31.print the pattern.

1 1 22 333 4444 55555

Program:

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter the number of rows: ";

cin >> n;

// Loop for each row

for (int i = 1; i <= n; ++i) {

// Print the number 'i' 'i' times in each row

for (int j = 1; j <= i; ++j) {

cout << i;

}

// Move to the next line after each row

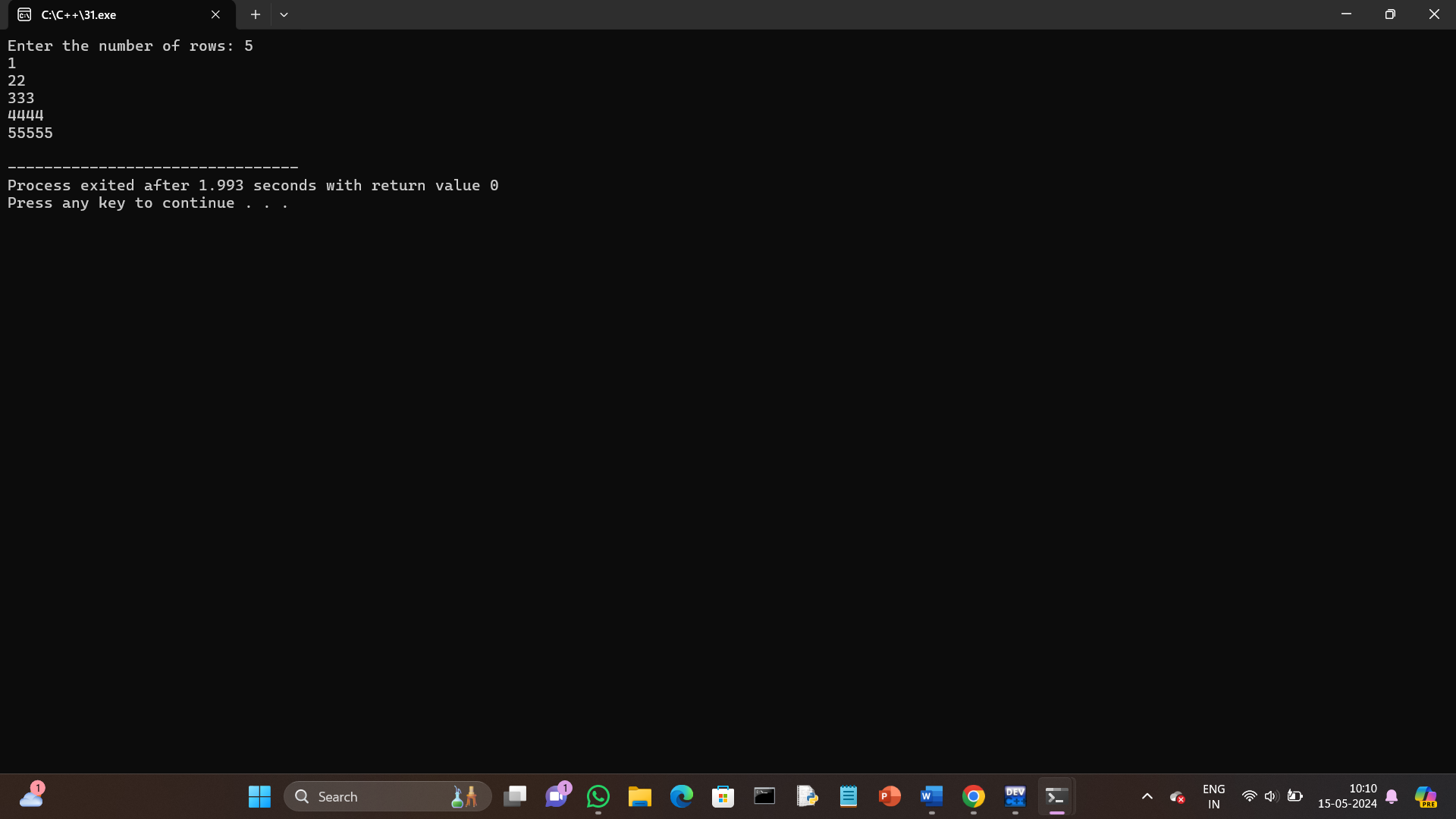
cout << endl;

}

return 0;

}

Output:



32.print the pattern.

