

main.cpp

Run

```
1 //33.Print Pascal Triangle Pattern using Nested For Loop
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     int rows;
8     cout << "Enter the number of rows for Pascal Triangle: ";
9     cin >> rows;
10
11     for (int i = 0; i < rows; i++) {
12         int coef = 1;
13         for (int space = 1; space <= rows - i; space++)
14             cout << " ";
15
16         for (int j = 0; j <= i; j++) {
17             cout << coef << " ";
18             coef = coef * (i - j) / (j + 1);
19         }
20         cout << endl;
21     }
22
23     return 0;
24 }
25
```

Output

Clear

```
/tmp/TClalqUpeV.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

```
1 //34. Print Diamond Pattern with * using Nested For Loop
2 #include <iostream>
3 using namespace std;
4 int main() {
5     int n;
6     cout << "Enter the number of rows for Diamond Pattern: ";
7     cin >> n;
8     for (int i = 1; i <= n; i++) {
9         for (int j = 1; j <= n - i; j++)
10             cout << " ";
11         for (int k = 1; k <= 2 * i - 1; k++)
12             cout << "**";
13         cout << endl;
14     }
15     for (int i = n - 1; i >= 1; i--) {
16         for (int j = 1; j <= n - i; j++)
17             cout << " ";
18         for (int k = 1; k <= 2 * i - 1; k++)
19             cout << "**";
20         cout << endl;
21     }
22     return 0;
23 }
24
```

Output

Clear

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

main.cpp

Run

```
1 //35. Program to Reverse the Elements in an Array
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     const int size = 5;
8     int arr[size];
9
10    cout << "Enter " << size << " elements in the array:" << endl;
11    for (int i = 0; i < size; i++)
12        cin >> arr[i];
13
14    cout << "Reversed array: ";
15    for (int i = size - 1; i >= 0; i--)
16        cout << arr[i] << " ";
17
18    return 0;
19 }
20
```

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

```
1 //36. Program to Insert an Element in an Array at a Specific Position
2
3 #include <iostream>
4 using namespace std;
5 int main() {
6     const int size = 5;
7     int arr[size + 1]; // Increased size to accommodate the new element
8     cout << "Enter " << size << " elements in the array:" << endl;
9     for (int i = 0; i < size; i++)
10         cin >> arr[i];
11     int pos, value;
12     cout << "Enter the position to insert the element: ";
13     cin >> pos;
14     cout << "Enter the value to insert: ";
15     cin >> value;
16     // Shift elements to create space for the new element
17     for (int i = size; i > pos; i--)
18         arr[i] = arr[i - 1];
19     // Insert the new element
20     arr[pos] = value;
21     cout << "Array after insertion: ";
22     for (int i = 0; i <= size; i++)
23         cout << arr[i] << " ";
24     return 0;
25 }
26
```

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

Output

Clear

```
//37. Program to Delete an Element in an Array at a Specific Position\n#include <iostream>\nusing namespace std;\n\nint main() {\n    const int size = 5;\n    int arr[size];\n\n    cout << "Enter " << size << " elements in the array:" << endl;\n    for (int i = 0; i < size; i++)\n        cin >> arr[i];\n\n    int pos;\n    cout << "Enter the position to delete the element: "; \n    cin >> pos;\n\n    for (int i = pos; i < size - 1; i++)\n        arr[i] = arr[i + 1];\n\n    cout << "Array after deletion: ";\n    for (int i = 0; i < size - 1; i++)\n        cout << arr[i] << " ";\n\n    return 0;\n}
```

```
/tmp/gWFnTe1Ww.o\nEnter 5 elements in the array:\n5\n6\n4\n3\n7\nEnter the position to delete the element: 3\nArray after deletion: 5 6 4 7 |
```

main.cpp

Run

Output

Clear

```
1 //38. Find the Sum of All Elements in an Array
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     const int size = 5;
8     int arr[size];
9     cout << "Enter " << size << " elements in the array:" << endl;
10    for (int i = 0; i < size; i++)
11        cin >> arr[i];
12    int sum = 0;
13    for (int i = 0; i < size; i++)
14        sum += arr[i];
15    cout << "Sum of elements: " << sum;
16    return 0;
17 }
```

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

main.cpp

Run

Output

Clear

```
1 //39. Find the Average of All Elements in an Array
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     const int size = 5;
8     int arr[size];
9
10    cout << "Enter " << size << " elements in the array:" << endl;
11    for (int i = 0; i < size; i++)
12        cin >> arr[i];
13
14    int sum = 0;
15    for (int i = 0; i < size; i++)
16        sum += arr[i];
17
18    double average = static_cast<double>(sum) / size;
19
20    cout << "Average of elements: " << average;
21
22    return 0;
23 }
```

```
/tmp/gWFnTe1Ww.o
Enter 5 elements in the array:
1
2
3
4
5
Average of elements:3.0
```

main.cpp

Run

