**26. Strong number**

#include <iostream>

using namespace std;

int factorial(int n) {

int fact = 1;

for(int i = 2; i <= n; i++) fact \*= i;

return fact;

}

bool isStrongNumber(int num) {

int sum = 0, originalNum = num;

while(num > 0) {

sum += factorial(num % 10);

num /= 10;

}

return sum == originalNum;

}

int main() {

int number;

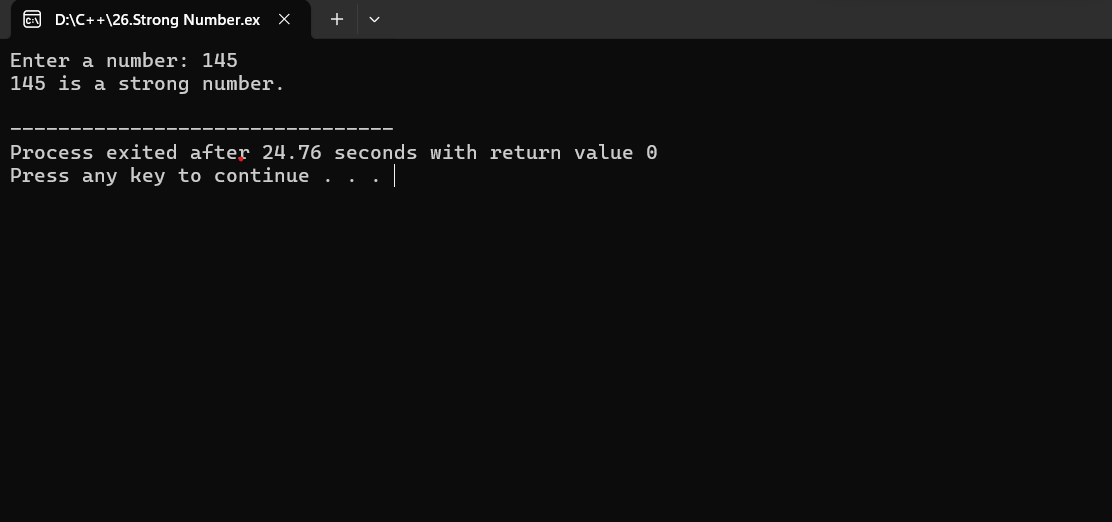
cout << "Enter a number: ";

cin >> number;

cout << number << (isStrongNumber(number) ? " is " : " is not ") << "a strong number." << endl;

return 0;

}



**27. buzz number**

#include <iostream>

using namespace std;

bool isBuzzNumber(int num) {

return (num % 7 == 0) || (num % 10 == 7);

}

int main() {

int number;