\* \*\* \*\*\* \*\*\*\* \*\*\*\*\*

Program:

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter the number of rows: ";

cin >> n;

// Loop for each row

for (int i = 1; i <= n; ++i) {

// Print '\*' i times in each row

for (int j = 1; j <= i; ++j) {

cout << "\*";

}

// Move to the next line after each row

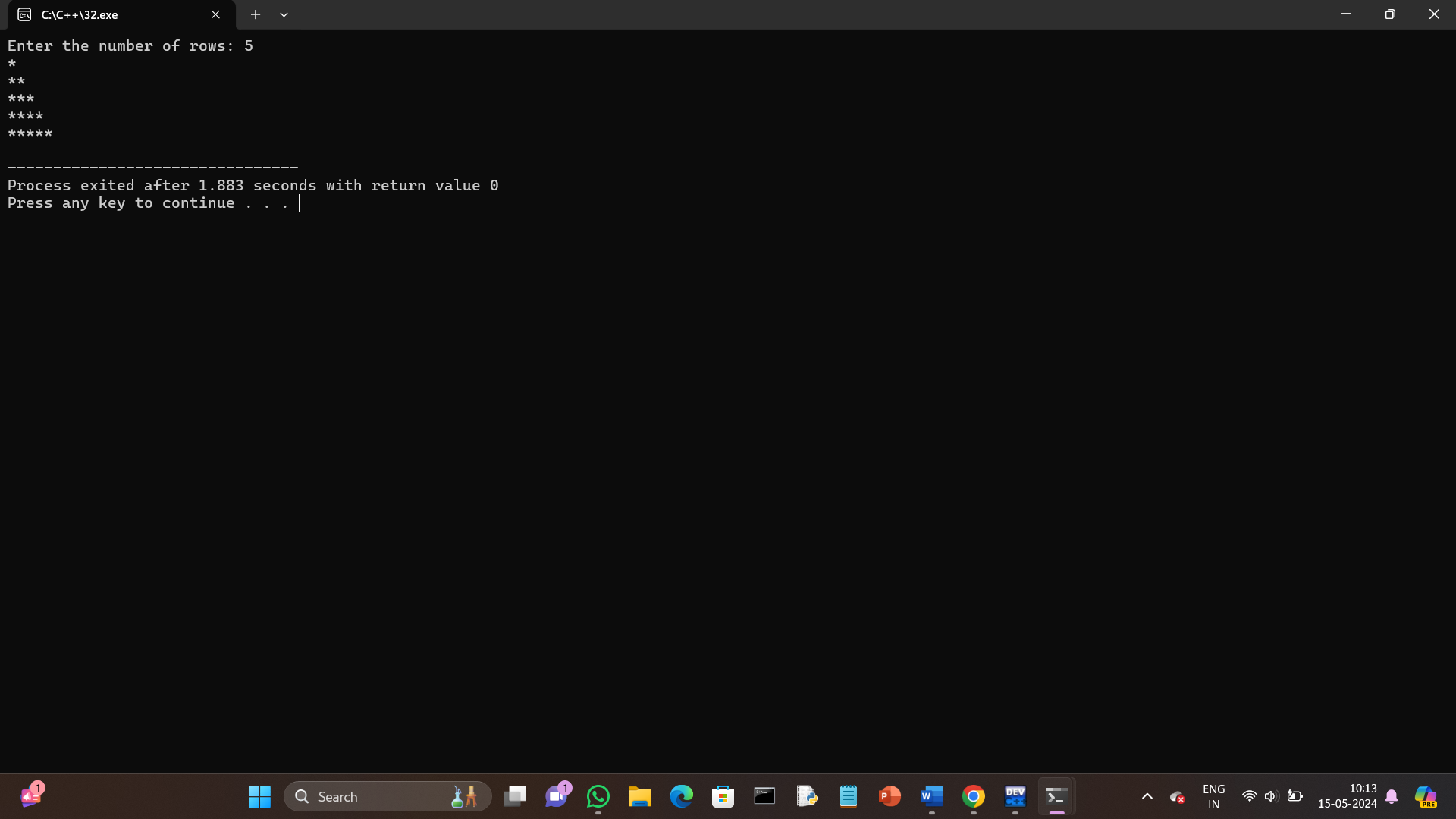
cout << endl;

}

return 0;

}

Output:



33.Print pascal triangle pattern nested for loop.

Program:

#include <iostream>

using namespace std;

// Function to calculate factorial

int factorial(int n) {

if (n <= 1) return 1;

return n \* factorial(n - 1);

}

// Function to calculate combination (n choose r)

int combination(int n, int r) {

return factorial(n) / (factorial(r) \* factorial(n - r));

}

int main() {

int rows;

cout << "Enter the number of rows for Pascal's triangle: ";

cin >> rows;

// Loop through each row

for (int i = 0; i < rows; i++) {

// Print leading spaces for each row

for (int space = 0; space < rows - i - 1; space++) {

cout << " ";

}

// Print values for each row

for (int j = 0; j <= i; j++) {

cout << combination(i, j) << " ";

