**Batch: C3-3 Roll No.: 16010122221**

**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:**

The programs to sort a 1D array and to multiply two 2D arrays were learnt using online virtual simulation lab. The simulation worked perfectly.

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

**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>

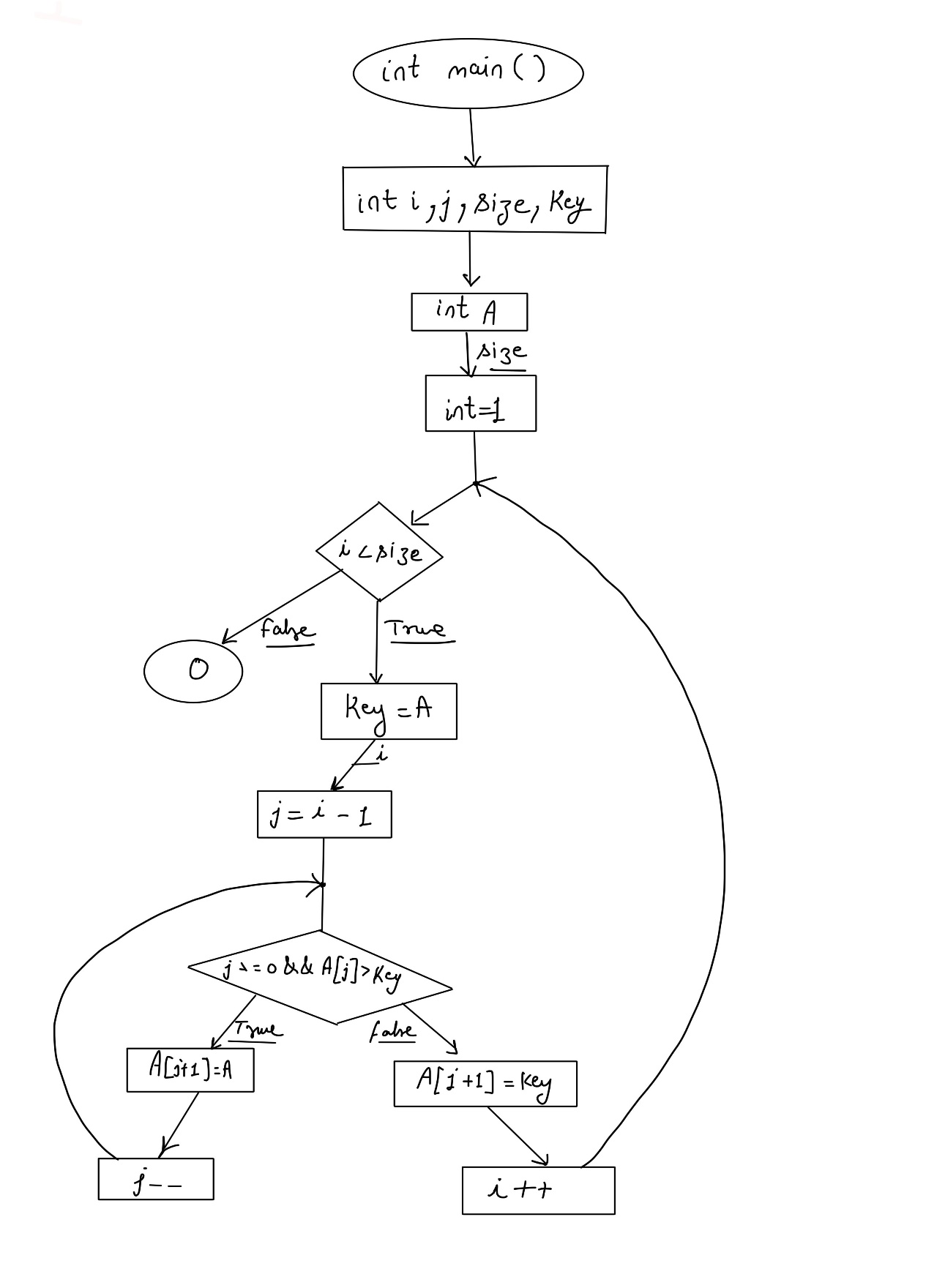
Virtual Lab experiment on Call by reference

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

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

Algorithm:-

1D:



**2D:**

Diagram

Description automatically generatedDiagram

Description automatically generated



**Implementation details/** **Simulation screenshots:**

1D:

int main(){

int i, j, size, key ;

int A[size];

for( i = 1 ; i < size ; i++ )

{

key = A[i];

j = i - 1;

while ( j >= 0 && A[j] > key )

{

A[j+1] = A[j];

j--;

}

A[j+1] = key;

}

return 0 ;

}

2D:

int main(){

int i,j,k;

int matA[i][j];

int matB[j][k];

int matMult[i][k];

int p,q,r;

for ( p = 0 ; p < i ;p++ )

{

for ( r = 0 ; r <k ; r++ )

{

matMult[p][r] = 0;

for ( q = 0 ; q < j ; q++ )

{

matMult[p][r] += matA[p][q]\*matB[q][r]

}

}

}

}

**Output(s)/Post-test Screenshots:**

1D:

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

2D:

A picture containing chart

Description automatically generated

Chart

Description automatically generated with medium confidence

POST TEST:

Graphical user interface, text, application, email

Description automatically generated

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

The programs to sort a 1D array and to multiply two 2D arrays were learnt using online virtual simulation lab. The simulation worked perfectly.

**Post Lab Descriptive Questions**

1. **Differentiate between Call by Value and Call by Reference.**

* In the Call by value method original value is not modified, whereas in the Call by reference method, the original value is modified.
* In Call by value, a copy of the variable is passed, whereas in Call by reference, a variable itself is passed.
* In Call by value, actual and formal arguments will be created in different memory locations, whereas in Call by reference, actual and formal arguments will be created in the same memory location.
* Call by value is the default method in programming languages like C++, PHP, Visual Basic NET, and C#, whereas Call by reference is supported only in Java language.
* Call by Value variables is passed using a straightforward method, whereas Call by Reference pointers are required to store the address of variables.

1. **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 ) ;

}

OUTPUT:

A picture containing text

Description automatically generated

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