Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Design and Analysis of Algorithm	**Aim: : Implementing Karatsuba: Large Integer Multiplication Algorithm using Divide and Conquer Approach**	
Experiment No: 05	**Date: 13/09/2025**	**Enrollment No: 92301733049**

AIM: Implementing Karatsuba: Large Integer Multiplication Algorithm using Divide and Conquer Approach

I. Karatsuba Algorithm

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

// Make lengths equal by adding leading zeros
void makeEqualLength(string &s1, string &s2) {
    int len1 = s1.size();
    int len2 = s2.size();
    if (len1 < len2)
        s1.insert(0, len2 - len1, '0');
    else if (len2 < len1)
        s2.insert(0, len1 - len2, '0');
}

// Add two decimal strings
string addStrings(string s1, string s2) {
    makeEqualLength(s1, s2);
    int carry = 0;
    string result = "";

    for (int i = s1.size() - 1; i >= 0; i--) {
        int sum = (s1[i] - '0') + (s2[i] - '0') + carry;
        result.insert(result.begin(), (sum % 10) + '0');
        carry = sum / 10;
    }
    if (carry) result.insert(result.begin(), carry + '0');
    return result;
}

// Subtract two decimal strings (s1 >= s2)
string subtractStrings(string s1, string s2) {
    makeEqualLength(s1, s2);
    int borrow = 0;
    string result = "";
```

```
    for (int i = s1.size() - 1; i >= 0; i--) {
        int diff = (s1[i] - '0') - (s2[i] - '0') - borrow;
        if (diff < 0) {
            diff += 10;
            borrow = 1;
        } else borrow = 0;
        result.insert(result.begin(), diff + '0');
    }
    // Remove leading zeros
    int pos = result.find_first_not_of('0');
    if (pos != string::npos) return result.substr(pos);
    return "0";
}

// Convert single digit integer to string (no to_string needed)
string intToString(int num) {
    string res = "";
    if (num >= 10) {
        res.push_back((num / 10) + '0');
        res.push_back((num % 10) + '0');
    } else {
        res.push_back(num + '0');
    }
    return res;
}

// Karatsuba multiplication
string karatsuba(string x, string y) {
    makeEqualLength(x, y);
    int n = x.size();

    // Base case
    if (n == 0) return "0";
    if (n == 1) {
        int prod = (x[0] - '0') * (y[0] - '0');
        return intToString(prod);
    }

    int mid = n / 2;
```

```
    // Split numbers

    string xL = x.substr(0, mid);
    string xR = x.substr(mid);
    string yL = y.substr(0, mid);
    string yR = y.substr(mid);

    // Recursive calls
    string P1 = karatsuba(xL, yL);
    string P2 = karatsuba(xR, yR);
    string P3 = karatsuba(addStrings(xL, xR), addStrings(yL, yR));

    // Karatsuba formula: (P1 * 10^(2m)) + ((P3 - P1 - P2) * 10^m) + P2
    string temp1 = P1;
    temp1.append(2 * (n - mid), '0');  // multiply by 10^(2m)

    string temp2 = subtractStrings(subtractStrings(P3, P1), P2);
    temp2.append(n - mid, '0');        // multiply by 10^m

    string result = addStrings(addStrings(temp1, temp2), P2);

    // Remove leading zeros
    int pos = result.find_first_not_of('0');
    if (pos != string::npos) return result.substr(pos);
    return "0";
}

int main() {
    string num1 = "3141592653589793238462643383279502884197169399375105820974944592";
    string num2 = "2718281828459045235360287471352662497757247093699959574966967627";

    cout << "Number 1: " << num1 << endl;
    cout << "Number 2: " << num2 << endl;

    string result = karatsuba(num1, num2);
    cout << "Product (Karatsuba): " << result << endl;

    return 0;
}
```

Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Design and Analysis of Algorithm	**Aim: : Implementing Karatsuba: Large Integer Multiplication Algorithm using Divide and Conquer Approach**	
Experiment No: 05	**Date: 13/09/2025**	**Enrollment No: 92301733049**