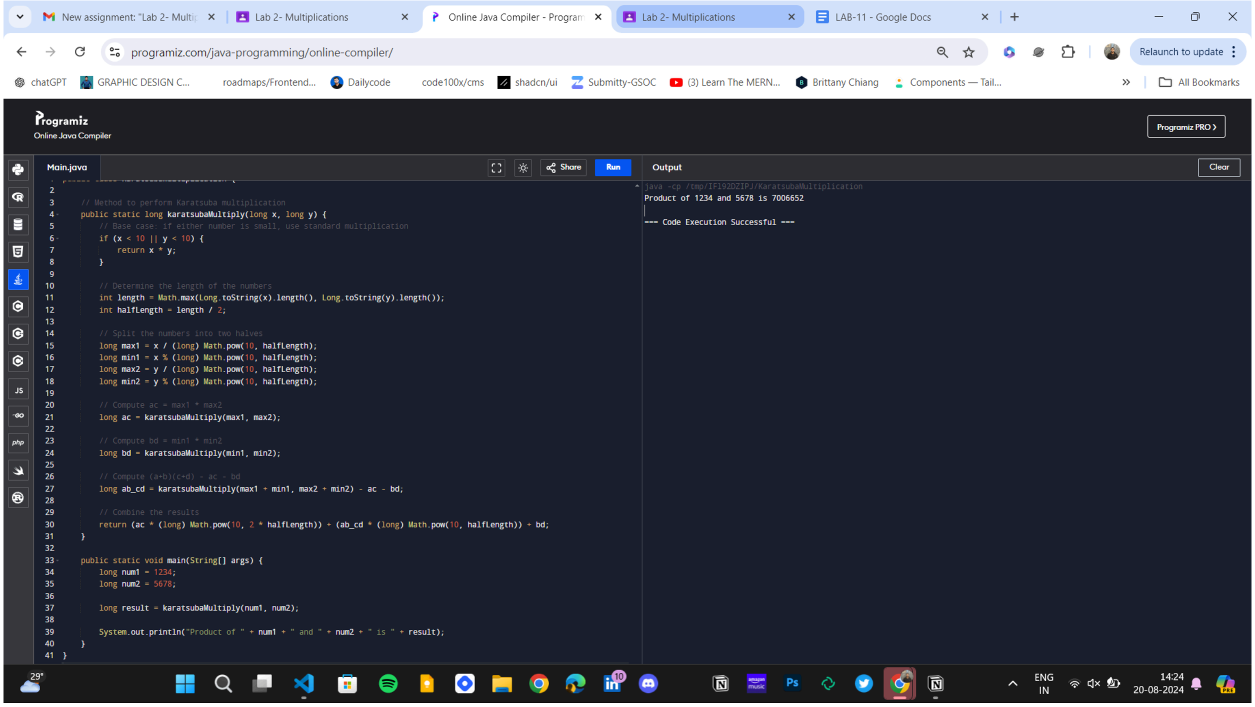
Advanced Algorithms

Assignment

NAME : G.Sai Bharath Chandra Reddy

Roll.No : 22cs2013

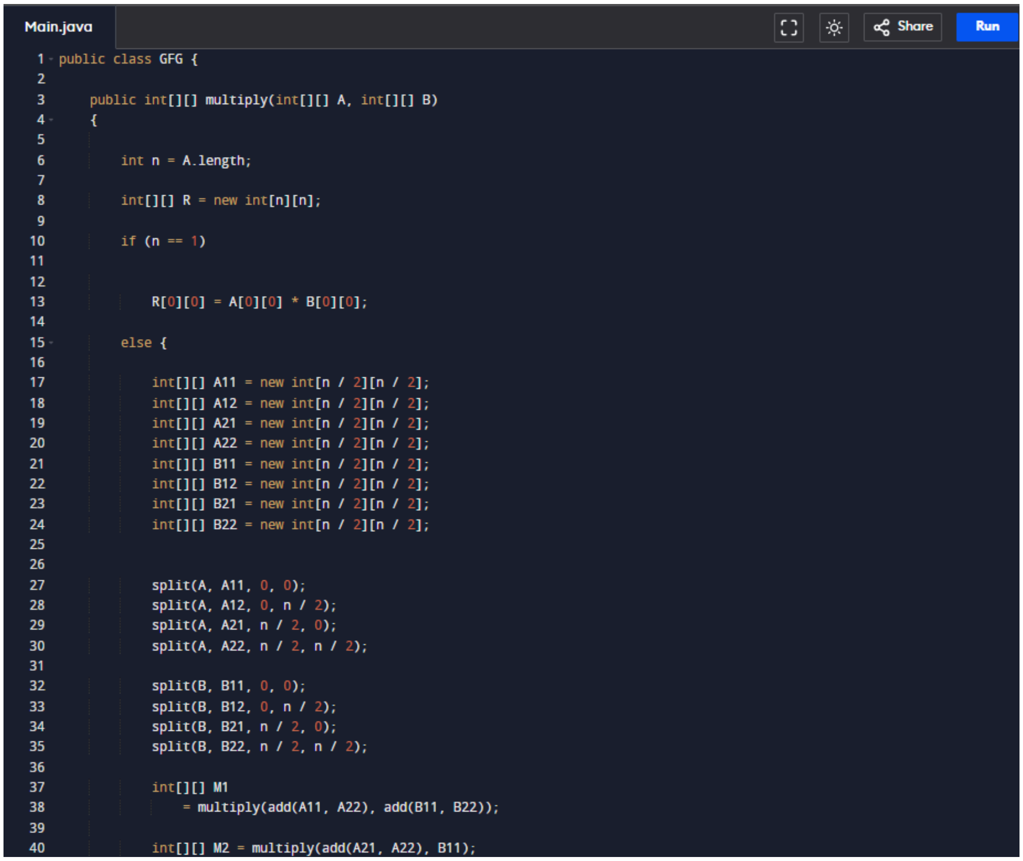
(A).Karatsuba Multiplication Algorithm.



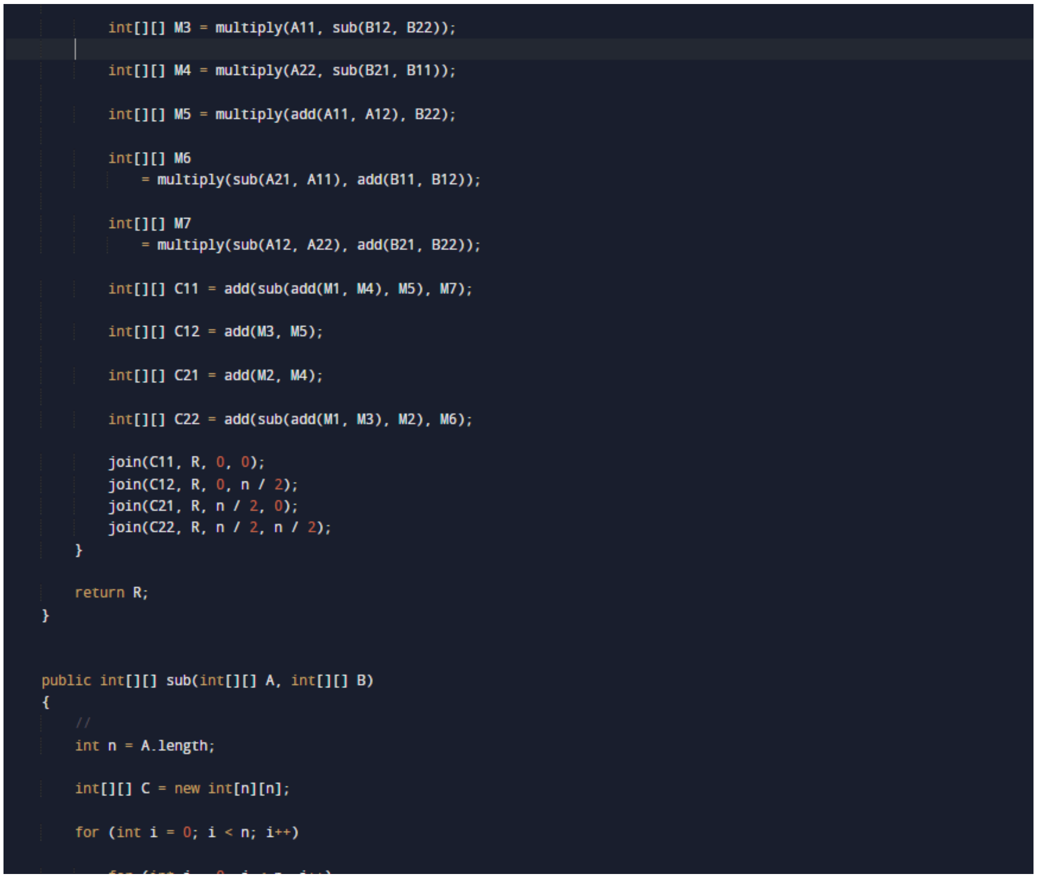
```java
public class KaratsubaMultiplication {

    // Method to perform Karatsuba multiplication
    public static long karatsubaMultiply(long x, long y) {
        // Base case: if either number is small, use standard multiplication
        if (x < 10 || y < 10) {
            return x * y;
        }

        // Determine the length of the numbers
