

**Lab's Scope:**

- **Two-dimensional Arrays**
- 

**Problem 1:**

Write a program declaring a 2D array with 3 rows and 5 columns. Initialize the array with integers. Your program should then find the largest and smallest values in the whole array and print them out.

**Solution:**

```
#include <iostream>
using namespace std;

int main() {
    const int ROWS = 3;
    const int COLS = 5;

    int arr[ROWS][COLS] = { {50,10,89,36,69},{87,94,52,37,78},{48,65,75,19,23} };
    int largest = arr[0][0];
    int smallest = arr[0][0];

    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLS; j++)
        {
            if (arr[i][j] > largest)
                largest = arr[i][j];
            if (arr[i][j] < smallest)
                smallest = arr[i][j];
        }
    }

    cout << "The largest number is : " << largest << endl;
    cout << "The smallest number is : " << smallest << endl;
    return 0;
}
```

**Problem 2:**

Write a program declaring a 2D array and initialize the array with integers. Your program should count the number of odd numbers in the array and print it out.

**Solution:**

```
#include <iostream>
using namespace std;

int main() {
    const int ROWS = 3;
    const int COLS = 5;

    int arr[ROWS][COLS] = {
        {50, 10, 89, 36, 69}, {87, 94, 52, 37, 78}, {48, 65, 75, 19, 23} };
    int count = 0;
    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLS; j++)
        {
            if ((arr[i][j] % 2) != 0)
                count++;
        }
    }
    cout << "The count of odd numbers is : " << count << endl;
    return 0;
}
```

**Problem 3:**

Write a program declaring a 2D array and initialize the array with integers. Your program should multiply all the elements in each column and store the results of multiplication in a one-dimensional array. Hint: Think carefully about how you will manage your index.

**Solution:**

```
#include <iostream>
using namespace std;

int main() {
    const int ROWS = 3;
    const int COLS = 5;
    int a[ROWS][COLS] = { {5, 3, 2, 1, 5}, {2, 3, 4, 3, 6}, {10, 9, 8, 4, 2} };
    int b[COLS];
    int product;
    int count = 0;
    for (int i = 0; i < COLS; i++)
    {
        product = 1;
        for (int j = 0; j < ROWS; j++)
        {
            product *= a[j][i];
        }
        b[i] = product;
        cout << b[i] << " ";
    }

    return 0;
}
```

**Problem 4:**

Write a program given a 2D array initialized with integers. Your program should ask the user to input a number and the program searches the array for this number. If the number is found your program should output the row and column of the element. If the element is not found, your program should print a message that the element is not found.

**Solution:**

```
#include <iostream>
using namespace std;

int main() {
    const int ROWS = 3;
    const int COLS = 5;
    int a[ROWS][COLS] = { {50,10,89,36,69},{87,94,52,37,78},{48,65,75,19,23} };
    int x;

    cout << "Enter the number you are searching for:";
    cin >> x;

    bool found = false;
    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLS; j++)
        {
            if (a[i][j] == x)
            {
                cout << "The element is found at row: " << i << endl;
                cout << "The element is found at column: " << j << endl;
                found = true;
                break;
            }
        }
    }

    if (found == false)
        cout << "The element is not found" << endl;
    return 0;
}
```

**Problem 5:**

Write a program that, given a 2D array where each row stores the grades of a student. Your program should calculate the average grade for each student.

**The output should be:**

Student 1 average grade is 50.5

Student 2 average grade is 60.0

Student average is 70.0

**Solution:**

```
#include <iostream>
using namespace std;

int main() {
    const int ROWS = 3;
    const int COLS = 5;
    int a[ROWS][COLS] = { {50,10,89,36,69},{87,94,52,37,78},{48,65,75,19,23} };
    float average;

    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLS; j++)
        {
            average += a[i][j];
        }
        average = average / COLS;
        cout << "Student " << i + 1 << " average grade is " << average << endl;
    }

    return 0;
}
```

**Problem 6:**

Given the below figure for multiplying two matrices. Write a program declaring and initializing two matrices of suitable sizes. Your program should multiply the two matrices and print out the output matrix.

$$\begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 4 \times 1 & 4 \times 2 & 4 \times 3 \\ 5 \times 1 & 5 \times 2 & 5 \times 3 \\ 6 \times 1 & 6 \times 2 & 6 \times 3 \end{bmatrix} = \begin{bmatrix} 4 & 8 & 12 \\ 5 & 10 & 15 \\ 6 & 12 & 18 \end{bmatrix}$$

Note that: multiplying a **3×1** by a **1×3** gets a **3×3** matrix result.

