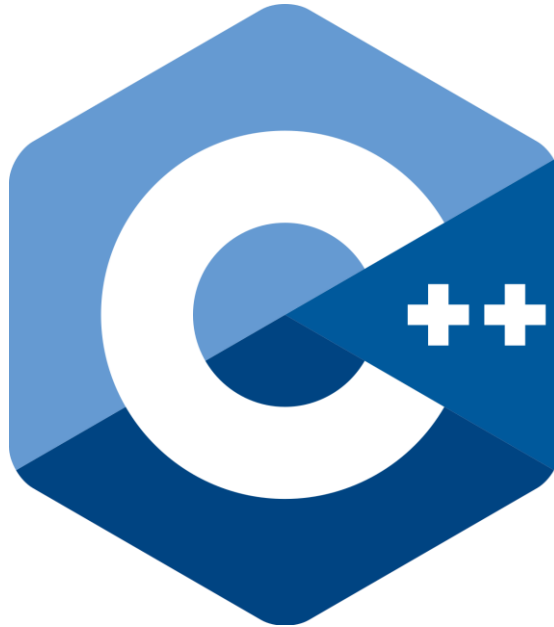
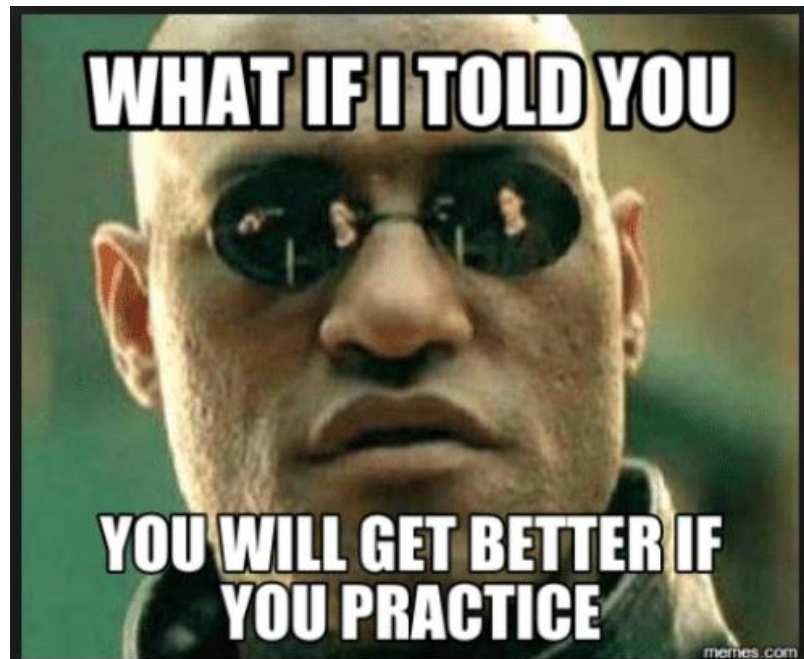




Fundamentals of Programming in C++





Make a Calculator

Take two numbers and

a mathematical operator as input.

print the output of the arithmetic operation.

```

#include <iostream>
using namespace std;

int main()
{
    float num1, num2;
    char operation;
    cout << "\t** Calculator **\n**To exit, enter the operation as '0'\n";
    bool loop = true;
    while (loop)
    {
        cout << "\nEnter 1st Number : ";
        cin >> num1;
        cout << "Enter the operation to perform (+ , - , / , * , %) : ";
        cin >> operation;
        cout << "Enter 2nd Number : ";
        cin >> num2;

        switch (operation)
        {
            case '+':
                cout << num1 << " + " << num2 << " = " << num1 + num2 << '\n';
                break;
            case '-':
                cout << num1 << " - " << num2 << " = " << num1 - num2 << '\n';
                break;
            case '/':
                cout << num1 << " / " << num2 << " = " << num1 / num2 << '\n';
                break;
            case '*':
                cout << num1 << " * " << num2 << " = " << num1 * num2 << '\n';
                break;
            case '%':
                cout << num1 << " % " << num2 << " = " << (int)num1 % (int)num2 << '\n';
                break;
            case '0':
                cout << "\t** Exiting the Calculator **\n";
                loop = false;
                break;

            default:
                cout << "Invalid Input, please try again.\n\n";
                break;
        }
    }
}

```

Given the size of an array and its elements,
write a function to get its maximum element.

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

int main()
{
    int n;
    cout << "Enter the size of array : ";
    cin >> n;
    int arr[n];
    printf("Enter the %02d array elements : ", n);
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }

    int max = -INT_MAX;
    for (int i = 0; i < n; i++)
    {
        if (arr[i] > max)
        {
            max = arr[i];
        }
    }
    cout << "The max element of the array is : " << max << '\n';
}
```

Using STL (Standard Template Library)

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;
    cout << "Enter the size of array : ";
    cin >> n;
    int arr[n];
    printf("Enter the %02d array elements : ", n);
    for (int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }

    sort(arr, arr + n);

    int max = arr[n - 1];

    cout << "The max element of the array is : " << max << '\n';
}
```

Matrix Addition

Take the dimensions $m \times n$ and the elements of two 2D arrays as input from the user and print the resultant array of the matrix addition.

```

#include <iostream>
using namespace std;

void matrixAddition(int rows, int cols)
{
    int matA[rows][cols];
    int matB[rows][cols];
    int matSum[rows][cols];

    cout << "Enter the elements of Matrix A : \n";
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            cin >> matA[i][j];

    cout << "Enter the elements of Matrix B : \n";
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            cin >> matB[i][j];

    // Doing the Sum
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            matSum[i][j] = matA[i][j] + matB[i][j];

    // Printing the Output
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
            printf("%0d ", matSum[i][j]);
        cout << '\n';
    }
}

int main()
{
    cout << "\t ** Matrix Addition **\n";
    cout << "Enter the dimensions m & n for the m*n matrix : ";
    int m, n;
    cin >> m >> n;
    matrixAddition(m, n);
}

```



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

int main()
{
    int rows;
    cin >> rows;
    for (int i = 1; i <= rows; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            cout << j << " ";
        }
        cout << '\n';
    }
}
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Inverted Half Pyramid

```
for (int i = rows; i ≥ 1; i--)  
{  
    for (int j = 1; j ≤ i; j++)  
    {  
        cout << "* ";  
    }  
    cout << '\n';  
}
```



```
* * * * *  
* * * *  
* * *  
* *  
*
```

Number Pyramid



Pascal's Triangle

```
      1
    2 3 2
  3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

```
      1
    1 1
  1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```