Lab's Scope:

- **Function part 2 (Calling by Reference)**

Problem 1:

Write a program that has a function converting an amount in Egyptian pounds into its equivalent amount of dollars and euros. The function should be a void function with three parameters representing the amount in Egyptian pounds, dollars, and euros. The function should affect the values of parameters in dollars and euros so the new values can be printed from the main application.

Hint: Assume 1 dollar = 47.5 pounds, and 1 euro = 52.5 pounds.

Solution:

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

void convert_currency(float pounds, float& dollars, float& euros)
{
    dollars = pounds / 47.5;
    euros = pounds / 52.5;
}
////////////////////////////////////////////////////////////////
int main() {
    float pounds, dollars, euros;
    cout << "Enter the amount in Egyptian pounds:";
    cin >> pounds;
    convert_currency(pounds, dollars, euros);
    cout << "The amount in dollars is: " << dollars << endl;
    cout << "The amount in euros is: " << euros << endl;
    return 0;
}
```

Problem 2:

Write a program that has a function called "count_punctation" that given a string, the function should count the number of commas "," and periods "." in the string. The main applications should be able to print the number of commas and periods calculated by the function.

Solution:

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

void count_punctuation(string s, int& periods, int& commas)
{
    for (int i = 0; i < s.length(); i++)
    {
        if (s[i] == '.')
            periods++;
        if (s[i] == ',')
            commas++;
    }
}
/////////////////////////////////////////////////////////////////
int main() {
    string sentence;
    int periods = 0, commas = 0;
    cout << "Enter a sentence:";
    getline(cin, sentence);
    count_punctuation(sentence, periods, commas);
    cout << "The number of periods is: " << periods << endl;
    cout << "the number of commas is: " << commas << endl;
    return 0;
}
```

Problem 3 :

Write a program that has two functions. The first function returns the minimum number of an array of integers. The second function passes by the array elements and decrements each element with the minimum number returned by the first function. Add a third function to print the array elements.

Example:

50	60	10	20	5	40	30

Minimum number is 5, then the array should change to:

45	55	5	15	0	35	25

Solution:

```
#include <iostream>
using namespace std;
const int SIZE = 7;
int getMin(int arr[])
{
    int min = arr[0];
    for (int i = 0; i < SIZE; i++)
    {
        if (arr[i] < min)
            min = arr[i];
    }
    return min;
}
////////////////////////////////////////////
void decrement(int arr[], int min)
{
    for (int i = 0; i < SIZE; i++)
    {
        arr[i] -= min;
    }
}
////////////////////////////////////////////
void printArray(int arr[])
{
    for (int i = 0; i < SIZE; i++)
    {
        cout << arr[i] << " ";
    }
}
////////////////////////////////////////////////////////////////
int main() {
    int arr[SIZE] = { 50,60,10,20,5,40,30 };
    int min = getMin(arr);
    decrement(arr, min);
    printArray(arr);
    return 0;
}
```

Problem 4:

Write a program that has a function called “convertToEven” that given an array of integers, checks all elements in the array. If any element is odd convert it to even by adding 1 to the array element. Add a second function to print the array elements.

Solution:

```
#include <iostream>
using namespace std;
const int SIZE = 5;

void convertToEven(int arr[])
{
    for (int i = 0; i < SIZE; i++)
    {
        if (arr[i] % 2 == 1)
            arr[i] += 1;
    }
}
////////////////////////////////////////////
void printArray(int arr[])
{
    for (int i = 0; i < SIZE; i++)
    {
        cout << arr[i] << " ";
    }
}
////////////////////////////////////////////////////////////////
int main() {
    int arr[SIZE] = { 53,62,11,15,6 };
    convertToEven(arr);
    printArray(arr);
    return 0;
}
```

Problem 5:

Write a program that has a function called “convertToLower” that given an array of characters, checks all elements in the array. If any element is an upper-case letter, convert it to lower-case. Add a second function to print the array elements.