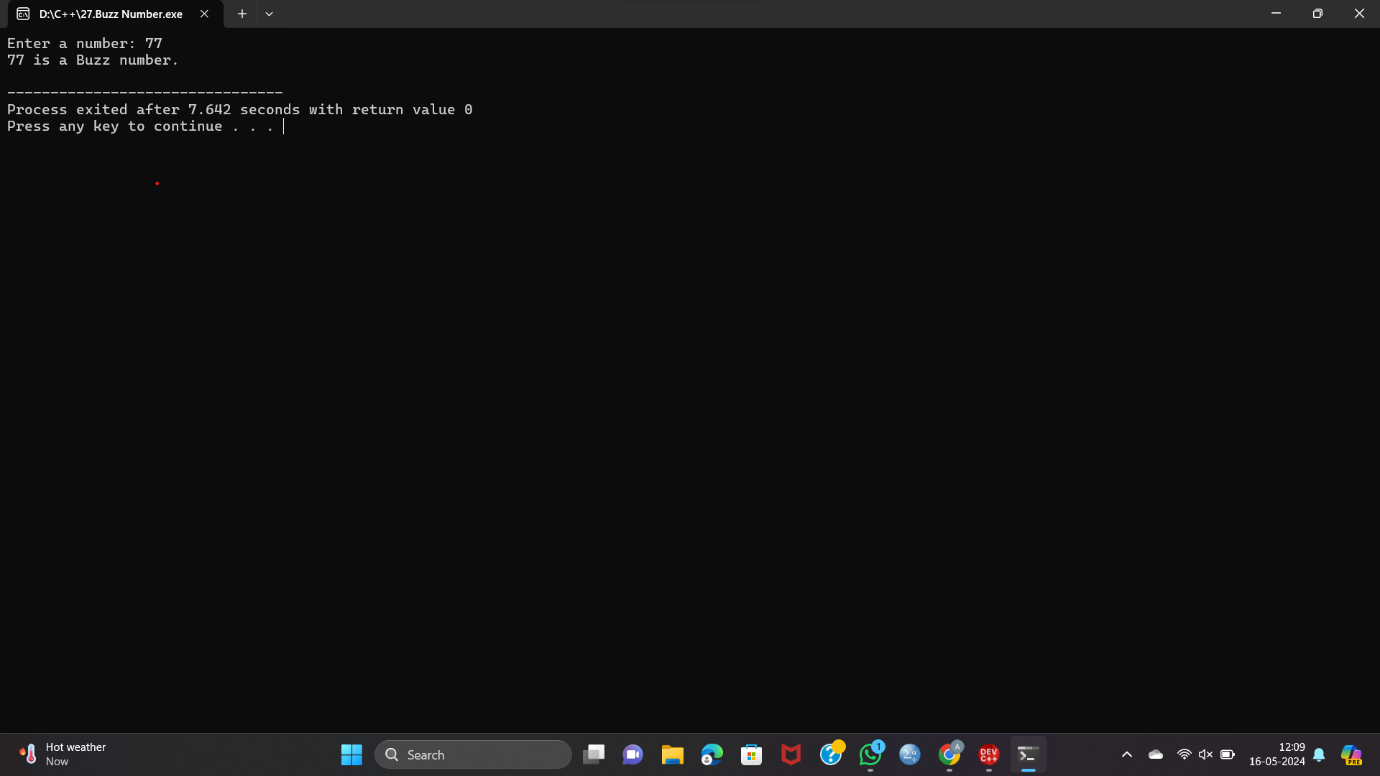
cout << "Enter a number: ";

cin >> number;

cout << number << (isBuzzNumber(number) ? " is " : " is not ") << "a Buzz number." << endl;

return 0;

}



**28. Neon number**

#include <iostream>

using namespace std;

bool isNeonNumber(int num) {

int square = num \* num;

int sum = 0;

while (square > 0) {

sum += square % 10;

square /= 10;

}

return (sum == num);

}

int main() {

int number;

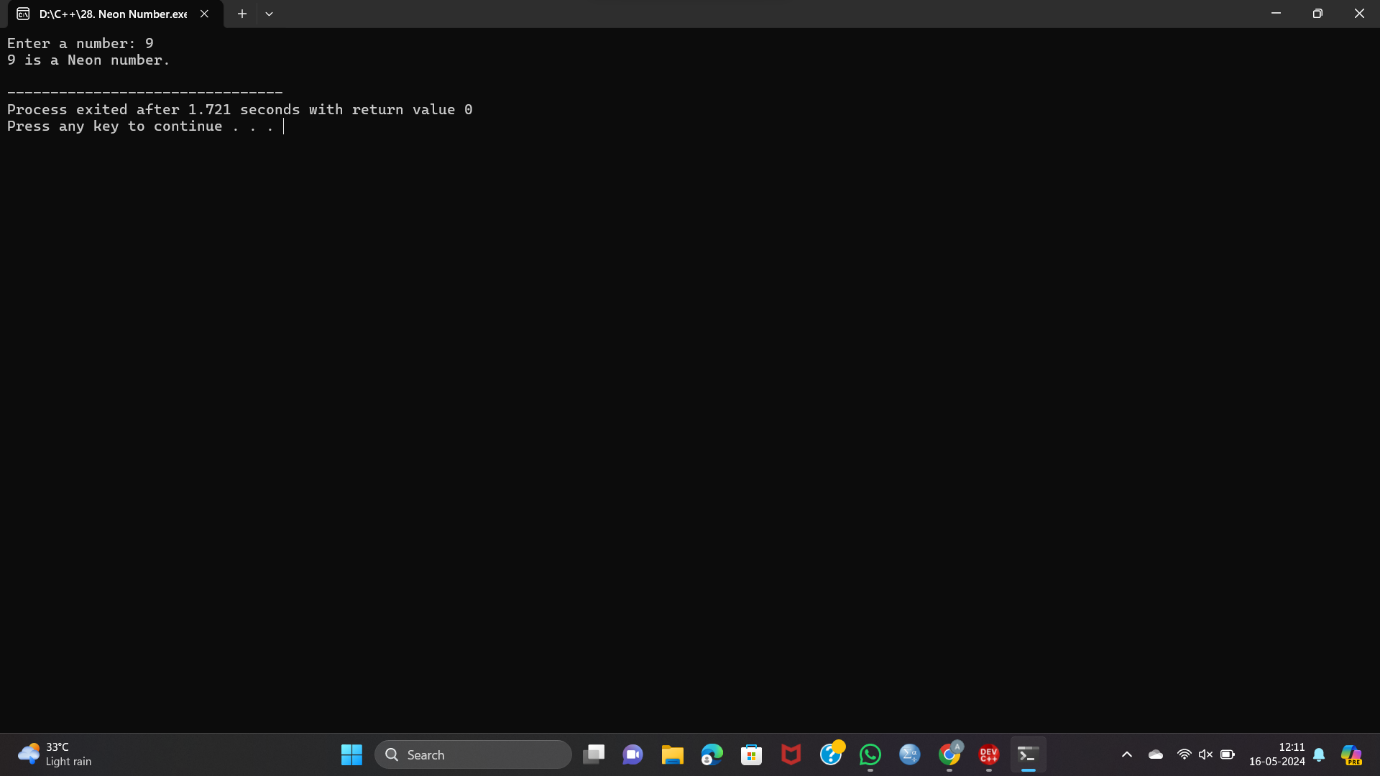
cout << "Enter a number: ";

