow you to make any changes in the actual variables. | Allows you to make changes in the values of variables by using function calls. |
| Passing of variable | Values of variables are passed using a straightforward method. | Pointer variables are required to store the address of variables. |
| Value modification | Original value not modified. | The original value is modified. |
| Memory Location | Actual and formal arguments will be created in different memory location | Actual and formal arguments will be created in the same memory location |
| Safety | Actual arguments remain safe as they cannot be modified accidentally. | Actual arguments are not Safe. They can be accidentally modified, so you need to handle arguments operations carefully. |

**2 ]**



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