Name	Tejas Jadhav
UID No.	2022301006
Class	COMPS A (B batch)
Experiment No.	4

Aim: Experiment on Dynamic Programming
- Take matrix count as input (up to 10)
- Generate random matrix whose order is between 15 to 46.
- Fill all matrix with random distribution of values 1 and 0.
- Determine the optimal parenthesizing of matrices.
- Perform matrix multiplication according to that parenthesizing.

Theory:

❖ **Dynamic Programming**

Dynamic Programming is a technique in computer programming that helps to efficiently solve a class of problems that have two properties:
1) Overlapping subproblems
2) Optimal substructure property

If any problem can be divided into subproblems, which in turn are divided into smaller subproblems, and if there are overlapping among these subproblems, then the solutions to these subproblems can be saved for future reference. In this way, efficiency of the CPU can be enhanced. This method of solving a solution is referred to as dynamic programming.

When developing a dynamic-programming algorithm, we follow a sequence of four steps:
1) Characterize the structure of an optimal solution.
2) Recursively define the value of an optimal solution.
3) Compute the value of an optimal solution (typically bottom-up fashion)
4) Construct an optimal solution from computed information.

Algorithm:

1) Matrix Chain Multiplication cost calulation

```
MCM(p)
  n = p.length – 1
  let m[1 .. n, 1 .. n] and s[1 .. n – 1, 2 .. n] be new tables
  for i = 1 to n
          m[i, i] = 0
  for l = 2 to n
          for i = 1 to n – l + 1
                    j = i + l – 1
                    m[i, j] = infinity
                    for k = i to j – 1
                              q = m[i, k] + m[k + 1, j] + p[i – 1] * p[k] * p[j]
                              if q < m[i, j]
                                        m[i, j] = q
                                        s[i, j] = k
  return m and s
```

2) Determining Optimal Parenthesization

```
OptimalParens(s, i, j)
  if  i == j
      return "M" + i
  else
      return "(" + OptimalParens(s, i, s[i][j]) + "*" + OptimalParens(s, s[i][j] + 1, j);
```

3) Convert Infix to Postfix

```
INFIX-TO-POSTFIX(E):
  S <- empty stack
  P <- empty list for postfix expression
  for each token t in E do:
     if t is an operand, append t to P
     if t is a left parenthesis, push t onto S
     if t is a right parenthesis, then:
        while the top of S is not a left parenthesis
                    pop operators from S and append them to P
        pop the left parenthesis from S and discard it
     if t is an operator, then:
        while there is an operator on top of S with greater or equal precedence
                     pop it from S and append it to P
        push t onto S
while S is not empty, pop operators from S and append them to P
return P
```

4) Postfix Evaluation

```
EVALUATE-POSTFIX(P):
   S <- empty stack
   for each token t in P do:
      if t is a number, push t onto S
      if t is the multiplication operator, then:
         pop the top two numbers a and b from S
         compute a * b
        push the result onto S
  the final result is the only element left on stack S
  return the result
```

5) Matrix Multiplication

MatrixMul(a, b, n)
    Let c be resultant matrix of size n x n
    For i = 1 to n
        for j = 1 to n
            c[i, j] = 0
            for k = 1 to n
                c[i, j] = c[i, j] + a[i, k] * b[k, j]
    return c

Code:

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

class Matrix {
   public:
    float** m;
    int row;
    int col;

    Matrix(int r, int c) {
        m = new float*[r];
        for (int i = 0; i < r; i++) {
            m[i] = new float[c];
        }

        for (int i = 0; i < r; i++) {
            for (int j = 0; j < c; j++) {
                m[i][j] = 0;
            }
        }

        row = r;
        col = c;
    }

    void fill_random_in_range(int min, int max) {
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < col; j++) {
                m[i][j] = rand() % (max - min + 1) + min;
            }
        }
    }
```

```
    static void print(Matrix* matrix, bool skip_zero = false, int w = 1) {
        int start = skip_zero ? 1 : 0;
        for (int i = start; i < matrix->row; i++) {
            for (int j = start; j < matrix->col; j++) {
                cout << left << setw(w) << matrix->m[i][j] << " ";
            }
            cout << endl;
        }
        cout << endl;
    }

    static long mul_count;
    static Matrix* multiply(Matrix* a, Matrix* b) {
        Matrix* c = new Matrix(a->row, b->col);

        for (int i = 0; i < a->row; i++) {
            for (int j = 0; j < b->col; j++) {
                int sum = 0;
                for (int k = 0; k < a->col; k++) {
                    sum += a->m[i][k] * b->m[k][j];
                    mul_count++;
                }
                c->m[i][j] = sum;
            }
        }
        return c;
    }
};

long Matrix::mul_count = 0;

void print_array(int* a, int n) {
    cout << "[ ";
    for (int i = 0; i < n; i++) {
        if (i == n - 1)
            cout << a[i];
        else
            cout << a[i] << ", ";
    }
    cout << "]" << endl;
}

int* gen_matrix_orders_in_range(int num, int min, int max) {
    int* p = new int[num + 1];
    srand(time(0));
    for (int i = 0; i <= num; i++) {
        p[i] = rand() % (max - min + 1) + min;
    }
    return p;
```

```
}

string optimal_parenthesization(Matrix* s, int i, int j) {
    if (i == j) {
        return "M" + to_string(i);

    } else {
        return "(" + optimal_parenthesization(s, i, s->m[i][j]) + "*" +
               optimal_parenthesization(s, s->m[i][j] + 1, j) + ")";
    }
}

string matrix_chain(int* p, int n, Matrix* m, Matrix* s) {
    int t = 1;
    for (int i = 1; i <= n - 1; i++) {
        for (int j = 1; j + t <= n; j++) {
            // j and (j + t) are indices of m
            // k = j to (j + t) - 1
            int min = INT_MAX;
            for (int k = j; k <= j + t - 1; k++) {
                int cost = m->m[j][k] + m->m[k + 1][j + t] +
                           p[j - 1] * p[k] * p[j + t];

                if (cost < min) {
                    min = cost;
                    m->m[j][j + t] = min;
                    s->m[j][j + t] = k;
                }
            }
        }
        t++;
    }

    return optimal_parenthesization(s, 1, n);
}

string to_postfix(string infix) {
    string postfix = "";
    vector<char> stack;

    for (int i = 0; i < infix.size(); i++) {
        char ch = infix[i];
        if (ch == '(') {
            stack.push_back(ch);
        } else if (ch == '*') {
            stack.push_back('*');
        } else if (ch == ')') {
            while (stack[stack.size() - 1] != '(') {
                postfix = postfix + stack.back();
                stack.pop_back();
```

```
            }
            stack.pop_back();
        } else {
            if (ch == 'M') {
                postfix += " ";
            }
            postfix = postfix + ch;
        }
    }

    while (stack.size() != 0) {
        char pop = stack.back();
        postfix = postfix + pop;
        stack.pop_back();
    }

    return postfix;
}

long count_normal = 0;

Matrix* eval_matrix_normal_mul(string postfix, Matrix** m_arr) {
    vector<Matrix*> eval;

    Matrix::mul_count = 0;

    for (int i = 0; i < postfix.size(); i++) {
        char ch = postfix[i];

        if (ch == 'M' || ch == ' ') {
            continue;
        }
        if (ch == '*') {
            Matrix* b = eval.back();
            eval.pop_back();

            Matrix* a = eval.back();
            eval.pop_back();

            Matrix* c = Matrix::multiply(a, b);

            eval.push_back(c);

        } else if (ch >= '1' || ch <= '9') {
            int index = ch - '0';
            if (ch == '1' && postfix[i + 1] == '0') {
                index = 10;
                i++;
            }
            eval.push_back(m_arr[index]);
```

```
        }
    }
    return eval.back();
}

int main() {
    int MATRIX_COUNT = 0;

    cout << "Enter number of matrices (<= 10) : ";
    cin >> MATRIX_COUNT;

    int* p = gen_matrix_orders_in_range(MATRIX_COUNT, 15, 46);

    cout << "\np[i] = ";
    print_array(p, MATRIX_COUNT + 1);
    cout << endl;

    Matrix** M = new Matrix*[MATRIX_COUNT + 1];

    for (int i = 1; i <= MATRIX_COUNT; i++) {
        M[i] = new Matrix(p[i - 1], p[i]);
        M[i]->fill_random_in_range(0, 1);
        cout << "Order of M" << i << " is (" << M[i]->row << ", " << M[i]-
>col
             << ")" << endl;
        Matrix::print(M[i]);
    }

    Matrix* m = new Matrix(MATRIX_COUNT + 1, MATRIX_COUNT + 1);
    Matrix* s = new Matrix(MATRIX_COUNT + 1, MATRIX_COUNT + 1);

    string optimum_inorder = matrix_chain(p, MATRIX_COUNT, m, s);

    cout << "\nCost matrix" << endl;
    Matrix::print(m, true, 8);

    cout << "\nParenthesization Matrix" << endl;
    Matrix::print(s, true);

    cout << "Optimal parenthesization : " << optimum_inorder << endl;

    string optimum_postfix = to_postfix(optimum_inorder);
    cout << "Postfix expression : " << optimum_postfix << endl;

    cout << "\nResult Of Multiplication : " << endl;