**Solution:**

```

#include <iostream>
using namespace std;

int main() {
    const int R1 = 3;
    const int C1 = 1;
    const int R2 = 1;
    const int C2 = 3;
    int a[R1][C1] = { 4,5,6 };
    int b[R2][C2] = { 1,2,3 };
    int c[R1][C2];

    // Multiplying matrix a and b and storing in array c.
    for (int i = 0; i < R1; i++)
    {
        for (int j = 0; j < C2; j++)
        {
            c[i][j] = 0;
            for (int k = 0; k < C1; k++)
            {
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }

    for (int i = 0; i < R1; i++)
    {
        for (int j = 0; j < C2; j++)
        {
            cout << c[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}

```

**Problem 7:**

A word puzzle game: Write a program that involves a 2D array of characters, the user will search for a word of 3 letters. Your program should search the array horizontally for the word and print out the row and column of the starting position of the word, or print “the word is not found” if the word is not in the array.

**Example:**

A	C	A	T	N
F	M	D	O	G
K	I	T	P	Q
M	S	A	F	A

Enter the word you are searching for:

DOG

The word is found starting from row 1 and column 2.

### Solution:

```
#include <iostream>
using namespace std;

int main() {
    const int ROWS = 4;
    const int COLS = 5;
    string word;

    char a[ROWS][COLS] = {
        {'A', 'C', 'A', 'T', 'N'}, {'F', 'M', 'D', 'O', 'G'}, {'K', 'I', 'T', 'P', 'Q'}, {'M', 'S', 'A', 'F', 'A'}
    };

    cout << "Enter the word you are searching for:" << endl;
    cin >> word;
    char letter1 = tolower(word.at(0));
    char letter2 = tolower(word.at(1));
    char letter3 = tolower(word.at(2));
    bool found = false;

    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLS - 2; j++)
        {
            if (tolower(a[i][j]) == letter1 && tolower(a[i][j + 1]) == letter2 &&
                tolower(a[i][j + 2]) == letter3)
            {
                cout << "The word is found starting from row " << i << " and column "
                << j << endl;
                found = true;
                break;
            }
        }
        if (found)
            break;
    }
}
```

### Problem 8:

Simple Sudoku game: Assume you have a 3x3 sudoku puzzle initialized as shown in the below figure, your program should insert the missing number instead of the zero by traversing the matrix horizontally to find the missing number. You should not repeat a number in the same row.

1	0	3
3	1	0
2	3	0

**Hint:**

- if number 1 is missing, the sum of the other two numbers is 5.
- if number 2 is missing, the sum of the other two numbers is 4.
- if number 3 is missing, the sum of the other two numbers is 3.

**Solution:**

```
#include <iostream>
using namespace std;

int main() {
    const int ROWS = 3;
    const int COLS = 3;
    int puzzle[ROWS][COLS] = { {1,0,3},{3,1,0},{2,3,0} };
    int sum[ROWS] = {};

    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLS; j++)
        {
            sum[i] += puzzle[i][j];
        }
    }

    //////////////////////////////////////
    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLS; j++)
        {
            if ((sum[i]==5) && (puzzle[i][j]==0))
            {
                puzzle[i][j] = 1;
                break;
            }
            if ((sum[i]==4) && (puzzle[i][j]== 0))
            {
                puzzle[i][j] = 2;
                break;
            }
            if ((sum[i] == 3) && (puzzle[i][j] == 0))
            {
                puzzle[i][j] = 3;
                break;
            }
        }
    }

    //////////////////////////////////////
    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLS; j++)
        {
            cout << puzzle[i][j] << " ";
        }
        cout << endl;
    }
}
```