LAB TASK # 03

CODE # 01:

```
#include<iostream>
using namespace std;
int main()
{
  int sum=0, mul=1,size,n;
  cout<<"Enter size of 2D Array: ";
  cin>>size;
  int arr[size][size];
  cout<<"Enter values in array: "<<endl;
  for(int i=0; i<size; i++)
  {
    for(int j=0; j<size; j++)
    {
      cin>>arr[i][j];
    }
  }

  for(int i=0; i<size; i++)
  {
    for(int j=0; j<size; j++)
    {
      sum=sum+arr[i][j];
    }
  }

  for(int i=0; i<size; i++)
  {
    for(int j=0; j<size; j++)
    {
      mul=mul*arr[i][j];
    }
  }
  cout<<"Sum of array: "<<sum<<"\n";
  cout<<"Mul of array: "<<mul<<"\n";
  n=size*size;
```

```
     double avg=sum/n;
     cout<<"Avg of array: "<<avg;
     return 0;
   }
```

CODE # 02:

```
#include<iostream>
using namespace std;
int main()
{
  int a,b;

  cout<<"Enter value of 1st var: "<<endl;
  cin>>a;

  cout<<"Enter value of 2nd var: "<<endl;
  cin>>b;

  cout<<"\n";

  cout<<"Before Swap"<<endl;
  cout<<"A: "<<a<<"\t B: "<<b<<endl;

  int *PTR1=&a;
  int *PTR2=&b;

  int temp=*PTR1;
  *PTR1=*PTR2;
  *PTR2=temp;

  cout<<"After Swap"<<endl;
  cout<<"A: "<<a<<"\t B: "<<b;

  return 0;
}
```

CODE # 03:

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

```
int main()
{
    int max=0;
   int arr[10];

   cout<<"Enter values in array: "<<endl;

   for(int i=0; i<10; i++)
   {
      cout<<i+1<<".\t";
      cin>>arr[i];
   }

   for(int i=0; i<10; i++)
   {
      if(max<arr[i])
      {
          max=arr[i];
      }
   }

   int min=max;

   for(int i=0; i<10; i++)
   {
      if(min>arr[i])
      {
          min=arr[i];
      }
   }
   cout<<"\n";
   cout<<"Largest value in the array is:\t"<<max<<endl;
   cout<<"\n";
   cout<<"Smallest value in the array is:\t"<<min;
   cout<<"\n";

   return 0;
}
```

CODE # 04:

```
#include<iostream>
using namespace std;
int main()
{
   const int month=12;
   double arr[month], TotalRain=0, AvgRain, maxRain, minRain;
   int maxMonth=1, minMonth=1;

   for(int i=0; i<month; i++)
   {
       cout<<"Enter total rainfall for month "<<i+1<<": ";
       cin>>arr[i];
   }

   maxRain=minRain=arr[0];

   for(int i=0; i<month; i++)
   {
       TotalRain+=arr[i];

       if(maxRain<arr[i])
      {
          maxRain=arr[i];
          maxMonth=i+1;
      }

      if(minRain>arr[i])
      {
          minRain=arr[i];
          minMonth=i+1;
      }
   }

   AvgRain=TotalRain/month;

   cout<<endl;
   cout<<"Total RainFall Over A Year: "<<TotalRain<<" units"<<endl;
   cout<<"Average Monthly RainFall: "<<AvgRain<<" units"<<endl;
   cout<<endl;
   cout<<"Month With The Highest RainFall: "<<maxMonth<<" ("<<maxRain<<" units)"<<endl;
```

```
  cout<<"Month With The Lowest RainFall: "<<minMonth<<" ("<<minRain<<" units)"<<endl;

  return 0;
}
```

CODE # 05:

```
#include <iostream>

using namespace std;

int getTotal(int** array, int rows, int cols) {
    int total = 0;
    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            total += array[i][j];
        }
    }
    return total;
}

double getAverage(int** array, int rows, int cols) {
    int total = getTotal(array, rows, cols);
    return static_cast<double>(total) / (rows * cols);
}

int getRowTotal(int** array, int row, int cols) {
    int rowTotal = 0;
    for (int i = 0; i < cols; ++i) {
        rowTotal += array[row][i];
    }
    return rowTotal;
}

int getColumnTotal(int** array, int rows, int col) {
    int colTotal = 0;
    for (int i = 0; i < rows; ++i) {
        colTotal += array[i][col];
    }
    return colTotal;
}
```

```
int getHighestInRow(int** array, int row, int cols) {
    int highest = array[row][0];
    for (int i = 1; i < cols; ++i) {
        if (array[row][i] > highest) {
            highest = array[row][i];
        }
    }
    return highest;
}

int getHighestInColumn(int** array, int rows, int col) {
    int highest = array[0][col];
    for (int i = 1; i < rows; ++i) {
        if (array[i][col] > highest) {
            highest = array[i][col];
        }
    }
    return highest;
}

int getValidIndex(int limit, const string& indexType) {
    int index;
    while (true) {
        cout << "Enter the " << indexType << " index (0 to " << limit - 1 << "): ";
        cin >> index;
        if (index >= 0 && index < limit) {
            return index;
        } else {
            cout << "Invalid " << indexType << " index! Please try again." << endl;
        }
    }
}

int main() {
    int rows, cols;

    cout << "Enter the number of rows: ";
    cin >> rows;
    cout << "Enter the number of columns: ";
```

```
cin >> cols;

while (rows <= 0 || cols <= 0) {
    cout << "Invalid number of rows or columns! Please enter positive values." << endl;
    cout << "Enter the number of rows: ";
    cin >> rows;
    cout << "Enter the number of columns: ";
    cin >> cols;
}

int** array = new int*[rows];
for (int i = 0; i < rows; ++i) {
    array[i] = new int[cols];
}

cout << "Enter the elements of the array:" << endl;
for (int i = 0; i < rows; ++i) {
    for (int j = 0; j < cols; ++j) {
        cin >> array[i][j];
    }
}

cout<<endl;
cout << "Array (" << rows << " x " << cols << "):" << endl;
for (int i = 0; i < rows; ++i) {
    for (int j = 0; j < cols; ++j) {
        cout << array[i][j] << " ";
    }
    cout << endl;
}

cout<<endl;
cout << "Total of all elements: " << getTotal(array, rows, cols) << "\n" << endl;
cout << "Average of all elements: " << getAverage(array, rows, cols) << "\n" << endl;

cout << "Calculate total of a specific row" << endl;
int row = getValidIndex(rows, "row");
cout << "Total of row " << row << ": " << getRowTotal(array, row, cols) << endl;

cout<<endl;
```

```
    cout << "Calculate total of a specific column" << endl;
    int col = getValidIndex(cols, "column");
    cout << "Total of column " << col << ": " << getColumnTotal(array, rows, col) << endl;

    cout<<endl;
    cout << "Find highest value in a specific row" << endl;
    row = getValidIndex(rows, "row");
    cout << "Highest in row " << row << ": " << getHighestInRow(array, row, cols) << endl;

    cout<<endl;
    cout << "Find highest value in a specific column" << endl;
    col = getValidIndex(cols, "column");
    cout << "Highest in column " << col << ": " << getHighestInColumn(array, rows, col) << endl;

    for (int i = 0; i < rows; ++i) {
        delete[] array[i];
    }
    delete[] array;

    return 0;
}
```