    Matrix::print(eval_matrix_normal_mul(optimum_postfix, M));

    cout << "Estimated Multiplication count: " << m->m[1][m->col - 1] <<
endl;
```

```
    cout << "Actual Multiplication count: " << Matrix::mul_count << endl;

    return 0;
}
```

Output:

1) Number of Matrices = 5

```
PS D:\Tejas\clg\daa\Experiment 04\code> g++ .\chain_mul.cpp
PS D:\Tejas\clg\daa\Experiment 04\code> ./a
Enter number of matrices (<= 10) : 5

p[i] = [ 31, 44, 19, 41, 24, 34]

Order of M1 is (31, 44)
1 1 0 1 1 0 0 1 1 1 0 1 1 0 1 1 1 0 1 0 1 0 0 1 1 0 1 1 0 0 0 1 0 0 0 1 0 1 1 1 0 1 0 0
0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 0 1 0 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 1 0 0 0 1 1 1
0 1 0 1 0 1 0 1 1 1 0 0 1 1 1 0 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0 1 1 1 1 1
0 0 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 0 1 1 0 1 0 1 1 0 1 0 0
0 0 1 1 0 0 1 1 1 0 1 0 1 0 0 1 1 1 0 1 1 0 0 0 1 1 0 1 0 0 1 0 0 1 0 1 1 1 0 0 1 0 1 0
0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 1 0 1 1 0 1 0 1 1 1 0 0 1 0 0 0 1 0 1 0
0 1 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0 1 1 1 0 0 1 1
1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 0 1 0 0 1 0 1
0 0 1 1 0 0 0 1 1 0 1 1 0 0 0 0 0 1 1 0 1 1 0 0 1 1 1 1 1 1 0 0 1 0 1 1 0 0 0 1 1 0 1 1
1 1 1 1 1 0 0 0 1 0 0 0 0 1 1 1 1 1 0 0 1 1 1 1 0 1 0 1 0 1 1 1 0 0 0 1 1 1 1 1 1 0 0 1
0 1 0 0 1 0 0 1 1 0 1 1 1 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 1 0 1 1 0 1 0 1 1 0 0 0 0 0 0 0
1 1 0 1 0 0 0 0 1 0 1 1 1 1 0 1 1 1 1 0 0 0 0 0 1 1 0 1 0 1 1 0 0 1 0 1 0 0 1 1 1 1 0 0
1 0 0 0 0 0 1 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 0 0 1 1 1 1 1 0 0 1 1 1 0 1 0 1 0 0 1 0 1 0
0 1 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 1 0 1 1 1 0 1 0 0 1 0 1 1 0 1
1 1 1 0 1 1 1 1 0 0 0 1 1 0 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 0 1 1 1 0
1 0 0 1 0 1 1 1 1 1 1 0 1 0 0 0 0 0 0 1 1 1 0 1 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 1 0 1 1 1
0 0 0 0 0 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 1 0 1
0 0 0 0 1 1 1 0 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 0 1 1 1 0 1 0 0 0 0 0 1 1 0 1 0 1 1 0 1 1
1 0 1 0 1 0 0 1 1 1 0 1 0 1 1 1 0 1 0 1 1 1 0 0 0 1 0 1 0 1 0 1 1 1 1 0 1 1 1 0 1 1 0 0
0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 1 1 0 1 0 1 1 0 0 0 0 0 0 1 1 1 1 0 1
0 0 1 1 1 0 1 0 0 0 0 0 0 0 1 1 1 1 1 0 1 1 0 1 0 1 0 1 1 1 1 1 1 0 1 0 0 0 1 0 0 1 1 1
1 0 0 1 0 1 0 0 0 0 1 1 0 1 1 1 1 1 0 1 1 0 1 0 0 1 1 1 1 0 1 1 1 1 0 1 0 1 1 0 0 0 0 0
0 1 1 1 0 0 1 0 0 1 1 1 0 0 1 1 0 1 1 0 0 0 0 1 1 0 1 0 1 1 1 0 0 0 1 0 0 1 1 0 1 0 0 0
0 0 1 0 1 1 1 1 1 0 1 0 0 0 1 1 1 1 0 1 0 0 1 0 1 1 1 0 0 0 1 1 0 0 1 1 1 0 1 0 1 1 1 0
1 0 1 1 1 1 0 1 1 0 1 1 0 1 0 0 1 0 0 0 1 1 1 1 0 1 0 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 1
1 0 0 0 1 1 0 1 1 1 0 0 0 1 1 0 0 1 1 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0 1 1 1 1
0 0 0 1 0 0 1 0 0 0 1 1 0 1 1 0 1 1 1 0 0 0 0 0 0 1 0 0 1 1 1 0 0 1 1 1 0 1 1 1 0 0 1 0
0 0 1 0 0 1 1 0 1 0 0 0 0 1 1 0 1 1 0 1 1 0 1 0 0 1 0 1 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 1
1 0 0 0 0 1 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 1 1 1 0 1 1 0 1 1 0 1 0 1 0 1 0 0
1 1 1 1 1 0 1 1 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 0 1 0 1 1 1 0 1
1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0 1 1 1 0 0 1 0 1 0 1 1 0 0
```

```
Order of M2 is (44, 19)
0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 1 0 1 1
1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1
1 0 0 1 0 1 1 0 0 1 0 1 0 1 0 1 0 0 1
1 0 1 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 0
0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 1
1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 1 1
0 0 0 1 0 0 0 1 0 1 1 0 0 1 0 1 0 0 0
1 1 0 0 0 0 0 0 1 0 0 1 0 0 1 1 1 1 0
1 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1
0 0 1 1 0 0 0 0 1 0 1 1 1 1 0 0 1 1 1
0 1 1 0 0 1 1 0 1 0 0 1 1 1 0 0 0 1 1
0 1 0 0 0 0 1 1 1 0 1 1 1 0 0 1 0 1 1
1 0 1 0 0 1 1 0 0 0 0 1 0 1 0 1 0 0 0
0 1 0 1 0 0 1 1 1 0 0 1 1 0 1 1 0 0 0
1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1
1 0 0 0 0 0 1 1 1 1 0 0 1 0 0 0 1 0 0
1 0 1 0 0 1 0 0 0 1 1 1 1 0 1 1 0 1 1
0 0 1 0 1 0 1 1 1 1 0 0 0 0 0 0 1 0 0
1 0 0 0 1 0 1 0 1 1 1 0 1 0 0 1 0 0 1
0 0 1 0 0 1 0 1 0 1 0 0 0 0 1 0 1 1 1
0 0 0 0 1 0 1 0 1 0 0 1 1 0 1 0 1 1 0
1 0 1 1 1 1 0 0 1 0 1 1 0 1 1 1 0 0 0
0 0 1 0 1 1 0 0 0 1 1 0 1 0 1 0 1 1 0
1 1 1 0 1 0 0 1 0 1 0 0 0 0 1 1 0 1 0
0 0 1 1 0 1 1 1 1 0 1 0 1 0 1 1 1 1 0
1 1 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0 0 1
1 0 0 0 0 1 0 0 0 1 1 0 0 1 1 1 1 0 0
1 0 1 1 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0
1 1 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0 0
1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 1 0
1 0 1 1 0 1 1 1 1 0 0 1 0 0 1 0 1 1 1
1 1 0 0 0 1 1 1 0 0 0 1 0 1 0 0 1 0 1
0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 0 0 0 0 1 0 1 0 0 1 0 1 1 0 1 0 1
1 0 1 0 1 0 1 1 0 0 1 0 1 0 1 1 0 1 0
0 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 1 0 0
0 0 0 1 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1
0 1 0 0 0 1 0 1 0 0 1 0 0 1 0 0 1 1 0
0 1 1 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1 0
1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 1 1 0
1 1 0 1 1 0 0 0 1 0 1 1 0 1 1 0 1 0 1
0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 0
0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0
0 1 1 0 0 0 1 0 1 1 1 1 0 0 1 1 1 0 1
```

```
Order of M3 is (19, 41)
0 1 0 0 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 1 0 1 0 1 1 0 1 0 0 1 1 1 1 1 0
1 1 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 1 1 1 0 1 1 1 0 0 1 0 0 1 1 0 1 1 1 0 1 0 1 0 0
0 0 1 0 1 0 1 0 0 0 1 1 1 0 0 1 0 0 0 1 0 1 0 1 0 1 0 1 1 1 0 0 1 0 0 1 1 0 0 0 0
0 1 0 0 1 1 0 1 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1
0 0 1 0 0 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 0 0 0 1 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 1 0
1 1 1 1 0 1 0 1 1 1 0 1 0 0 0 0 0 0 1 1 1 0 0 1 1 1 1 0 1 0 1 1 1 0 0 1 0 1 0 1 0
0 0 0 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 0 1 1 0
0 0 1 0 0 0 0 0 1 1 1 0 1 1 0 1 1 1 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 0 1 1 0 1 0 1 1
1 1 1 0 1 0 0 1 0 0 0 1 1 0 0 1 0 0 1 1 1 1 1 1 0 0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 1
0 0 0 1 0 1 1 0 1 1 0 0 0 0 1 1 0 1 0 1 1 0 1 0 1 1 1 1 0 1 0 1 1 1 0 0 1 0 0 0 0
0 1 0 0 1 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 1 0 1 0 0 0 0 1 0 0 0 0 1 1
0 0 0 0 1 0 1 0 0 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 0 0 0 1 1 0 0
1 1 1 1 1 0 0 0 0 1 0 1 0 0 1 0 1 1 0 0 0 1 0 0 0 1 0 1 0 1 1 0 0 1 0 0 1 1 0 0 0
0 1 1 1 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 1 1 0 1 0 1 0 0 0 0 1 0 0
0 0 1 0 0 1 1 0 1 0 1 1 1 1 0 0 1 0 0 1 1 0 0 1 1 1 0 0 0 1 1 0 1 0 1 0 1 1 0 0 1
1 0 0 0 0 1 0 1 1 1 0 0 1 0 0 0 1 1 0 1 0 1 1 1 0 1 0 1 0 1 0 1 1 0 0 0 1 1 1 1 0
0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1
0 1 1 0 1 1 0 1 0 0 1 1 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 0
1 1 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 1 1 1 0 1 0 0 1 1 0 1 0 1 0 0 0 0 1 1 1 0 0

