main.cpp

Run

```cpp
//33.Print Pascal Triangle Pattern using Nested For Loop

#include <iostream>
using namespace std;

int main() {
    int rows;
    cout << "Enter the number of rows for Pascal Triangle: ";
    cin >> rows;

    for (int i = 0; i < rows; i++) {
        int coef = 1;
        for (int space = 1; space <= rows - i; space++)
            cout << "  ";

        for (int j = 0; j <= i; j++) {
            cout << coef << "   ";
            coef = coef * (i - j) / (j + 1);
        }
        cout << endl;
    }

    return 0;
}

```

Output

Clear

```
/tmp/TCIalqUpeV.o
Enter the number of rows for Pascal Triangle: 5
        1
      1   1
    1   2   1
  1   3   3   1
1   4   6   4   1
```

main.cpp

Run

```cpp
//34. Print Diamond Pattern with * using Nested For Loop
#include <iostream>
using namespace std;
int main() {
    int n;
    cout << "Enter the number of rows for Diamond Pattern: ";
    cin >> n;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n - i; j++)
            cout << " ";
        for (int k = 1; k <= 2 * i - 1; k++)
            cout << "*";
        cout << endl;
    }
    for (int i = n - 1; i >= 1; i--) {
        for (int j = 1; j <= n - i; j++)
            cout << " ";
        for (int k = 1; k <= 2 * i - 1; k++)
            cout << "*";
        cout << endl;
    }
    return 0;
}

```

Output

Clear

```
/tmp/vh2t5PkYG6.o
Enter the number of rows for Diamond Pattern: 5
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```

main.cpp

Run

```cpp
//35. Program to Reverse the Elements in an Array

#include <iostream>
using namespace std;

int main() {
    const int size = 5;
    int arr[size];

    cout << "Enter " << size << " elements in the array:" << endl;
    for (int i = 0; i < size; i++)
        cin >> arr[i];

    cout << "Reversed array: ";
    for (int i = size - 1; i >= 0; i--)
        cout << arr[i] << " ";

    return 0;
}

```

Output

Clear

```
/tmp/wUpq4ZKFYi.o
Enter 5 elements in the array:
3
4
5
6
7
Reversed array: 7 6 5 4 3
```

main.cpp

Run

```cpp
//36. Program to Insert an Element in an Array at a Specific Position

#include <iostream>
using namespace std;
int main() {
    const int size = 5;
    int arr[size + 1]; // Increased size to accommodate the new element
    cout << "Enter " << size << " elements in the array:" << endl;
    for (int i = 0; i < size; i++)
        cin >> arr[i];
    int pos, value;
    cout << "Enter the position to insert the element: ";
    cin >> pos;
    cout << "Enter the value to insert: ";
    cin >> value;
    // Shift elements to create space for the new element
    for (int i = size; i > pos; i--)
        arr[i] = arr[i - 1];
    // Insert the new element
    arr[pos] = value;
    cout << "Array after insertion: ";
    for (int i = 0; i <= size; i++)
        cout << arr[i] << " ";
    return 0;
}

```

Output

Clear

```
/tmp/b4VnxV0ZtH.o
Enter 5 elements in the array:
5
6
7
8
9
Enter the position to insert the element: 3
Enter the value to insert: 2
Array after insertion: 5 6 7 2 8 9
```

main.cpp

Run

```cpp
//37. Program to Delete an Element in an Array at a Specific Position

#include <iostream>
using namespace std;

int main() {
    const int size = 5;
    int arr[size];

    cout << "Enter " << size << " elements in the array:" << endl;
    for (int i = 0; i < size; i++)
        cin >> arr[i];

    int pos;
    cout << "Enter the position to delete the element: ";
    cin >> pos;

    for (int i = pos; i < size - 1; i++)
        arr[i] = arr[i + 1];

    cout << "Array after deletion: ";
    for (int i = 0; i < size - 1; i++)
        cout << arr[i] << " ";

    return 0;
}
```

Output

Clear

```
/tmp/gWFnNTe1Ww.o
Enter 5 elements in the array:
5
6
4
3
7
Enter the position to delete the element: 3
Array after deletion: 5 6 4 7
```

main.cpp

Run

```cpp
//38. Find the Sum of All Elements in an Array

#include <iostream>
using namespace std;

int main() {
    const int size = 5;
    int arr[size];
    cout << "Enter " << size << " elements in the array:" << endl;
    for (int i = 0; i < size; i++)
        cin >> arr[i];
    int sum = 0;
    for (int i = 0; i < size; i++)
        sum += arr[i];
    cout << "Sum of elements: " << sum;
    return 0;
}
```

Output

Clear

```
/tmp/gWFnNTe1Ww.o
Enter 5 elements in the array:
5
6
7
8
9
Sum of elements: 35
```

main.cpp

Run

Output

Clear

```cpp
//39. Find the Average of All Elements in an Array

#include <iostream>
using namespace std;

int main() {
    const int size = 5;
    int arr[size];

    cout << "Enter " << size << " elements in the array:" << endl;
    for (int i = 0; i < size; i++)
        cin >> arr[i];

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

    double average = static_cast<double>(sum) / size;

    cout << "Average of elements: " << average;

    return 0;
}
```

```
/tmp/gWFnNTe1Ww.o
Enter 5 elements in the array:
1
2
3
4
5
Average of elements:3.0
```

main.cpp

Run