cin >> number;

cout << number << (isNeonNumber(number) ? " is " : " is not ") << "a Neon number." << endl;

return 0;

}



**29. Abundant number**

#include <iostream>

using namespace std;

bool isAbundantNumber(int num) {

int sum = 0;

for (int i = 1; i <= num / 2; ++i) {

if (num % i == 0) {

sum += i;

}

}

return sum > num;

}

int main() {

int number;

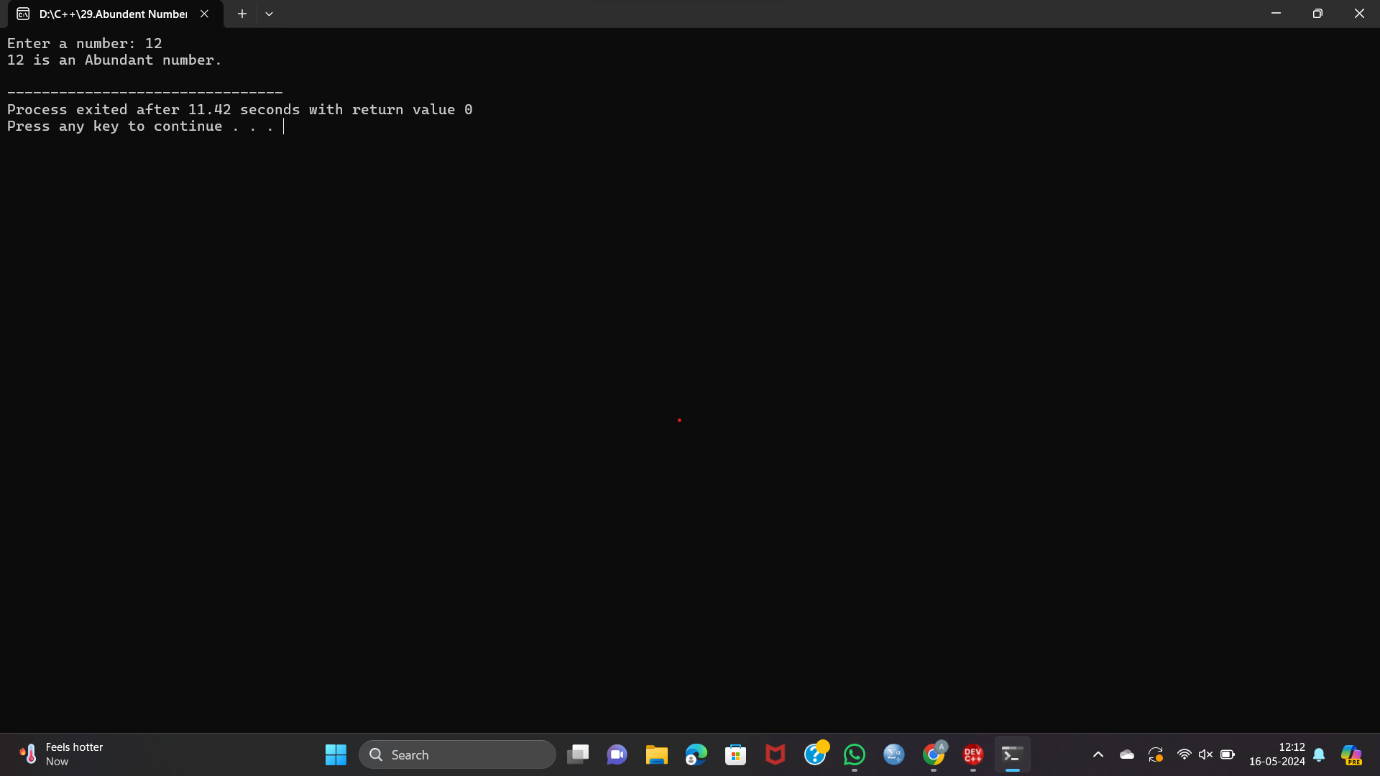
cout << "Enter a number: ";