Order of M4 is (41, 24)
0 0 1 1 1 0 1 0 1 0 1 1 0 0 1 0 0 1 0 1 0 1 0 1
0 0 1 1 0 1 0 1 0 0 1 1 1 0 1 1 0 1 0 0 0 1 1 0
1 0 1 1 0 0 1 1 0 0 0 1 0 0 1 0 0 1 1 1 0 0 0 1
1 1 1 0 0 0 1 1 0 1 1 0 1 1 1 1 0 1 1 0 0 1 1 0
1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 1
1 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0
0 0 1 1 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1 0 1 1 0 1
0 1 1 1 0 0 0 1 1 0 0 0 1 0 1 1 0 0 0 0 0 1 1 0
0 1 1 1 1 0 1 0 1 0 0 0 0 1 1 1 0 1 1 1 0 0 0 0
0 1 1 0 0 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 0 1 1 1
0 0 1 1 0 0 1 0 0 0 1 0 0 1 1 0 0 1 1 0 1 0 1 1
1 1 0 1 0 1 0 0 0 0 1 0 1 1 0 0 1 0 1 0 0 1 0 0
1 0 0 1 1 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 0 1 0 0
1 1 0 0 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1 1 1 0 1 0
1 1 1 0 1 0 1 0 1 1 1 0 1 0 1 0 1 0 0 1 1 1 1 0
1 0 0 1 1 1 1 0 1 0 1 1 1 1 0 1 1 0 0 1 1 1 0 1
1 1 1 0 1 1 0 1 0 0 0 1 1 1 0 1 0 0 1 0 0 0 0 0
0 1 0 1 0 0 1 0 1 1 0 0 1 1 1 0 1 1 0 1 0 0 0 0
0 0 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 1 0 0 1 0
0 1 0 1 0 1 0 0 1 1 0 0 0 1 1 0 0 1 1 1 1 0 1 1
1 1 0 0 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 0
0 0 0 0 0 1 0 1 0 1 1 0 0 1 1 0 1 1 0 1 0 0 1 0
1 0 1 0 1 0 1 0 1 1 0 0 0 1 1 1 1 1 1 1 1 0 1 1
0 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 0
0 1 1 0 0 1 1 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0
1 1 0 0 0 1 0 0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 0 1
1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 0 0
0 0 0 1 1 0 0 1 0 1 0 0 0 1 0 1 1 0 0 0 0 1 1 1
```

Bharatiya Vidya Bhavan's

Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India

(Autonomous College Affiliated to University of Mumbai)

```
Order of M5 is (24, 34)
0 0 0 1 1 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 1 0 0 1
0 0 1 0 0 0 0 1 0 1 0 1 0 1 1 0 0 1 1 1 0 0 1 0 0 0 1 1 0 1 0 1 0 0
0 1 1 1 1 1 0 1 1 1 0 0 1 1 1 0 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 0 0 0
0 1 0 0 1 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 1 1 0 1 1 1 0 1 1 1 0 1 1 1
1 0 1 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 1 1 1 1 0 0 1 1 1 1
1 0 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 0 1 0 1 0 0 0 1 1 0 1 0 1 0 0 0 0
0 0 1 0 1 0 0 1 1 0 1 1 0 1 0 1 0 0 1 1 1 1 0 1 0 1 1 0 0 0 0 1 1 1
0 1 0 1 1 1 1 0 1 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 0 0 0
0 1 1 1 1 0 1 1 1 1 1 1 0 0 1 0 1 1 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1 1
1 0 1 0 0 0 0 1 0 1 1 1 0 0 0 0 1 0 0 1 0 1 1 0 1 1 1 1 1 0 0 1 1 1
1 0 1 1 1 0 0 1 0 1 0 0 0 1 1 0 1 1 0 1 1 1 1 1 1 0 0 0 1 1 0 0 1 0
1 0 0 0 1 1 1 0 1 1 1 0 0 1 1 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 1 1 1
0 1 1 0 0 1 0 1 1 1 0 0 1 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 0 1 1 0
1 0 1 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 0
0 1 0 0 0 1 0 1 0 0 1 0 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 0 0 0 0 1 0 1
1 0 1 0 1 0 1 1 0 0 1 1 1 0 0 1 0 1 0 0 0 1 0 1 0 0 1 0 0 0 1 0 0 0
1 0 0 0 0 1 1 1 0 1 0 0 1 0 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1 0 1 0 1 0
1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 0 1 1 1 1 0 0 1 0 0 0 1 1 1 0 1 0
1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 0 0 1 0 0 1 0 0 1 1 1 0 1 0 1 1 0 1 1 0 1 1 1 1 1 1 0 0 0
1 1 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 1 0 1 1
1 0 0 1 1 1 0 0 1 0 1 1 1 0 0 0 0 1 1 0 1 1 1 0 0 1 1 1 1 0 0 0 0 0
1 1 1 1 0 0 1 1 0 0 0 1 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 1
1 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 1 0 1 0 1 1 1


Cost matrix
0        25916    50065    58748    80142
0        0        34276    38760    62624
0        0        0        18696    34200
0        0        0        0        33456
0        0        0        0        0


Parenthesization Matrix
0 1 2 2 2
0 0 2 2 2
0 0 0 3 4
0 0 0 0 4
0 0 0 0 0

Optimal parenthesization : ((M1*M2)*((M3*M4)*M5))
Postfix expression :  M1 M2* M3 M4* M5**
```