```
1 //40. Find the Second Largest Element in an Array
2 #include <iostream>
3 using namespace std;
4 int main() {
5     const int size = 5;
6     int arr[size];
7     cout << "Enter " << size << " elements in the array:" << endl;
8     for (int i = 0; i < size; i++)
9         cin >> arr[i];
10    int firstLargest, secondLargest;
11    if (arr[0] > arr[1]) {
12        firstLargest = arr[0];
13        secondLargest = arr[1];
14    } else {
15        firstLargest = arr[1];
16        secondLargest = arr[0];
17    }
18    for (int i = 2; i < size; i++) {
19        if (arr[i] > firstLargest) {
20            secondLargest = firstLargest;
21            firstLargest = arr[i];
22        } else if (arr[i] > secondLargest && arr[i] != firstLargest) {
23            secondLargest = arr[i];
24        }
25    }
26    cout << "Second Largest Element: " << secondLargest;
27    return 0;
28 }
```

Output

Clear

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


main.cpp















Run

Output



Clear

```
1 //41. Find the Number of Occurrences of a Value in an Array
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     const int size = 5;
8     int arr[size];
9
10    cout << "Enter " << size << " elements in the array:" << endl;
11    for (int i = 0; i < size; i++)
12        cin >> arr[i];
13
14    int value;
15    cout << "Enter the value to find occurrences: ";
16    cin >> value;
17
18    int count = 0;
19    for (int i = 0; i < size; i++) {
20        if (arr[i] == value)
21            count++;
22    }
23
24    cout << "Number of occurrences of " << value << ": " << count;
25
26    return 0;
27 }
```

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



main.cpp



Run

```
1 //42. Merge Two Arrays
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     const int size1 = 3, size2 = 4;
8     int arr1[size1] = {1, 2, 3};
9     int arr2[size2] = {4, 5, 6, 7};
10
11     int mergedSize = size1 + size2;
12     int mergedArr[mergedSize];
13
14     for (int i = 0; i < size1; i++)
15         mergedArr[i] = arr1[i];
16
17     for (int i = 0; i < size2; i++)
18         mergedArr[size1 + i] = arr2[i];
19
20     cout << "Merged Array: ";
21     for (int i = 0; i < mergedSize; i++)
22         cout << mergedArr[i] << " ";
23
24     return 0;
25 }
```

Output

Clear

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

main.cpp

Run

Output

Clear

```
1 //43. Create a Dynamic Array Using Pointers and Display the Values
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     int size;
8     cout << "Enter the size of the dynamic array: ";
9     cin >> size;
10
11     int *dynamicArr = new int[size];
12
13     cout << "Enter " << size << " elements in the array:" << endl;
14     for (int i = 0; i < size; i++)
15         cin >> dynamicArr[i];
16
17     cout << "Dynamic Array: ";
18     for (int i = 0; i < size; i++)
19         cout << dynamicArr[i] << " ";
20
21     delete[] dynamicArr;
22
23     return 0;
24 }
```

```
/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 |
```

Waiting for pqhbl1.pubgalaxy.com...

The screenshot displays a C++ development environment with two main panels: a source code editor on the left and an output console on the right.

Source Code Editor:

- File Name:** main.cpp
- Code Content:**

```
1 //46. Multiply Two Matrices
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     const int rows1 = 2, cols1 = 2, rows2 = 2, cols2 = 2;
8     int matrix1[rows1][cols1] = {{1, 2},
9                                     {3, 4}};
10    int matrix2[rows2][cols2] = {{5, 6},
11                                  {7, 8}};
12    int productMatrix[rows1][cols2];
13    for (int i = 0; i < rows1; i++)
14        for (int j = 0; j < cols2; j++)
15            productMatrix[i][j] = 0;
16    for (int i = 0; i < rows1; i++)
17        for (int j = 0; j < cols2; j++)
18            for (int k = 0; k < cols1; k++)
19                productMatrix[i][j] += matrix1[i][k] * matrix2[k][j];
20    cout << "Product of Matrices:" << endl;
21    for (int i = 0; i < rows1; i++) {
22        for (int j = 0; j < cols2; j++)
23            cout << productMatrix[i][j] << " ";
24        cout << endl;
25    }
26    return 0;
27 }
```

Output Console:

- Title Bar:** Output
- Action Button:** Clear
- Content:**

```
/tmp/gWFnTe1Ww.o
Product of Matrices:
19 22
43 50
```

main.cpp

Run

Output

Clear

```
1 //47. Find the Sum of Diagonals of a Matrix
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     const int size = 3;
8     int matrix[size][size] = {{1, 2, 3},
9                               {4, 5, 6},
10                              {7, 8, 9}};
11     int sumDiagonals = 0;
12
13     for (int i = 0; i < size; i++)
14         sumDiagonals += matrix[i][i];
15
16     for (int i = 0; i < size; i++)
17         sumDiagonals += matrix[i][size - i - 1];
18
19     cout << "Sum of Diagonals: " << sumDiagonals;
20
21     return 0;
22 }
```

/tmp/gWFnNTe1Ww.o

Sum of Diagonals: 30