**DESIGN AND ANALYSIS OF**

**ALGORITHMS**

**LAB ASSIGNMAENT**

**Chain Matrix Multiplication and Its Time Complexity**

**Submitted BY:**

**Gaurav Yadav**

**11911038**

**CSE**

* **Source Code:**

#include<stdio.h>

#include<stdlib.h>

int minimum\_cost(int matrix[20], int t){

int x, small;

if(t == 1)

return matrix[0];

else{

small = matrix[0];

for(x = 1; x < t; x++){

if(matrix[x] < small)

small = matrix[x];

}

return small;

}

}

int main(){

int t, i, l, j, k, limit,f;

int matrix[30], multiplier[10][15]={0}, columns[15], rows[15], temp[15];

printf("\nEnter Total Number of Matrices:\t");

scanf("%d", &limit);

for(i = 0; i < limit; i++){

printf("\nEnter Number of Rows of Matrix %d:\t", i + 1);

scanf("%d", &rows[i]);

printf("Enter Number of Columns of Matrix %d:\t", i + 1);

scanf("%d", &columns[i]);

}

printf("\n\n\n");

for(i = 0; i < limit; i++)

temp[i]=rows[i];

temp[i] = columns[i - 1];

for(l = 2; l <= limit; l++){

for(j = l, i = 1; j <= limit; j++, i++){

t = 0;

for(k = i; k < j; k++){

matrix[t] = (multiplier[i][k] + multiplier[k + 1][j]) + (temp[i - 1] \* temp[k] \* temp[j]);

t++;

}

multiplier[i][j] = minimum\_cost(matrix, t);

}

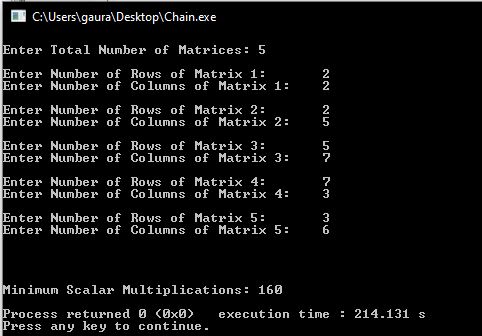
}

printf("\nMinimum Scalar Multiplications:\t%d\n", multiplier[1][limit]);

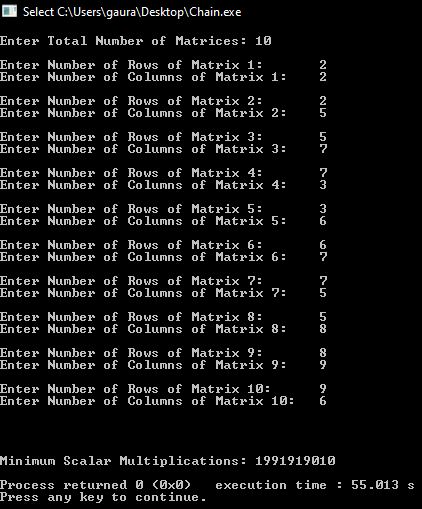
return 0;

}

* **Output:**
* **5:**

****

* **10:**

****

* **15:**

****

* **Time Complexity:**