        int length = Math.max(Long.toString(x).length(), Long.toString(y).length());
        int halfLength = length / 2;

        // Split the numbers into two halves
        long max1 = x / (long) Math.pow(10, halfLength);
        long min1 = x % (long) Math.pow(10, halfLength);
        long max2 = y / (long) Math.pow(10, halfLength);
        long min2 = y % (long) Math.pow(10, halfLength);

        // Compute ac = max1 * max2
        long ac = karatsubaMultiply(max1, max2);

        // Compute bd = min1 * min2
        long bd = karatsubaMultiply(min1, min2);

        // Compute (a+b)(c+d) - ac - bd
        long ab_cd = karatsubaMultiply(max1 + min1, max2 + min2) - ac - bd;

        // Combine the results
        return (ac * (long) Math.pow(10, 2 * halfLength)) + (ab_cd * (long) Math.pow(10, halfLength)) + bd;
    }

    public static void main(String[] args) {
        long num1 = 1234;
        long num2 = 5678;

        long result = karatsubaMultiply(num1, num2);

        System.out.println("Product of " + num1 + " and " + num2 + " is " + result);
    }
}
```

```
java -cp /tmp/IF192DZIPJ/KaratsubaMultiplication
Product of 1234 and 5678 is 7006652

=== Code Execution Successful ===
```