cin >> number;

cout << number << (isAbundantNumber(number) ? " is " : " is not ") << "an Abundant number." << endl;

return 0;

}



**30. Narcissistic number**

#include <iostream>

#include <cmath>

using namespace std;

bool isNarcissisticNumber(int num) {

int originalNum = num, sum = 0, n = 0;

int temp = num;

while (temp > 0) {

temp /= 10;

n++;

}

temp = num;

while (temp > 0) {

int digit = temp % 10;

sum += pow(digit, n);

temp /= 10;

}

return (sum == originalNum);

}

int main() {

int number;

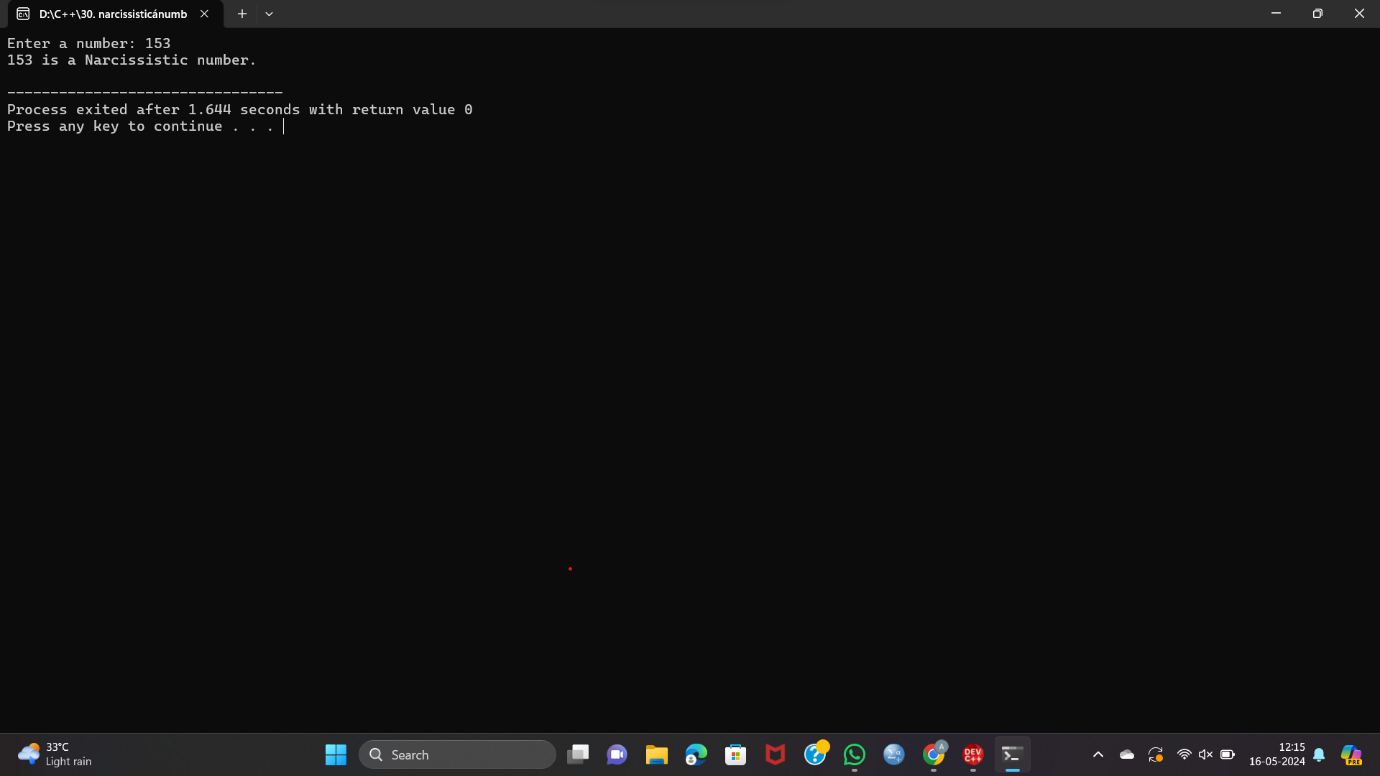
cout << "Enter a number: ";