```
Result Of Multiplication :
29792 22035 25876 22924 27170 26482 22879 39544 23147 23365 28181 28817 13723 28595 30979 26995 21241 32484 34213 25553 34094 29746 28175 22108 23270 24680 27895 33142 28405 26684 21561 24186 30911 24096
28479 21113 24787 21941 26009 25342 21867 37861 22186 22432 26980 27594 13163 27372 29662 25836 20363 31127 32776 24469 32626 28449 26997 21130 22290 23626 26715 31726 27193 25553 20635 23137 29566 23018
27630 20440 24076 21292 25268 24573 21212 36762 21534 21726 26163 26766 12759 26600 28757 25068 19744 30192 31845 23765 31673 27638 26179 20585 21589 22924 25928 30783 26295 24793 20014 22448 28693 22371
21790 16166 18988 16815 19922 19364 16754 29010 16981 17146 20661 21137 10024 20978 22736 19799 15581 23833 25110 18768 24948 21740 20648 16217 17020 18086 20433 24278 20778 19575 15795 17719 22646 17663
27299 20264 23768 21047 24934 24256 20968 36339 21267 21479 25876 26480 12578 26249 28466 24808 19530 29882 31421 23502 31249 27241 25884 20287 21321 22649 25592 30397 26065 24501 19769 22192 28371 22096
20748 15419 18065 15980 18962 18441 15951 27612 16171 16361 19712 20156 9581 19981 21657 18875 14841 22717 23915 17870 23761 20717 19662 15422 16255 17238 19482 23109 19814 18640 15046 16871 21543 16795
29267 21689 25481 22546 26731 25995 22464 38926 22772 22997 27740 28369 13495 28130 30475 26538 20918 31986 33639 25193 33514 29259 27726 21764 22893 24289 27436 32559 27924 26235 21164 23790 30400 23712
25665 19086 22456 19790 23427 22801 19710 34249 20010 20195 24310 24963 11839 24698 26809 23324 18380 28158 29565 22180 29393 25636 24407 19115 20021 21310 24108 28597 24525 23059 18560 20957 26721 20828
28432 21066 24829 21887 25975 25302 21851 37836 22147 22369 26931 27612 13173 27313 29600 25737 20320 31093 32704 24453 32562 28450 26977 21133 22243 23633 26706 31674 27154 25491 20554 23164 29545 23034
34849 25846 30375 26825 31836 30977 26794 46390 27146 27382 33007 33838 16072 33550 36342 31652 24899 38144 40149 30012 39919 34830 33021 25968 27217 28943 32694 38801 33241 31277 25215 28374 36233 28257
22816 16907 19825 17605 20852 20342 17574 30291 17753 17924 21581 22046 10532 21918 23771 20635 16299 24903 26196 19553 26169 22826 21622 16938 17858 18971 21368 25402 21865 20455 16492 18533 23720 18487
30235 22372 26312 23291 27587 26900 23258 40177 23509 23732 28606 29271 13943 29003 31449 27393 21580 32996 34726 25925 34577 30162 28639 22425 23579 25093 28333 33670 28866 27074 21876 24579 31412 24471
26251 19509 22878 20214 24012 23367 20192 34935 20484 20671 24914 25487 12137 25284 27387 23844 18777 28731 30213 22606 30108 26289 24892 19545 20564 21840 24645 29218 25121 23585 19018 21367 27318 21276
34431 25501 29984 26497 31447 30651 26478 45772 26809 27058 32600 33390 15904 33092 35850 31195 24591 37621 39598 29575 39422 34416 32615 25580 26919 28620 32288 38349 32867 30854 24893 28025 35789 27898
34438 25509 30080 26528 31448 30624 26450 45864 26811 27064 32595 33418 15913 33124 35875 31229 24609 37664 39642 29650 39447 34428 32676 25627 26894 28588 32347 38370 32844 30884 24911 28067 35796 27918
28280 20973 24612 21793 25828 25136 21745 37640 22021 22208 26780 27410 13023 27212 29479 25727 20212 30929 32571 24327 32382 28217 26775 21054 22050 23443 26501 31481 26943 25384 20506 22975 29388 22898
26006 19258 22620 20050 23756 23170 19960 34591 20265 20417 24558 25180 12001 24995 27088 23556 18560 28437 29946 22333 29833 26005 24667 19330 20313 21536 24365 28998 24800 23350 18801 21147 26993 21036
31728 23509 27566 24437 28995 28250 24365 42139 24713 24947 30035 30719 14651 30495 33043 28751 22642 34642 36490 27234 36389 31741 30039 23563 24854 26289 29722 35320 30282 28495 22976 25752 32906 25638
31368 23321 27299 24175 28666 27930 24171 41749 24446 24680 29741 30424 14479 30228 32757 28529 22448 34350 36142 26984 35958 31339 29727 23351 24534 26106 29432 34944 30012 28163 22739 25522 32650 25444
26286 19485 22891 20254 24030 23402 20242 34940 20473 20672 24917 25464 12152 25274 27376 23834 18787 28710 30230 22574 30123 26293 24902 19550 20561 21862 24662 29259 25130 23574 19041 21363 27331 21301
28910 21420 25258 22242 26376 25667 22191 38493 22515 22708 27367 28082 13338 27776 30093 26206 20665 31630 33263 24922 33075 28897 27431 21526 22549 24018 27141 32186 27591 25902 20893 23574 30095 23457
29555 21870 25715 22725 26986 26298 22697 39273 23013 23207 27948 28650 13646 28402 30752 26778 21100 32301 33988 25374 33836 29546 27984 21955 23085 24536 27698 32902 28200 26471 21358 24039 30719 23917
26990 19967 23480 20754 24611 23985 20727 35828 20984 21167 25533 26160 12425 25876 28066 24414 19218 29427 30985 23167 30872 26956 25545 20025 21085 22369 25283 30034 25770 24187 19513 21959 28020 21848
31503 23438 27476 24318 28799 28007 24179 42015 24582 24832 29869 30584 14536 30373 32921 28674 22581 34566 36338 27196 36135 31479 29930 23461 24642 26147 29580 35115 30100 28333 22832 25642 32763 25518
29885 22165 26061 23043 27361 26621 23014 39790 23308 23522 28317 28982 13826 28827 31186 27135 21381 32715 34479 25721 34311 29929 28336 22302 23388 24871 28070 33298 28535 26840 21639 24317 31086 24252
26126 19383 22696 20104 23856 23274 20114 34688 20354 20558 24731 25301 12068 25129 27233 23704 18662 28541 30088 22422 29967 26121 24731 19426 20458 21702 24483 29090 24978 23461 18936 21216 27145 21139
24367 18033 21211 18771 22272 21710 18734 32367 18988 19137 23062 23602 11263 23390 25334 22024 17383 26587 28006 20895 27913 24383 23085 18091 19060 20241 22851 27155 23283 21842 17637 19814 25321 19735
23180 17228 20284 17853 21178 20576 17785 30948 18089 18252 21950 22568 10684 22365 24227 21087 16624 25473 26756 20068 26557 23170 22028 17312 18090 19279 21773 25824 22124 20815 16743 18933 24169 18829
24904 18493 21768 19179 22707 22109 19119 33193 19370 19605 23595 24202 11493 23951 25994 22649 17820 27277 28649 21477 28486 24833 23643 18497 19439 20650 23392 27725 23777 22352 18029 20305 25879 20156
28969 21489 25199 22294 26429 25746 22313 38480 22541 22744 27440 28074 13322 27831 30199 26317 20665 31618 33314 24901 33166 28911 27409 21550 22603 24030 27131 32204 27673 26007 20993 23548 30114 23473
22383 16618 19500 17250 20433 19893 17214 29800 17420 17624 21229 21722 10323 21522 23348 20358 16020 24497 25755 19257 25601 22306 21217 16613 17484 18590 20997 24926 21380 20074 16217 18207 23265 18116

Estimated Multiplication count: 80142
Actual Multiplication count: 80142
PS D:\Tejas\clg\daa\Experiment 04\code>
```

2) Number of Matrices = 10

```
PS D:\Tejas\clg\daa\Experiment 04\code> ./a
Enter number of matrices (<= 10) : 10

p[i] = [ 20, 30, 27, 38, 15, 44, 25, 20, 44, 28, 23]

Order of M1 is (20, 30)
0 1 1 1 0 0 1 1 0 1 1 0 0 0 1 1 0 0 1 1 1 1 0 1 0 1 1 0 0 1
0 1 0 1 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 1 0 1 0
0 0 0 1 1 1 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1
0 0 1 0 1 0 1 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 1 1 1 1 1 0 0
0 1 0 1 1 0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 1 1 1 0
1 0 1 1 1 1 0 0 0 1 1 1 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 0 0 0
0 0 0 0 1 0 0 1 0 0 1 1 1 1 1 1 0 0 1 0 1 0 0 1 0 0 0 0 1 1
1 1 0 1 1 1 1 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 1 1 0 0 1 0
1 1 1 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 1 0 1 0 0 1 0 0 1 0 0 1
1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0 1 1 0 1 0
0 0 0 1 0 0 0 0 1 1 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1
1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 0 0 0 1 1
1 0 1 0 1 0 1 0 1 1 1 0 1 1 0 0 1 0 1 1 1 1 0 0 0 1 0 1 0 1
0 1 1 0 1 1 0 1 1 1 0 1 0 1 0 1 0 1 0 0 1 1 0 0 1 1 0 0 0 1
0 0 0 1 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 0 0 1 0 0 0 0 0
1 1 0 1 1 1 0 0 0 0 0 1 1 0 0 0 1 0 1 0 1 0 1 1 1 0 0 0 0 1
0 0 0 0 1 0 1 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 1 0 1 1 0 1
0 0 0 1 0 0 1 0 1 1 1 0 1 1 1 0 1 1 0 1 1 1 0 0 1 0 0 0 0 0
1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 1 0
1 1 1 0 1 1 1 1 0 1 1 0 1 0 0 1 1 1 0 0 0 1 0 0 0 0 1 1 1 1

Order of M2 is (30, 27)
0 0 0 1 1 1 0 1 0 1 0 0 1 0 0 1 0 1 1 0 0 1 0 1 1 1 0
1 1 1 0 0 1 1 0 0 1 0 1 0 0 1 1 0 0 1 0 1 1 0 1 1 1 1
1 1 1 0 0 1 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0
0 0 0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0
0 1 1 0 0 0 1 1 0 1 0 1 0 1 0 1 1 1 0 1 1 0 1 1 1 1 0
1 1 0 1 1 0 1 0 1 1 0 0 0 1 0 1 0 0 1 1 1 1 1 1 0 1 1
0 1 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 1 0 0 1 1 1 1 1 0 0
0 0 0 1 0 1 0 1 0 1 0 0 1 1 0 1 0 0 1 0 0 0 0 1 0 1 0
0 0 1 0 1 1 1 1 1 0 0 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0
1 0 1 1 0 0 0 0 1 1 1 1 1 0 0 1 0 0 0 0 0 1 1 1 1 0 0
1 0 1 0 1 1 0 0 0 1 0 1 0 1 1 1 1 0 1 0 0 0 0 1 0 0 1
1 1 0 1 1 1 1 0 1 1 0 0 0 0 1 1 0 1 1 1 0 1 0 0 1 1 1
0 1 0 0 0 1 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1
```

```
1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0
1 0 0 1 1 1 1 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 1 1 0 0 1
1 1 1 0 1 0 1 1 0 0 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 1
0 1 1 1 1 0 1 0 1 0 0 0 1 0 1 1 0 1 1 1 1 1 0 0 0 0 1
1 1 1 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0
1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0
1 0 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1 0 0
1 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 1 0 1 1 1 0 1 1 1
1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 1 0
0 1 1 1 0 1 1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 1 1
1 1 0 0 0 0 0 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1
1 1 0 1 0 0 1 1 0 1 1 0 1 1 1 0 1 0 0 1 1 1 0 0 0 0 1
0 1 1 1 1 1 1 1 1 0 0 1 1 0 0 0 1 0 1 1 1 0 1 0 0 0 1
1 1 1 1 1 0 1 0 0 0 0 1 1 0 1 0 0 1 0 1 0 0 1 1 1 1 0
0 1 1 0 1 1 0 0 1 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 1 0