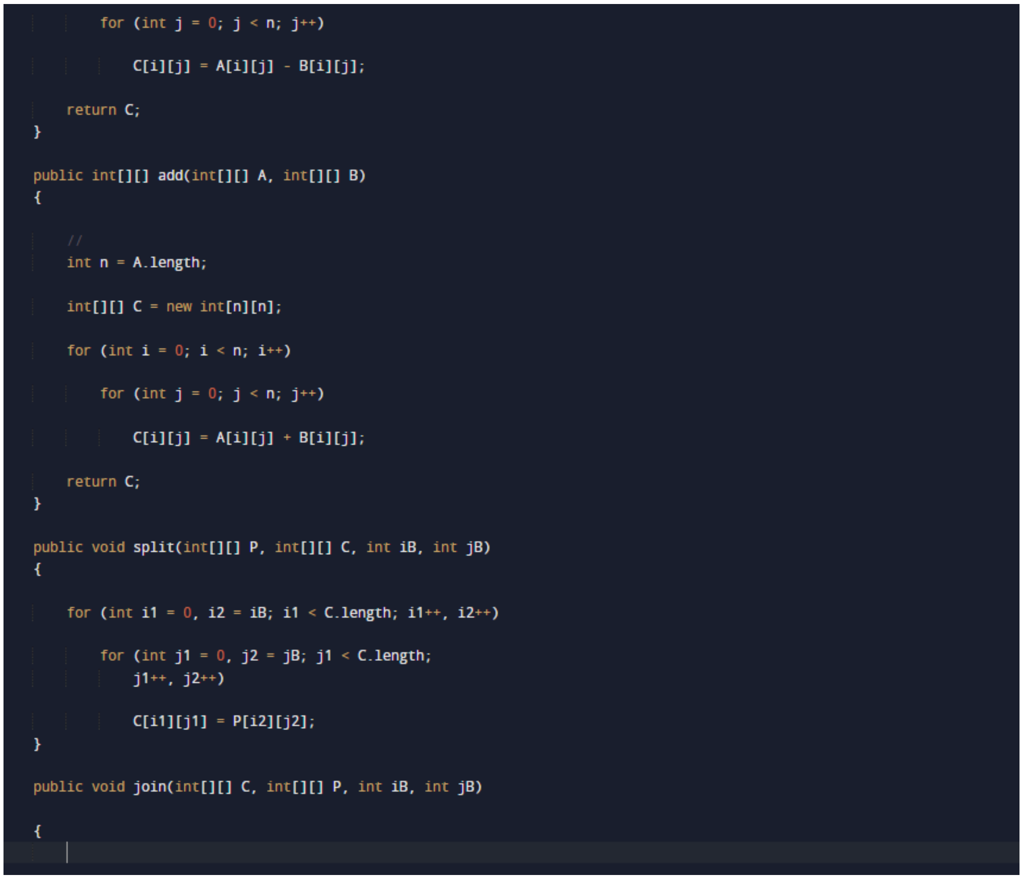
(B). Strassen Matrix Algorithm



```java
public class GFG {

    public int[][] multiply(int[][] A, int[][] B)
    {

        int n = A.length;

        int[][] R = new int[n][n];

        if (n == 1)


            R[0][0] = A[0][0] * B[0][0];

        else {

            int[][] A11 = new int[n / 2][n / 2];
            int[][] A12 = new int[n / 2][n / 2];
            int[][] A21 = new int[n / 2][n / 2];
            int[][] A22 = new int[n / 2][n / 2];
            int[][] B11 = new int[n / 2][n / 2];
            int[][] B12 = new int[n / 2][n / 2];
            int[][] B21 = new int[n / 2][n / 2];
            int[][] B22 = new int[n / 2][n / 2];


            split(A, A11, 0, 0);
            split(A, A12, 0, n / 2);
            split(A, A21, n / 2, 0);
            split(A, A22, n / 2, n / 2);

            split(B, B11, 0, 0);
            split(B, B12, 0, n / 2);
            split(B, B21, n / 2, 0);
            split(B, B22, n / 2, n / 2);

            int[][] M1
                = multiply(add(A11, A22), add(B11, B22));

            int[][] M2 = multiply(add(A21, A22), B11);
```