CODE # 06:

```
#include <iostream>

using namespace std;

int main() {
    int size;

    cout << "Enter the number of integers: ";
    cin >> size;

    int* array = new int[size];

    cout << "Enter " << size << " integers:" << endl;
    for (int i = 0; i < size; ++i) {
        cin >> array[i];
    }
    int sumOdd = 0;
    for (int i = 0; i < size; ++i) {
        if (array[i] % 2 != 0) {
            sumOdd += array[i];
        }
    }

    cout << "Sum of odd integers: " << sumOdd << endl;

    delete[] array;

    return 0;
}
```

CODE # 07:

```
#include <iostream>

using namespace std;

int main() {
    int variable;
    cout<<"Enter Number: ";
    cin>>variable;
    int* pointer = &variable;
    cout<<endl;

    cout << "Value of variable: " << *pointer << endl;
    cout<<endl;
    cout << "Address of variable: " << pointer << endl;
    cout<<endl;
    cout << "Address of pointer: " << &pointer << endl;

    return 0;
}
```

CODE # 08:

```
#include <iostream>

using namespace std;
```

```
int main() {
    int a, b;
    int* ptrA = &a;
    int* ptrB = &b;

    cout << "Enter an integer value for a: ";
    cin >> a;
    cout << "Enter an integer value for b: ";
    cin >> b;

    cout << "Value of a (using pointer): " << *ptrA << endl;
    cout << "Value of b (using pointer): " << *ptrB << endl;

    cout << "Address of a: " << ptrA << endl;
    cout << "Address of b: " << ptrB << endl;

    return 0;
}
```

CODE # 09:

```
#include <iostream>
#include <cstdlib>

using namespace std;

void Addition(int a, int b);
void Subtraction(int a, int b);
void Division(int a, int b);
void Multiplication(int a, int b);
int Power(int base, int exponent);

int main() {
    int choice, a, b, number, exp;

    while (true) {
        cout << "\nCalculator Menu:" << endl;
        cout << "1. Addition" << endl;
        cout << "2. Subtraction" << endl;
        cout << "3. Multiplication" << endl;
        cout << "4. Division" << endl;
        cout << "5. Power" << endl;
        cout << "6. Exit" << endl;
        cout << "Enter your choice (1-6): ";
        cin >> choice;

        system("cls");

        if (choice == 6) {
            cout << "Exiting the calculator." << endl;
            break;
        }

        switch (choice) {
            case 1:
                cout << "Enter two integers: ";
                cin >> a >> b;
                Addition(a, b);
                break;
            case 2:
                cout << "Enter two integers: ";
                cin >> a >> b;
                Subtraction(a, b);
                break;
            case 3:
                cout << "Enter two integers: ";
                cin >> a >> b;
                Multiplication(a, b);
                break;
            case 4:
                cout << "Enter two integers: ";
                cin >> a >> b;
                Division(a, b);
                break;
            case 5:
                cout << "Enter the base and exponent: ";
                cin >> number >> exp;
                cout << "Result of Power: " << Power(number, exp) << endl;
                break;
            default:
                cout << "Invalid choice! Please try again." << endl;
        }

        system("pause");
        system("cls");
    }

    return 0;
}

void Addition(int a, int b) {
    cout << "Result of Addition: " << (a + b) << endl;
}

void Subtraction(int a, int b) {
    cout << "Result of Subtraction: " << (a - b) << endl;
}

void Division(int a, int b) {
    if (b == 0) {
        cout << "Division by zero is not allowed." << endl;
    } else {
        cout << "Result of Division: " << (a / b) << endl;
    }
}


void Multiplication(int a, int b) {
    cout << "Result of Multiplication: " << (a * b) << endl;
}

int Power(int base, int exponent) {
    int result = 1;
    for (int i = 0; i < exponent; ++i) {
        result *= base;
    }
    return result;
}
```