Order of M3 is (27, 38)
0 1 1 0 1 1 1 0 1 0 0 1 0 0 1 1 1 1 0 1 0 0 1 0 0 1 1 1 0 1 1 0 0 1 1 1 1 0
0 0 0 0 1 0 0 0 1 1 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 1 1 1 1 0 0 1 0 1 0 1 0 0
1 1 0 0 1 1 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 1 1 0 1 0 1 0 0 0 1
0 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 1 0 1 1 0 0 0 1 0 1 1 1
1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 1 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 1 0 0 1 1 1 1
1 0 0 1 0 1 0 0 1 1 1 0 1 0 0 0 1 0 1 1 1 1 1 1 1 0 0 1 1 1 0 1 0 0 0 0 0 1
1 1 0 0 1 0 1 1 0 0 0 1 0 1 1 1 1 1 0 1 1 0 1 1 0 0 1 0 1 1 1 0 1 1 1 1 1 1
1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 0 1 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 1 0 1 1
0 0 0 0 0 1 1 0 1 0 0 0 1 1 1 0 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1
0 0 0 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 0 1 1 0 1 1 1 0 1 1
1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 1 1 0 1 1
0 1 1 1 0 1 0 1 0 1 1 1 0 1 0 0 1 1 1 0 0 1 0 1 0 1 1 0 0 1 0 0 0 0 0 0 1 1
0 0 1 0 1 1 1 1 1 0 1 1 1 0 1 1 0 0 1 0 0 0 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 0
1 1 1 0 1 1 1 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0 0 0 1 0
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 1 1 1 0 0 1 1 1 0 1 0 1 1 1 0 0 0 1 1
1 1 0 0 0 1 0 1 1 1 1 0 1 0 0 0 1 1 1 1 1 1 1 1 0 1 0 0 0 0 1 0 1 0 1 1 0 0
0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0
1 1 0 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 1 0 1 1 1 0 0 1 0 0 1 1 1 0 1
1 0 0 0 1 0 0 0 0 1 1 1 0 1 1 1 1 1 1 0 1 0 0 1 1 1 0 1 1 1 1 1 0 0 1 1 0 1
0 0 0 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 1 1 0 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 1
1 1 0 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 1 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 1 1 1 1
1 0 0 1 1 1 0 0 0 1 0 1 1 0 0 0 0 1 0 1 0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 0
1 1 0 0 1 1 1 1 0 1 1 0 1 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 0 1
0 1 0 0 0 0 1 1 0 1 1 0 1 1 1 1 0 1 1 0 0 1 0 1 1 0 0 0 0 1 0 1 1 0 0 1 1 0
```

```
1 1 0 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 1 1 0 0 1 0 1 0 1 0 1 0 0 1 0 1 1 1 1 1
1 0 0 1 1 1 0 0 0 1 0 1 1 0 0 0 0 1 0 1 0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 0
1 1 0 0 1 1 1 1 0 1 1 0 1 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 0 1
0 1 0 0 0 0 1 1 0 1 1 0 1 1 1 1 0 1 1 0 0 1 0 1 1 0 0 0 0 1 0 1 1 0 0 1 1 0
0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 1 1 1 1 0 1 0 0 1 1 1 0
1 1 1 0 1 1 0 0 0 1 1 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1
1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 1 0 0 1

Order of M4 is (38, 15)
1 1 1 0 0 1 0 1 1 0 0 1 0 1 0
0 0 1 0 0 0 1 0 1 1 1 0 1 1 1
0 0 0 0 0 0 1 1 0 1 0 0 0 1 0
0 1 1 1 1 1 0 1 0 1 0 0 1 0 0
0 1 1 0 0 1 1 0 0 0 0 0 1 0 1
0 1 1 0 0 1 0 1 1 0 0 0 0 1 1
1 0 1 1 0 0 1 0 1 1 1 1 0 0 0
0 1 0 1 0 1 0 0 1 1 1 0 0 1 1
1 0 1 0 0 1 0 1 1 0 0 0 0 0 0
1 0 1 0 1 0 0 0 1 0 1 0 1 0 1
1 1 0 1 0 0 1 0 1 1 0 0 0 1 0
0 0 0 1 1 0 0 1 0 1 1 0 0 0 1
0 1 1 0 0 1 0 1 1 0 0 0 1 0 1
1 1 0 1 0 1 1 1 1 0 0 0 0 1 1
1 0 0 0 1 1 0 0 1 1 0 1 0 0 1
0 0 0 1 1 0 1 1 0 0 1 1 1 1 0
1 1 1 1 1 1 0 0 0 1 0 0 1 0 1
1 0 1 0 0 1 0 0 0 0 1 0 1 1 1
0 0 0 1 0 0 1 0 1 1 1 1 0 0 1
0 0 0 0 1 0 1 1 1 1 1 0 1 0 0
0 1 1 1 0 0 0 1 0 0 1 0 1 1 1
1 1 1 0 0 0 1 0 1 1 1 1 0 1 0
1 1 1 1 0 1 1 1 1 1 0 0 0 1 0
0 0 1 1 1 1 1 0 1 0 0 0 0 1 1
0 1 1 0 0 0 1 1 0 1 1 1 0 1 0
0 1 1 0 0 0 1 1 0 0 1 1 1 0 1
0 1 1 1 0 0 1 1 1 1 0 0 0 1 1
0 0 1 0 1 1 1 1 1 1 1 1 0 0 0
0 0 1 0 0 1 0 0 1 1 1 1 1 1 1
0 1 1 1 1 1 1 0 0 0 0 0 0 1 1
1 0 0 0 0 1 1 0 0 0 0 1 1 1 1
1 1 0 0 1 0 0 1 0 1 0 1 0 1 1
```

```
Windows PowerShell
1 0 1 1 1 0 1 1 0 0 1 1 0 1 0
0 1 0 1 0 1 1 0 0 0 1 1 0 1 0
0 0 0 0 0 1 1 0 1 0 0 1 1 1 1
0 1 0 0 0 1 1 1 0 1 0 1 0 1 1

Order of M5 is (15, 44)
1 0 0 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 1 0 1 0 0 1 1 1 0 1 1
1 1 1 0 0 0 0 0 1 1 0 1 1 1 0 1 1 0 1 0 1 0 1 1 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 1 1 0
1 1 0 0 1 1 0 1 0 1 0 1 1 1 1 1 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 0 1 1 1 1
1 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0 1 0
0 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1
0 1 0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 0 1 1 1 1 1 1 0 1 1
0 1 0 1 0 0 1 1 1 0 0 0 0 1 0 0 0 1 1 1 1 1 0 1 0 0 0 0 1 1 0 1 0 0 0 0 0 0 1 1 0 1 1 0
1 0 0 1 1 1 1 0 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 1 1 1 1 0 1 1 1 1 0 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 1 1 1 0 0 1 1 0 1 1 1 0 0 1 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 1
1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 1 0 1 1 0 0 1 1 0 1 1 1 0 0 1 1 0 1 0 1 1 0 0 0 1 1
1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 0
0 0 1 1 1 1 1 0 1 0 0 1 0 0 1 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 0 1 0 1 1 0 1 1 0 1 1
0 1 0 1 1 0 1 0 1 1 1 0 1 0 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 1 1 0 1 1 0 0 0 1 0 1 1 1 1 1
1 1 0 0 0 0 1 0 0 1 0 1 1 1 0 1 1 0 0 1 0 0 1 0 1 1 0 1 0 0 0 1 0 1 0 0 0 0 1 1 1 1 1 1
1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 0 1 1 0 1 0 0 0 0 0 0 1 1 0 1 0 1 1 0 1 1