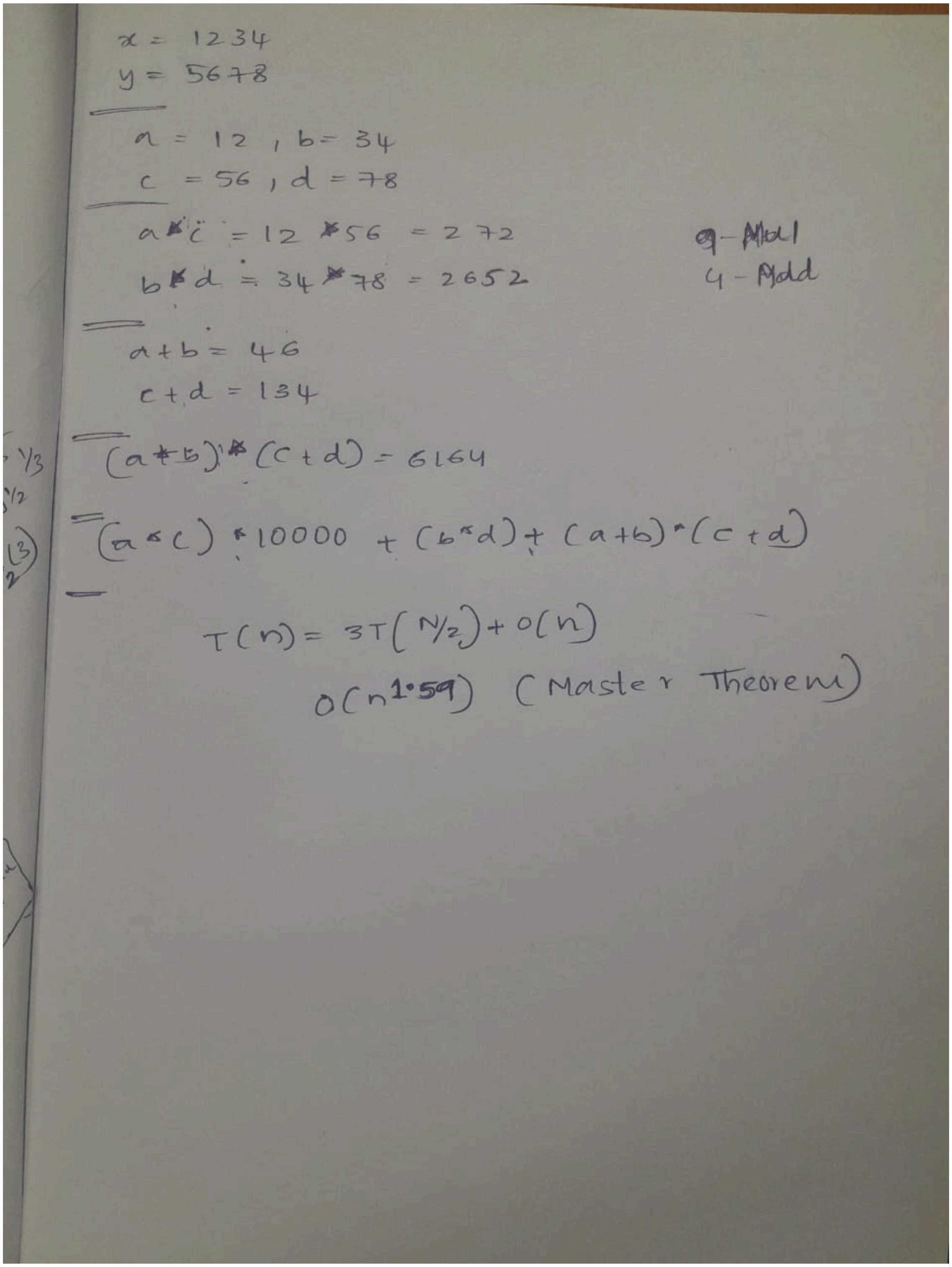


```
        int[][] M3 = multiply(A11, sub(B12, B22));

        int[][] M4 = multiply(A22, sub(B21, B11));

        int[][] M5 = multiply(add(A11, A12), B22);

        int[][] M6
            = multiply(sub(A21, A11), add(B11, B12));

        int[][] M7
            = multiply(sub(A12, A22), add(B21, B22));

        int[][] C11 = add(sub(add(M1, M4), M5), M7);

        int[][] C12 = add(M3, M5);

        int[][] C21 = add(M2, M4);

        int[][] C22 = add(sub(add(M1, M3), M2), M6);

        join(C11, R, 0, 0);
        join(C12, R, 0, n / 2);
        join(C21, R, n / 2, 0);
        join(C22, R, n / 2, n / 2);
    }

    return R;
}


public int[][] sub(int[][] A, int[][] B)
{
    //
    int n = A.length;

    int[][] C = new int[n][n];

    for (int i = 0; i < n; i++)
```



```
        for (int j = 0; j < n; j++)

            C[i][j] = A[i][j] - B[i][j];

    return C;
}

public int[][] add(int[][] A, int[][] B)
{

    //
    int n = A.length;

    int[][] C = new int[n][n];

    for (int i = 0; i < n; i++)

        for (int j = 0; j < n; j++)

            C[i][j] = A[i][j] + B[i][j];

    return C;
}

public void split(int[][] P, int[][] C, int iB, int jB)
{

    for (int i1 = 0, i2 = iB; i1 < C.length; i1++, i2++)

        for (int j1 = 0, j2 = jB; j1 < C.length;
            j1++, j2++)

            C[i1][j1] = P[i2][j2];
}

public void join(int[][] C, int[][] P, int iB, int jB)

{
```

(C)



$x = 1234$

$y = 5678$

$a = 12 , b = 34$

$c = 56 , d = 78$

$a * c = 12 * 56 = 272$

$b * d = 34 * 78 = 2652$

9 - Mul

4 - Add

$a + b = 46$

$c + d = 134$

$(a*b)*(c+d) = 6164$

$(a*c)*10000 + (b*d) + (a+b)*(c+d)$

$T(n) = 3T(N/2) + O(n)$

$O(n^{1.59})$ (Master Theorem)