Hint: you could use built-in functions isupper() and tolower().

Solution:

```
#include <iostream>
using namespace std;
const int SIZE = 5;

void convertToLower(char arr[])
{
    for (int i = 0; i < SIZE; i++)
    {
        if (isupper(arr[i]))
            arr[i] = tolower(arr[i]);
    }
}
////////////////////////////////////////////
void printArray(char arr[])
{
    for (int i = 0; i < SIZE; i++)
    {
        cout << arr[i] << " ";
    }
}
////////////////////////////////////////////////////////////////
int main() {
    char arr[SIZE] = { 'H','E','l','L','o' };
    convertToLower(arr);
    printArray(arr);
    return 0;
}
```

Problem 6:

Write a program that has a function called “swapElements” given an array of integers, the function should swap the elements between the beginning and the end of the array elements till it reaches the middle of the array. Add a second function to print the array elements.

Example:

If the input array is:

10	20	30	40	50

The array should change to:

50	40	30	20	10

Solution:

```
#include <iostream>
using namespace std;
const int SIZE = 5;

void swapElements(int arr[])
{
    int temp;
    for (int i = 0; i < SIZE / 2; i++)
    {
        temp = arr[i];
        arr[i] = arr[SIZE - 1 - i];
        arr[SIZE - 1 - i] = temp;
    }
}
////////////////////////////////////////////
void printArray(int arr[])
{
    for (int i = 0; i < SIZE; i++)
    {
        cout << arr[i] << " ";
    }
}
////////////////////////////////////////////////////////////////
int main() {
    int arr[SIZE] = { 10,20,30,40,50 };
    swapElements(arr);
    printArray(arr);
    return 0;
}
```

Problem 7:

Write a program that has a function called “isIdentical” that given two matrices, the function should return true if the two matrices are identical, or false otherwise. Two matrices are identical if they have the same elements at the same positions.

Solution:

```
#include <iostream>
using namespace std;
const int ROWS = 3;
const int COLS = 4;

bool isIdentical(int a[][COLS], int b[][COLS])
{
    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLS; j++)
        {
            if (a[i][j] != b[i][j])
                return false;
        }
    }
    return true;
}
/////////////////////////////////////////////////////////////////////////////////
////////////////////
int main() {
    int a[ROWS][COLS] = { {10,20,30,40},{50,60,70,80},{100,200,300,400} };
    int b[ROWS][COLS] = { {10,20,30,40},{50,60,70,80},{100,200,300,400} };
    bool check = isIdentical(a, b);
    if (check)
        cout << "The two matrics are identical." << endl;
    else
        cout << "The two matrics are not identical." << endl;
    return 0;
}
```

Problem 8:

Write a program that has a function called “calculateAverage” that given a matrix where the last column contains the value 0, the function should calculate the average for each row, and insert this average in the last column of the row. Add a second function to print the matrix.

Example:

If the input matrix is:

10	20	30	40	0
11	25	30	80	0
35	66	88	97	0

Then the output matrix is:

10	20	30	40	25
11	25	30	80	36.5
35	66	88	97	71.5

Solution:

```
#include <iostream>
using namespace std;
const int ROWS = 3;
const int COLS = 5;

void calculateAverage(float a[][COLS])
{
    float sum, avg;
    for (int i = 0; i < ROWS; i++)
    {
        sum = 0;
        for (int j = 0; j < COLS - 1; j++)
        {
            sum += a[i][j];
        }
        avg = sum / (COLS - 1);
        a[i][COLS - 1] = avg;
    }
}
////////////////////////////////////////////
void printArray(float a[][COLS])
{
    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLS; j++)
        {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
}
////////////////////////////////////////////////////////////////
int main() {
    float a[ROWS][COLS] = { {10,20,30,40,0},{11,25,30,80,0},{35,66,88,97,0} };
    calculateAverage(a);
    printArray(a);
    return 0;
}
```