Order of M6 is (44, 25)
1 0 1 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 1 1 0 0
0 0 0 1 1 0 0 0 1 1 1 1 0 1 0 0 0 0 1 1 0 1 1 1 1
1 0 1 1 0 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 1 0 0 1 0
1 0 0 1 0 0 0 1 1 1 0 1 0 1 0 1 1 0 1 1 1 1 1 0 0
1 1 1 1 0 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 0 1 1 1 1
0 0 1 1 1 1 1 1 0 1 0 1 1 0 0 0 1 0 0 1 0 1 0 1 1
0 1 0 0 0 0 0 1 1 0 0 1 0 1 1 1 1 0 0 0 0 1 1 0 1
0 1 0 1 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 0 0 0 1 0 1
1 1 0 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 0 0 0 1 0 1 0
0 1 1 0 1 0 1 1 1 1 0 0 0 1 0 0 1 0 0 1 1 0 1 1 0
0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1 1 0 1
1 0 0 0 1 0 1 0 0 1 0 0 0 0 1 1 1 1 0 1 1 1 1 0 1
1 0 1 1 1 1 0 0 0 1 1 1 0 1 1 0 1 0 1 1 0 1 1 0 1
0 1 0 0 1 0 1 1 0 0 1 0 0 1 0 1 0 1 0 0 0 1 1 1 1
0 0 1 1 1 1 0 1 1 1 0 0 0 1 0 0 1 0 0 1 1 1 1 1 0
0 1 0 1 0 0 1 0 0 1 1 0 1 0 0 1 1 0 0 0 1 0 1 0 1
0 1 0 0 0 1 1 0 1 1 0 1 0 1 1 0 0 0 0 1 1 1 0 1 1
0 1 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 0 1 0 1 1 1 1 0
```

Snipping Tool

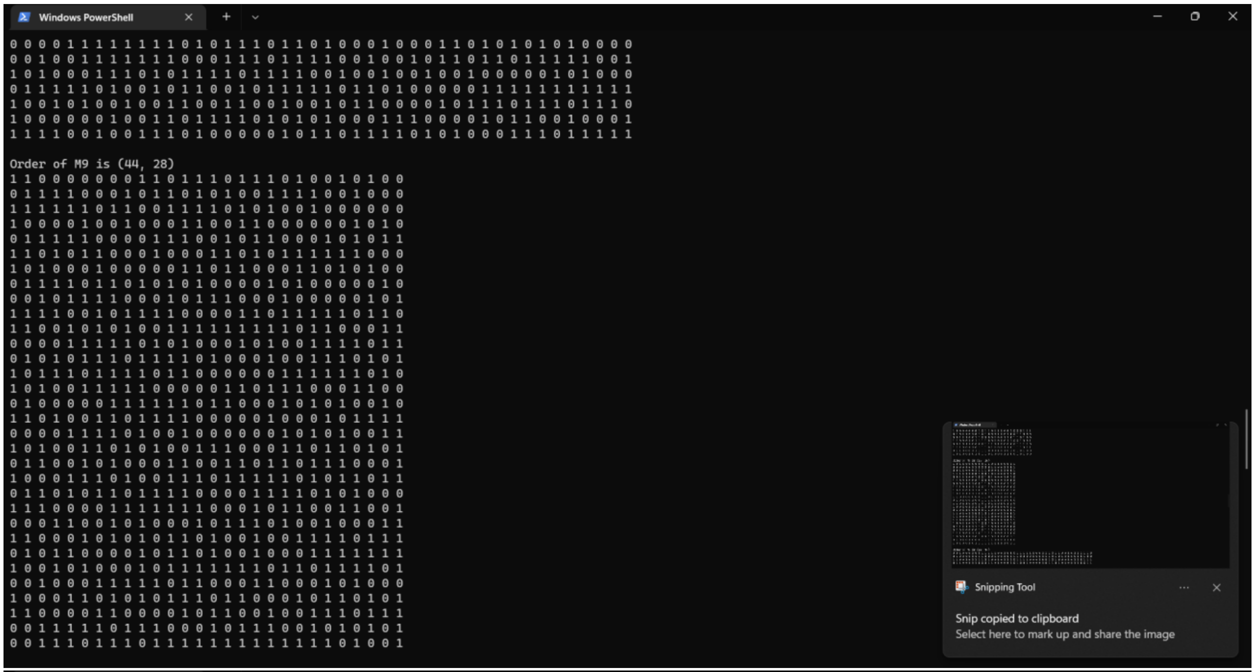
Snip copied to clipboard

Select here to mark up and share the image

```
Windows PowerShell
1 0 0 0 0 0 0 0 1 0 1 1 0 1 1 1 0 0 1 0 0 0 1 1 1
0 1 0 1 1 0 1 0 1 0 1 0 0 0 1 1 0 1 0 0 1 1 0 0 0
0 0 1 1 1 0 0 1 1 0 0 0 0 1 0 0 1 1 0 1 1 0 1 0 0
1 1 0 0 0 1 0 1 1 1 1 0 0 1 0 1 0 1 1 0 0 1 0 0 1
0 0 1 0 1 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 1 1 1 0
1 1 1 0 0 1 1 0 1 1 1 0 1 1 0 0 1 1 0 1 1 1 0 1 1
0 1 1 0 0 0 1 1 1 1 1 0 0 0 1 0 1 0 1 0 1 0 1 0 0
1 0 1 0 0 1 0 1 0 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 0

Order of M7 is (25, 20)
1 0 1 1 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1
0 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 1 1 0 1
0 0 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 1
1 1 1 1 1 1 0 0 1 0 0 1 0 1 0 0 0 1 1 0
0 0 1 1 1 0 0 1 0 0 1 0 1 0 0 0 1 0 1 0
1 1 0 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0
0 0 0 1 1 1 1 0 0 1 1 0 1 0 1 0 0 1 1 0
1 1 1 0 1 1 1 1 0 0 1 1 0 1 0 1 0 0 0 0
1 0 1 1 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 0
1 0 1 1 1 1 1 1 0 0 0 1 0 1 0 0 1 0 0 0
1 1 0 0 0 0 0 0 0 1 1 0 1 0 1 1 1 0 0 1
0 1 0 1 0 0 1 0 0 1 1 1 1 1 1 0 1 0 0 0
1 1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 0 1 1 0
1 0 0 1 0 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1
0 1 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1
1 1 0 1 0 1 1 0 0 1 0 1 0 1 0 1 1 0 0 1
1 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1
1 1 0 1 0 0 1 1 0 1 0 1 0 1 1 1 1 0 1 1
1 0 0 1 1 0 1 1 1 0 0 1 0 0 1 0 1 1 1 0
0 1 1 1 1 1 1 1 1 0 1 1 0 0 1 0 0 0 0 0
0 1 0 1 0 0 0 0 1 0 1 1 0 1 0 0 1 1 0 0
1 1 1 0 0 1 1 1 0 1 1 1 1 0 0 0 1 1 0 0
1 0 0 0 1 1 1 0 1 0 0 1 0 1 1 1 0 0 1 1
0 1 0 1 0 1 0 0 0 1 1 1 1 0 1 0 1 1 0 0
1 0 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1

Order of M8 is (20, 44)
0 0 1 0 0 0 1 0 0 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 0 0 1 1 1 0 1 1 0 0 1 0 1 1 1 1 1 0
0 1 0 0 0 0 1 1 1 1 0 0 0 1 0 0 1 0 1 1 1 0 0 0 0 1 0 0 1 1 1 0 1 0 1 0 1 0 0 1 0 1 0 0
1 1 0 0 1 0 1 1 1 1 0 1 0 0 1 0 1 1 1 0 1 1 1 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 1 1 0 1 1 1
0 1 1 0 0 1 1 1 0 0 0 1 1 0 1 0 0 0 1 0 1 0 0 1 0 0 0 0 1 0 1 1 0 1 0 1 0 1 1 1 0 0 1 1
```



```
Windows PowerShell
0 0 0 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 0 1 0 0 0 1 0 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 0
0 0 1 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 1 0 0 1 0 0 1 0 1 1 0 1 1 0 1 1 1 1 1 0 0 1
1 0 1 0 0 0 1 1 1 0 1 0 1 1 1 1 0 1 1 1 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0
0 1 1 1 1 1 0 1 0 0 1 0 1 1 0 0 1 0 1 1 1 1 1 0 1 1 0 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1
1 0 0 1 0 1 0 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 1 1 0 1 1 1 0
1 0 0 0 0 0 0 1 0 0 1 1 0 1 1 1 1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 0 0 1 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 1 0 1 1 0 1 1 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1

Order of M9 is (44, 28)
1 1 0 0 0 0 0 0 0 1 1 0 1 1 1 0 1 1 1 0 1 0 0 1 0 1 0 0
0 1 1 1 1 0 0 0 1 0 1 1 0 1 0 1 0 0 1 1 1 1 0 0 1 0 0 0
1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 0 1 0 1 0 0 1 0 0 0 0 0 0
1 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 1 0 1 0
0 1 1 1 1 1 0 0 0 0 1 1 1 0 0 1 0 1 1 0 0 0 1 0 1 0 1 1
1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 0 1 0 1 1 1 1 1 1 0 0 0
1 0 1 0 0 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 0 1 0 1 0 0
0 1 1 1 1 0 1 1 0 1 0 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 1 0
0 0 1 0 1 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 0 0 0 0 1 0 1
1 1 1 1 0 0 1 0 1 1 1 1 0 0 0 0 1 1 0 1 1 1 1 1 0 1 1 0
1 1 0 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 1 1
0 0 0 0 1 1 1 1 1 0 1 0 1 0 0 0 1 0 1 0 0 1 1 1 1 0 1 1
0 1 0 1 0 1 1 1 0 1 1 1 1 0 1 0 0 0 1 0 0 1 1 1 0 1 0 1
1 0 1 1 1 0 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 1 0 1 0
1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 1 1 0 1 1 1 0 0 0 1 1 0 0
0 1 0 0 0 0 0 1 1 1 1 1 1 0 1 1 0 0 0 1 0 1 0 1 0 0 1 0
1 1 0 1 0 0 1 1 0 1 1 1 1 0 0 0 0 0 1 0 0 0 1 0 1 1 1 1
0 0 0 0 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 1 1
1 0 1 0 0 1 1 0 1 0 1 0 0 1 1 1 0 0 0 1 1 0 1 1 0 1 0 1
0 1 1 0 0 1 0 1 0 0 0 1 1 0 0 1 1 0 1 1 0 1 1 1 0 0 0 1
1 0 0 0 1 1 1 0 1 0 0 1 1 1 0 1 1 1 1 1 0 1 0 1 1 0 1 1
0 1 1 0 1 0 1 1 0 1 1 1 1 0 0 0 0 1 1 1 1 0 1 0 1 0 0 0
1 1 1 0 0 0 0 1 1 1 1 1 1 1 0 0 0 1 0 1 1 0 0 1 1 0 0 1
0 0 0 1 1 0 0 1 0 1 0 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 1 1
1 1 0 0 0 1 0 1 0 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 0 1 1 1
0 1 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 1 0 0 0 1 1 1 1 1 1 1
1 0 0 1 0 1 0 0 0 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 1
0 0 1 0 0 0 1 1 1 1 1 0 1 1 0 0 0 1 1 0 0 0 1 0 1 0 0 0
1 0 0 0 1 1 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 0 1
1 1 0 0 0 0 1 1 0 0 0 0 1 0 1 1 0 0 1 0 0 1 1 1 0 1 1 1
0 0 1 1 1 1 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 1 0 1 0 1 0 1
0 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1
```


```
Windows PowerShell
0 0 1 1 0 0 0 0 1 1 1 1 0 0 1 0 1 1 1 1 0 1 1 0 0 0 0 0
1 1 1 1 1 1 0 1 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 0 0 1
0 0 1 1 0 0 1 1 0 0 1 0 1 1 0 0 0 0 1 0 1 1 1 1 1 1 0 1
1 0 1 1 0 1 1 0 0 1 1 1 1 0 1 1 0 0 0 0 0 0 1 0 1 0 0 1
1 0 0 1 0 0 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 0 1 0 0 0 0
1 0 1 0 1 0 1 0 1 0 1 1 1 1 0 0 0 1 1 0 1 1 1 1 0 1 1 1
1 1 1 1 1 1 0 1 1 1 0 0 1 1 0 0 1 0 1 1 0 1 0 0 0 0 1 0
0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 1
1 0 1 1 0 0 0 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 0 0 1 1
0 0 0 0 0 1 0 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 1 0 0
1 0 0 0 1 1 1 0 1 1 0 0 0 1 0 1 0 1 1 1 0 1 0 0 1 1 0 0
1 1 0 1 0 0 0 1 1 1 0 0 0 1 1 1 1 0 1 0 1 1 1 0 0 0 1 0

Order of M10 is (28, 23)
1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 1 0 0 1
1 0 1 1 1 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 1 0 1
1 0 0 1 0 1 0 1 0 1 0 1 1 1 1 0 0 0 0 0 1 1 0
0 0 0 0 1 0 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 1 0
0 0 0 0 1 1 1 1 0 0 1 0 0 1 0 1 0 0 1 1 1 1 1
1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 1
1 0 0 0 1 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1 0 1 0
1 1 1 1 1 1 1 0 0 1 0 1 0 0 0 1 1 0 1 0 1 1 1
0 1 1 1 0 0 0 1 0 1 0 0 1 0 1 0 1 0 0 1 0 0 0
1 1 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 0 1 1 0 0
1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 0 0 0 0 1 1 1
1 1 0 1 1 1 0 0 1 0 1 0 1 1 1 1 0 0 1 1 0 0 0
1 0 0 1 0 0 1 1 0 1 1 1 1 0 0 1 0 1 0 1 0 0 0
0 1 1 0 0 1 0 0 1 1 1 1 0 0 1 0 1 1 1 0 1 1 0
0 1 0 1 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0 0 1 0 0
0 0 0 0 1 1 1 0 1 0 0 1 1 0 0 0 1 1 0 1 1 0 0
1 1 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 1
1 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1
1 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 0 0 1 1 1 0 0
1 1 0 0 0 0 1 0 0 1 0 1 0 1 1 1 0 0 1 0 0 0 1
1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 0 0
0 1 1 0 1 0 1 1 1 1 1 0 0 1 0 1 1 1 0 0 1 1 0
1 1 0 1 1 1 1 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 0
1 1 0 0 1 1 0 1 0 0 0 1 1 0 1 1 0 0 1 0 1 0 0
1 1 0 1 1 1 0 0 1 0 1 1 1 1 1 1 0 1 1 0 1 1 1
1 0 0 0 0 1 1 0 1 1 1 0 1 0 1 0 0 1 1 1 1 0 1
1 0 1 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 1 0 1
```

```
Windows PowerShell

Cost matrix
0       16200   36720   36540   49740   60540   66540   84140   100620  108780
0       0       30780   27540   47340   55290   60540   84540   95820   103230
0       0       0       15390   33210   42015   47490   70410   82410   90045
0       0       0       0       25080   30750   35400   62280   71640   78450
0       0       0       0       0       16500   24000   37200   55680   65340
0       0       0       0       0       0       22000   60720   69440   74320
0       0       0       0       0       0       0       22000   38640   49020
0       0       0       0       0       0       0       0       24640   37520
0       0       0       0       0       0       0       0       0       28336
0       0       0       0       0       0       0       0       0       0


Parenthesization Matrix
0 1 2 1 4 4 4 7 4 4
0 0 2 2 4 4 4 4 4 4
0 0 0 3 4 4 4 4 4 4
0 0 0 0 4 4 4 4 4 4
0 0 0 0 0 5 6 7 8 9
0 0 0 0 0 0 6 7 6 6
0 0 0 0 0 0 0 7 7 7
0 0 0 0 0 0 0 0 8 9
0 0 0 0 0 0 0 0 0 9
0 0 0 0 0 0 0 0 0 0

Optimal parenthesization : ((M1*(M2*(M3*M4)))*(((((M5*M6)*M7)*M8)*M9)*M10))
Postfix expression :  M1 M2 M3 M4*** M5 M6* M7* M8* M9* M10**

Result Of Multiplication :
2.452e+010 1.88683e+010 1.12789e+010 1.63807e+010 2.08541e+010 1.5714e+010 1.64145e+010 1.44227e+010 1.67275e+010 2.12797e+010 2.11121e+010 1.99851e+010 2.15729e+010 1.46
545e+010 2.05626e+010 1.65174e+010 1.48818e+010 1.34608e+010 1.45183e+010 1.77039e+010 2.12866e+010 1.38146e+010 1.34805e+010
1.51968e+010 1.1694e+010 6.99033e+009 1.01523e+010 1.29248e+010 9.73908e+009 1.01732e+010 8.93874e+009 1.03672e+010 1.31885e+010 1.30847e+010 1.23862e+010 1.33702e+010 9.
0824e+009 1.27441e+010 1.0237e+010 9.22332e+009 8.34259e+009 8.99803e+009 1.09723e+010 1.31928e+010 8.5619e+009 8.35479e+009
1.48695e+010 1.14421e+010 6.83976e+009 9.93361e+009 1.26464e+010 9.52931e+009 9.95411e+009 8.74621e+009 1.01439e+010 1.29045e+010 1.28029e+010 1.21194e+010 1.30822e+010 8
.88677e+009 1.24696e+010 1.00165e+010 9.02466e+009 8.1629e+009 8.80422e+009 1.0736e+010 1.29087e+010 8.37747e+009 8.17484e+009
2.24228e+010 1.72544e+010 1.03142e+010 1.49796e+010 1.90705e+010 1.437e+010 1.50105e+010 1.31891e+010 1.52967e+010 1.94596e+010 1.93064e+010 1.82757e+010 1.97277e+010 1.3
401e+010 1.88039e+010 1.51047e+010 1.3609e+010 1.23094e+010 1.32765e+010 1.61896e+010 1.9466e+010 1.2633e+010 1.23275e+010
2.66753e+010 2.05267e+010 1.22703e+010 1.78205e+010 2.26872e+010 1.70952e+010 1.78573e+010 1.56904e+010 1.81977e+010 2.31501e+010 2.29678e+010 2.17417e+010 2.3469e+010 1.
59425e+010 2.237e+010 1.79692e+010 1.61899e+010 1.46439e+010 1.57944e+010 1.926e+010 2.31577e+010 1.50289e+010 1.46653e+010
2.49534e+010 1.92017e+010 1.14782e+010 1.66702e+010 2.12227e+010 1.59917e+010 1.67046e+010 1.46775e+010 1.70231e+010 2.16557e+010 2.14852e+010 2.03383e+010 2.19541e+010 1
```