}

// Move to the next line for the next row

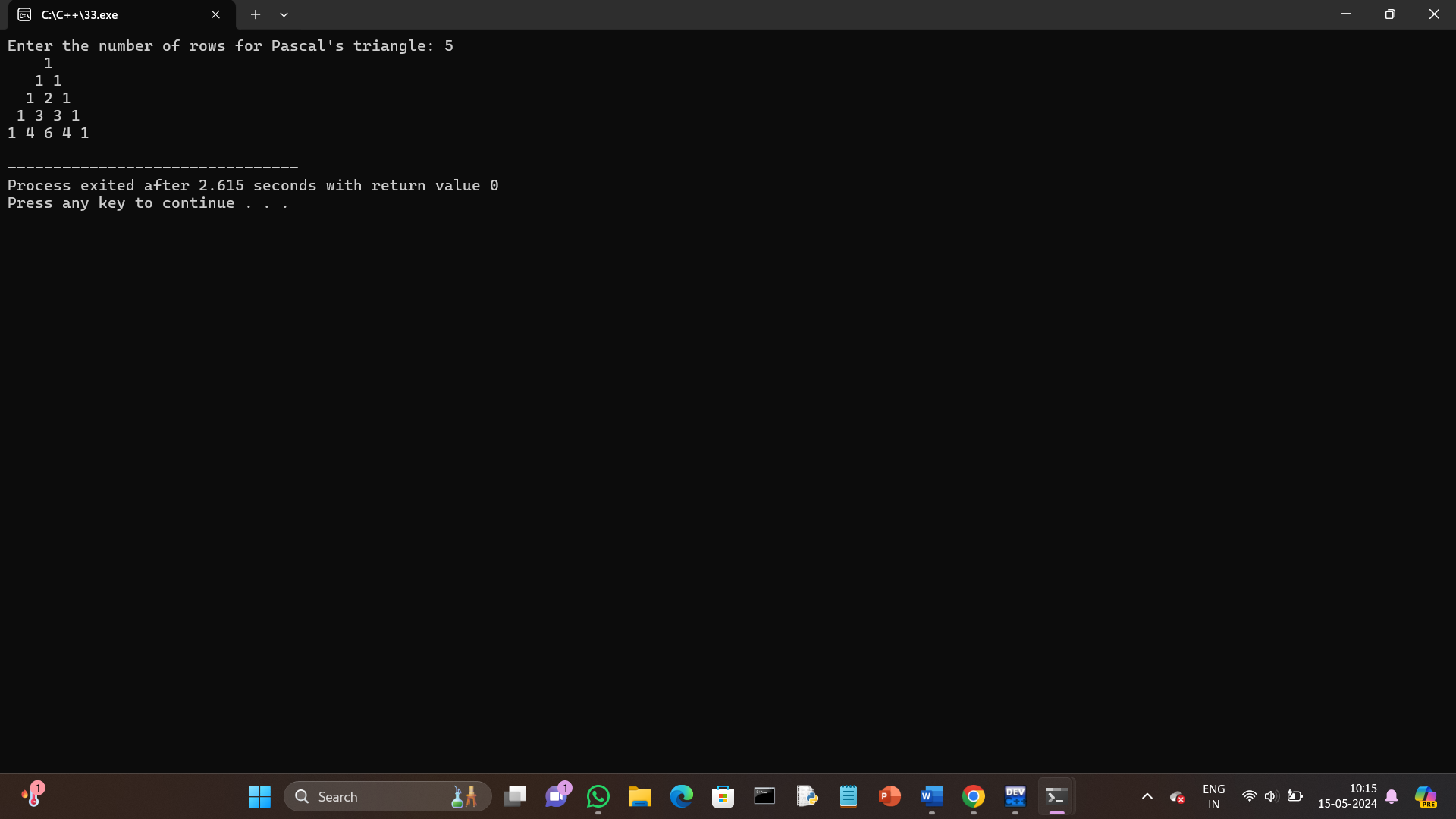
cout << endl;

}

return 0;

}

Output:



34.Print diamond pattern with \* using nested for loop.

Program:

#include <iostream>

using namespace std;

int main() {

int n, i, j, space;

cout << "Enter the number of rows (odd number): ";

cin >> n;

// Upper part of the diamond

for (i = 1; i <= n; i += 2) {

// Print spaces

for (space = 0; space < (n - i) / 2; space++) {

cout << " ";

}

// Print asterisks

for (j = 1; j <= i; j++) {

cout << "\*";

}

// Move to the next line

cout << endl;

}

// Lower part of the diamond

for (i = n - 2; i >= 1; i -= 2) {

// Print spaces

for (space = 0; space < (n - i) / 2; space++) {

cout << " ";

}

// Print asterisks

for (j = 1; j <= i; j++) {

cout << "\*";

}

// Move to the next line

cout << endl;

}

return 0;

}

Output:

