

**Experiment No. 4**

**Title: Integer Multiplication by Divide and Conquer rule.**

**Description of the Problem Statement:**

Divide and conquer is used in this code to recursively compute the product of two integers. Three recursive calls are used to compute the product from the input numbers. If either number has a size of 1, the product is simply the product of both.



**Experiment and result:**

Code**:**

    #include <stdio.h>

    #include <math.h>

    long long multiply(long long x, long long y) {

    int sX = floor(log10(x)) + 1;

    int sY = floor(log10(y)) + 1;

    int n = fmax(sX, sY);

    if (n == 1) {

    return x \* y;

    }

    long long a = x / pow(10, n / 2);

    long long b = x % (long long)pow(10, n / 2);

    long long c = y / pow(10, n / 2);

    long long d = y % (long long)pow(10, n / 2);

    long long ac = multiply(a, c);

    long long bd = multiply(b, d);

    long long abcd = multiply(a + b, c + d);

    return ac \* pow(10, n) + (abcd - ac - bd) \* pow(10, n / 2) + bd;

    }

    int main() {

    long long a, b;

    printf("Enter two numbers to multiply: ");

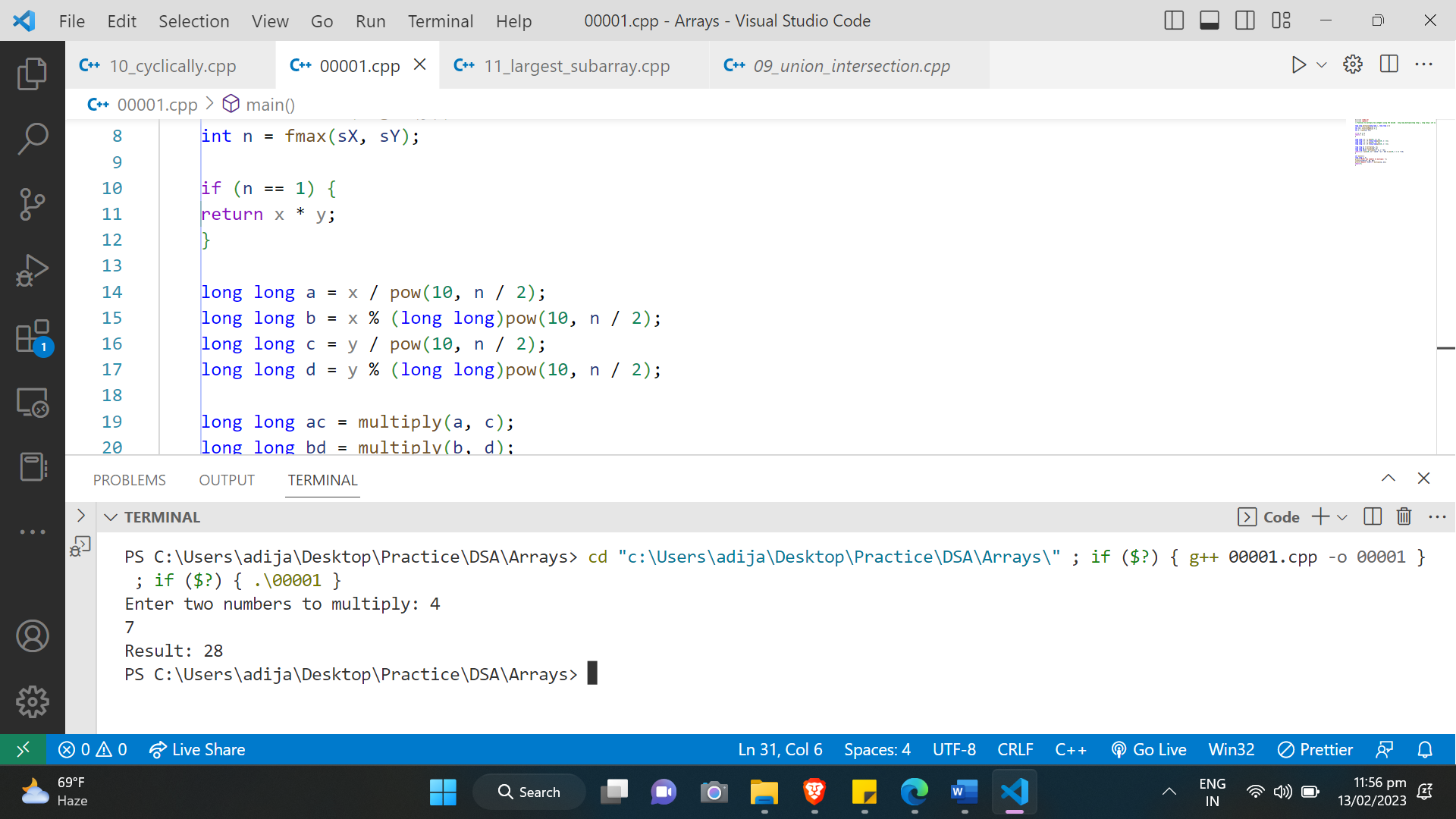
    scanf("%lld%lld", &a, &b);

    printf("Result: %lld\n", multiply(a, b));

    return 0;

    }

**Output:**



**Conclusions:**

**Divide and conquer was used successfully to multiply two integers in the above experiment.**