```
Windows PowerShell

Result Of Multiplication :
2.452e+010 1.88683e+010 1.12789e+010 1.63807e+010 2.08541e+010 1.5714e+010 1.64145e+010 1.44227e+010 1.67275e+010 2.12797e+010 2.11121e+010 1.99851e+010 2.15729e+010 1.46
545e+010 2.05626e+010 1.65174e+010 1.48818e+010 1.34608e+010 1.45183e+010 1.77039e+010 2.12866e+010 1.38146e+010 1.34805e+010
1.51968e+010 1.1694e+010 6.99033e+009 1.01523e+010 1.29248e+010 9.73908e+009 1.01732e+010 8.93874e+009 1.03672e+010 1.31885e+010 1.30847e+010 1.23862e+010 1.33702e+010 9.
0824e+009 1.27441e+010 1.0237e+010 9.22332e+009 8.34259e+009 8.99803e+009 1.09723e+010 1.31928e+010 8.5619e+009 8.35479e+009
1.48695e+010 1.14421e+010 6.83976e+009 9.93361e+009 1.26464e+010 9.52931e+009 9.95411e+009 8.74621e+009 1.01439e+010 1.29045e+010 1.28029e+010 1.21194e+010 1.30822e+010 8
.88677e+009 1.24696e+010 1.00165e+010 9.02466e+009 8.1629e+009 8.80422e+009 1.0736e+010 1.29087e+010 8.37747e+009 8.17484e+009
2.24228e+010 1.72544e+010 1.03142e+010 1.49796e+010 1.90705e+010 1.437e+010 1.50105e+010 1.31891e+010 1.52967e+010 1.94596e+010 1.93064e+010 1.82757e+010 1.97277e+010 1.3
401e+010 1.88039e+010 1.51047e+010 1.3609e+010 1.23094e+010 1.32765e+010 1.61896e+010 1.9466e+010 1.2633e+010 1.23275e+010
2.66753e+010 2.05267e+010 1.22703e+010 1.78205e+010 2.26872e+010 1.70952e+010 1.78573e+010 1.56904e+010 1.81977e+010 2.31501e+010 2.29678e+010 2.17417e+010 2.3469e+010 1.
59425e+010 2.237e+010 1.79692e+010 1.61899e+010 1.46439e+010 1.57944e+010 1.926e+010 2.31577e+010 1.50289e+010 1.46653e+010
2.49534e+010 1.92017e+010 1.14782e+010 1.66702e+010 2.12227e+010 1.59917e+010 1.67046e+010 1.46775e+010 1.70231e+010 2.16557e+010 2.14852e+010 2.03383e+010 2.19541e+010 1
.49134e+010 2.0926e+010 1.68093e+010 1.51448e+010 1.36986e+010 1.47749e+010 1.80167e+010 2.16628e+010 1.40587e+010 1.37187e+010
1.95299e+010 1.50283e+010 8.98348e+009 1.3047e+010 1.661e+010 1.2516e+010 1.30739e+010 1.14875e+010 1.33232e+010 1.6949e+010 1.68155e+010 1.59179e+010 1.71825e+010 1.1672
1e+010 1.63778e+010 1.31559e+010 1.18532e+010 1.07213e+010 1.15636e+010 1.41009e+010 1.69545e+010 1.10031e+010 1.0737e+010
2.09183e+010 1.60967e+010 9.62215e+009 1.39746e+010 1.77909e+010 1.34058e+010 1.40034e+010 1.23041e+010 1.42704e+010 1.8154e+010 1.8011e+010 1.70495e+010 1.84041e+010 1.2
5019e+010 1.75422e+010 1.40912e+010 1.26959e+010 1.14835e+010 1.23857e+010 1.51034e+010 1.81599e+010 1.17854e+010 1.15003e+010
2.45505e+010 1.88917e+010 1.12929e+010 1.64011e+010 2.08801e+010 1.57335e+010 1.64349e+010 1.44406e+010 1.67483e+010 2.13061e+010 2.11384e+010 2.00099e+010 2.15997e+010 1
.46727e+010 2.05882e+010 1.65379e+010 1.49003e+010 1.34775e+010 1.45364e+010 1.77259e+010 2.13131e+010 1.38318e+010 1.34972e+010
3.2512e+010 2.50181e+010 1.49551e+010 2.17198e+010 2.76513e+010 2.08358e+010 2.17646e+010 1.91235e+010 2.21795e+010 2.82155e+010 2.79934e+010 2.64989e+010 2.86042e+010 1.
94309e+010 2.72647e+010 2.1901e+010 1.97323e+010 1.78481e+010 1.92504e+010 2.34742e+010 2.82247e+010 1.83173e+010 1.78742e+010
1.58273e+010 1.21791e+010 7.28034e+009 1.05735e+010 1.3461e+010 1.01431e+010 1.05953e+010 9.3096e+009 1.07973e+010 1.37357e+010 1.36275e+010 1.29001e+010 1.39249e+010 9.4
5922e+009 1.32728e+010 1.06617e+010 9.60598e+009 8.68871e+009 9.37134e+009 1.14276e+010 1.37402e+010 8.91711e+009 8.70142e+009
1.52522e+010 1.17366e+010 7.01582e+009 1.01893e+010 1.29719e+010 9.7746e+009 1.02103e+010 8.97134e+009 1.0405e+010 1.32366e+010 1.31324e+010 1.24313e+010 1.3419e+010 9.11
552e+009 1.27906e+010 1.02743e+010 9.25695e+009 8.37301e+009 9.03084e+009 1.10124e+010 1.32409e+010 8.59311e+009 8.38526e+009
2.50398e+010 1.92682e+010 1.1518e+010 1.67279e+010 2.12962e+010 1.60471e+010 1.67625e+010 1.47284e+010 1.7082e+010 2.17308e+010 2.15597e+010 2.04087e+010 2.20302e+010 1.4
9651e+010 2.09985e+010 1.68675e+010 1.51973e+010 1.37461e+010 1.48261e+010 1.80792e+010 2.17379e+010 1.41075e+010 1.37662e+010
2.42386e+010 1.86516e+010 1.11494e+010 1.61927e+010 2.06147e+010 1.55336e+010 1.62261e+010 1.42571e+010 1.65354e+010 2.10354e+010 2.08698e+010 1.97557e+010 2.13252e+010 1
.44862e+010 2.03266e+010 1.63278e+010 1.4711e+010 1.33062e+010 1.43517e+010 1.75006e+010 2.10423e+010 1.3656e+010 1.33257e+010
1.80261e+010 1.38712e+010 8.29179e+009 1.20424e+010 1.53311e+010 1.15523e+010 1.20673e+010 1.0603e+010 1.22974e+010 1.5644e+010 1.55208e+010 1.46922e+010 1.58595e+010 1.0
7734e+010 1.51168e+010 1.21429e+010 1.09405e+010 9.89582e+009 1.06733e+010 1.30152e+010 1.56491e+010 1.0156e+010 9.91029e+009
2.18265e+010 1.67956e+010 1.00399e+010 1.45813e+010 1.85633e+010 1.39878e+010 1.46114e+010 1.28383e+010 1.489e+010 1.89421e+010 1.8793e+010 1.77897e+010 1.92031e+010 1.30
447e+010 1.83038e+010 1.4703e+010 1.32471e+010 1.19821e+010 1.29235e+010 1.57591e+010 1.89483e+010 1.22971e+010 1.19996e+010
2.60184e+010 2.00213e+010 1.19681e+010 1.73817e+010 2.21285e+010 1.66743e+010 1.74176e+010 1.5304e+010 1.77496e+010 2.25801e+010 2.24023e+010 2.12063e+010 2.28911e+010 1.
555e+010 2.18192e+010 1.75268e+010 1.57912e+010 1.42833e+010 1.54055e+010 1.87857e+010 2.25874e+010 1.46588e+010 1.43042e+010
1.95221e+010 1.50223e+010 8.97988e+009 1.30418e+010 1.66034e+010 1.2511e+010 1.30687e+010 1.14829e+010 1.33179e+010 1.69422e+010 1.68088e+010 1.59115e+010 1.71756e+010 1.
16674e+010 1.63713e+010 1.31506e+010 1.18484e+010 1.0717e+010 1.1559e+010 1.40953e+010 1.69477e+010 1.09987e+010 1.07327e+010
1.26354e+010 9.72299e+009 5.81212e+009 8.44113e+009 1.07463e+010 8.09758e+009 8.45855e+009 7.43214e+009 8.61982e+009 1.09656e+010 1.08793e+010 1.02985e+010 1.11167e+010 7
.55159e+009 1.05961e+010 8.51158e+009 7.66875e+009 6.93647e+009 7.48144e+009 9.12299e+009 1.09692e+010 7.11881e+009 6.94662e+009
2.76735e+010 2.12949e+010 1.27295e+010 1.84874e+010 2.35362e+010 1.7735e+010 1.85256e+010 1.62776e+010 1.88788e+010 2.40165e+010 2.38274e+010 2.25554e+010 2.43473e+010 1.
65392e+010 2.32072e+010 1.86417e+010 1.67958e+010 1.51919e+010 1.63855e+010 1.99808e+010 2.40243e+010 1.55913e+010 1.52142e+010
```

```
Estimated Multiplication count: 108780
Actual Multiplication count: 108780
PS D:\Tejas\clg\daa\Experiment 04\code>
```

Observation:

- The cost table tells the minimum cost for each pair of multiplication possible.
- The actual cost is equal to the minimum cost estimated by matrix chain multiplication.
- The parantehsization table also tells the cost for not only the entire multiplication but also any intermediate multiplication, hence, complying to the optimal substructure property.

CONCLUSION:

After conducting this experiment, i have learnt to use dynamic programming approach to determine the optimal matric chain multiplication paranthesization. I have also learnt how the dynamic programming approach solves the problem and the importance of the optimal substructure property along with using tables to lookup previously calculated values.