cin >> number;

cout << number << (isNarcissisticNumber(number) ? " is " : " is not ") << "a Narcissistic number." << endl;

return 0;

}



**31. print the pattern 1 22 333 4444 55555**

#include <iostream>

using namespace std;

int main() {

int n = 5;

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

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

cout << i;

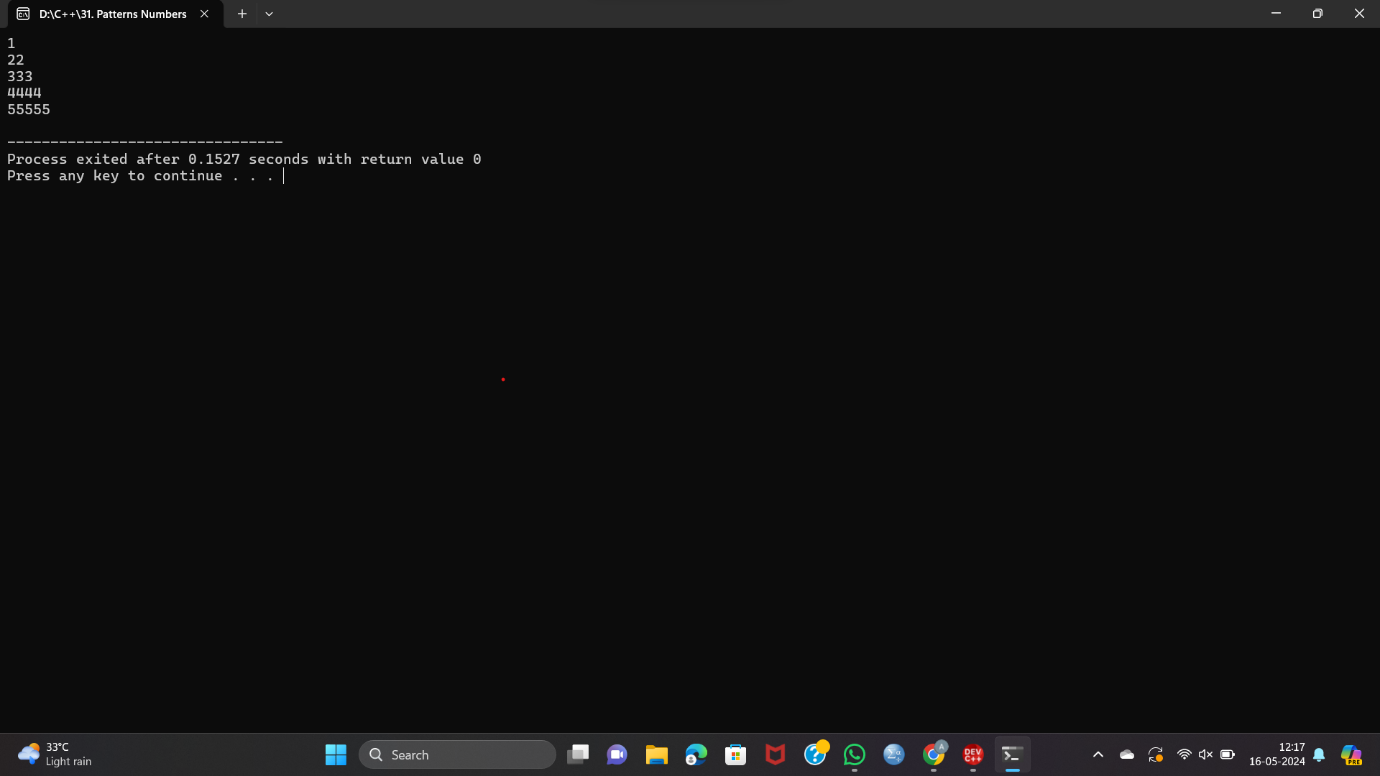
}

cout << endl;