```cpp
//40. Find the Second Largest Element in an Array
#include <iostream>
using namespace std;
int main() {
    const int size = 5;
    int arr[size];
    cout << "Enter " << size << " elements in the array:" << endl;
    for (int i = 0; i < size; i++)
        cin >> arr[i];
    int firstLargest, secondLargest;
    if (arr[0] > arr[1]) {
        firstLargest = arr[0];
        secondLargest = arr[1];
    } else {
        firstLargest = arr[1];
        secondLargest = arr[0];
    }
    for (int i = 2; i < size; i++) {
        if (arr[i] > firstLargest) {
            secondLargest = firstLargest;
            firstLargest = arr[i];
        } else if (arr[i] > secondLargest && arr[i] != firstLargest) {
            secondLargest = arr[i];
        }
    }
    cout << "Second Largest Element: " << secondLargest;
    return 0;
}
```

Output

Clear

```
/tmp/gWFnNTe1Ww.o
Enter 5 elements in the array:
5
6
7
8
9
Second Largest Element: 8
```

main.cpp

Run

```cpp
//41. Find the Number of Occurrences of a Value in an Array

#include <iostream>
using namespace std;

int main() {
    const int size = 5;
    int arr[size];

    cout << "Enter " << size << " elements in the array:" << endl;
    for (int i = 0; i < size; i++)
        cin >> arr[i];

    int value;
    cout << "Enter the value to find occurrences: ";
    cin >> value;

    int count = 0;
    for (int i = 0; i < size; i++) {
        if (arr[i] == value)
            count++;
    }

    cout << "Number of occurrences of " << value << ": " << count;

    return 0;
}
```

Output

Clear

```
/tmp/gWFnNTe1Ww.o
Enter 5 elements in the array:
5
6
7
8
9
Enter the value to find occurrences: 9
```

main.cpp

Run

```cpp
//42. Merge Two Arrays

#include <iostream>
using namespace std;

int main() {
    const int size1 = 3, size2 = 4;
    int arr1[size1] = {1, 2, 3};
    int arr2[size2] = {4, 5, 6, 7};

    int mergedSize = size1 + size2;
    int mergedArr[mergedSize];

    for (int i = 0; i < size1; i++)
        mergedArr[i] = arr1[i];

    for (int i = 0; i < size2; i++)
        mergedArr[size1 + i] = arr2[i];

    cout << "Merged Array: ";
    for (int i = 0; i < mergedSize; i++)
        cout << mergedArr[i] << " ";

    return 0;
}
```

Output

Clear

```
/tmp/gWFnNTe1Ww.o
Merged Array: 1 2 3 4 5 6 7
```

main.cpp

Run

```cpp
//43. Create a Dynamic Array Using Pointers and Display the Values

#include <iostream>
using namespace std;

int main() {
    int size;
    cout << "Enter the size of the dynamic array: ";
    cin >> size;

    int *dynamicArr = new int[size];

    cout << "Enter " << size << " elements in the array:" << endl;
    for (int i = 0; i < size; i++)
        cin >> dynamicArr[i];

    cout << "Dynamic Array: ";
    for (int i = 0; i < size; i++)
        cout << dynamicArr[i] << " ";

    delete[] dynamicArr;

    return 0;
}
```

Output

Clear

```
/tmp/gWFnNTe1Ww.o
Enter the size of the dynamic array: 5
Enter 5 elements in the array:
5
6
7
8
9
Dynamic Array: 5 6 7 8 9
```

main.cpp

Run

```cpp
//45. Add Two Matrices

#include <iostream>
using namespace std;

int main() {
    const int rows = 2, cols = 2;
    int matrix1[rows][cols] = {{1, 2},
                               {3, 4}};
    int matrix2[rows][cols] = {{5, 6},
                               {7, 8}};
    int sumMatrix[rows][cols];

    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

    cout << "Sum of Matrices:" << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++)
            cout << sumMatrix[i][j] << " ";
        cout << endl;
    }

    return 0;
}
```

Output

Clear

```
/tmp/gWFnNTe1Ww.o
Sum of Matrices:
6 8
10 12
```

main.cpp

Run

Output

Clear

```cpp
//46. Multiply Two Matrices

#include <iostream>
using namespace std;

int main() {
    const int rows1 = 2, cols1 = 2, rows2 = 2, cols2 = 2;
    int matrix1[rows1][cols1] = {{1, 2},
                                 {3, 4}};
    int matrix2[rows2][cols2] = {{5, 6},
                                 {7, 8}};
    int productMatrix[rows1][cols2];
    for (int i = 0; i < rows1; i++)
        for (int j = 0; j < cols2; j++)
            productMatrix[i][j] = 0;
    for (int i = 0; i < rows1; i++)
        for (int j = 0; j < cols2; j++)
            for (int k = 0; k < cols1; k++)
                productMatrix[i][j] += matrix1[i][k] * matrix2[k][j];
    cout << "Product of Matrices:" << endl;
    for (int i = 0; i < rows1; i++) {
        for (int j = 0; j < cols2; j++)
            cout << productMatrix[i][j] << " ";
        cout << endl;
    }
    return 0;
}
```

```
/tmp/gWFnNTe1Ww.o
Product of Matrices:
19 22
43 50
```

main.cpp

Run

Output

Clear

```cpp
//47. Find the Sum of Diagonals of a Matrix

#include <iostream>
using namespace std;

int main() {
    const int size = 3;
    int matrix[size][size] = {{1, 2, 3},
                              {4, 5, 6},
                              {7, 8, 9}};
    int sumDiagonals = 0;

    for (int i = 0; i < size; i++)
        sumDiagonals += matrix[i][i];

    for (int i = 0; i < size; i++)
        sumDiagonals += matrix[i][size - i - 1];

    cout << "Sum of Diagonals: " << sumDiagonals;

    return 0;
}
```

```
/tmp/gWFnNTe1Ww.o
Sum of Diagonals: 30
```