}

return 0;

}



**32. print the pattern \* \*\* \*\*\* \*\*\*\* \*\*\*\*\***

#include <iostream>

using namespace std;

int main() {

int n = 5;

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

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

cout << '\*';

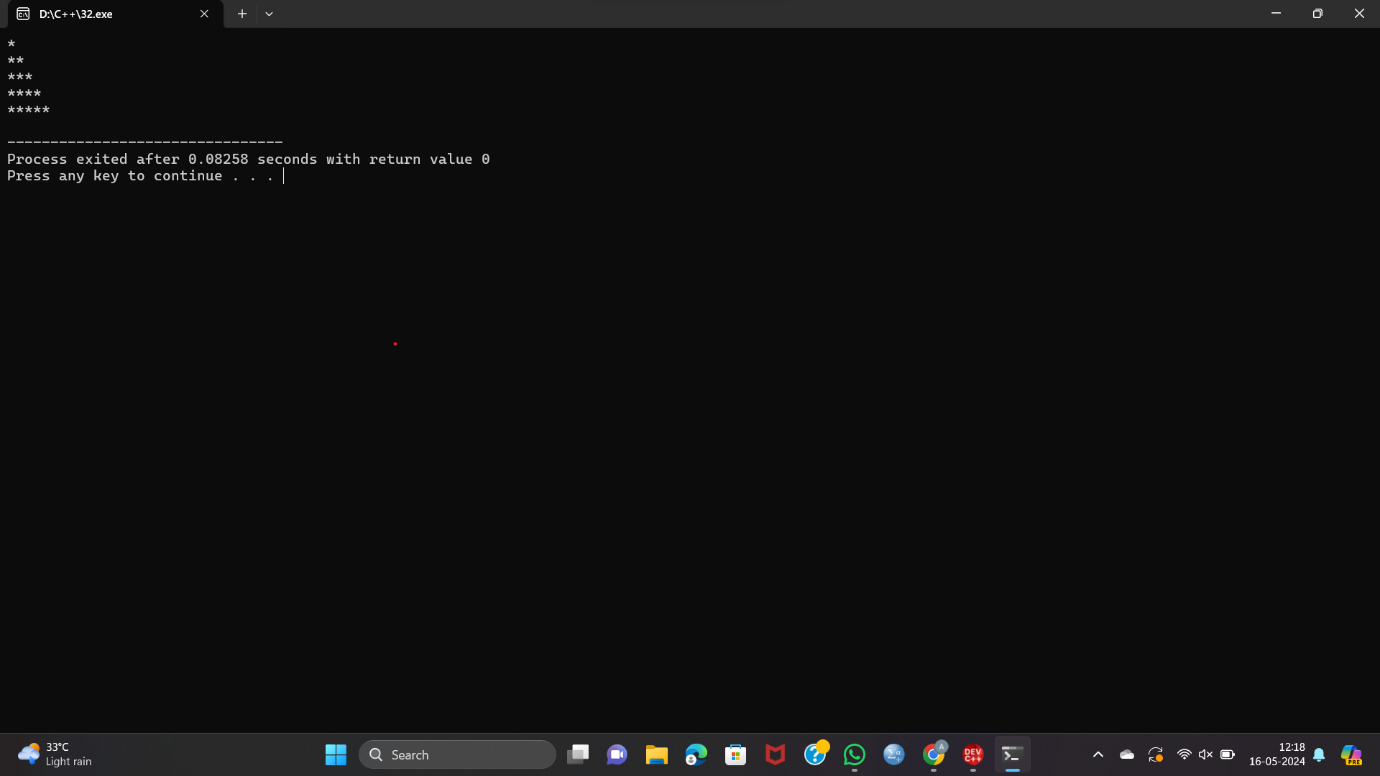
}

cout << endl;

}

return 0;

}



**33. Print pascal triangle pattern nested for loop**

#include <iostream>

using namespace std;

// Function to calculate factorial of a number

int factorial(int n) {

int fact = 1;

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

fact \*= i;

}

return fact;

}

int binomialCoefficient(int n, int k) {

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

}

int main() {

int rows;

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

cin >> rows;

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

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

cout << " ";

}

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

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

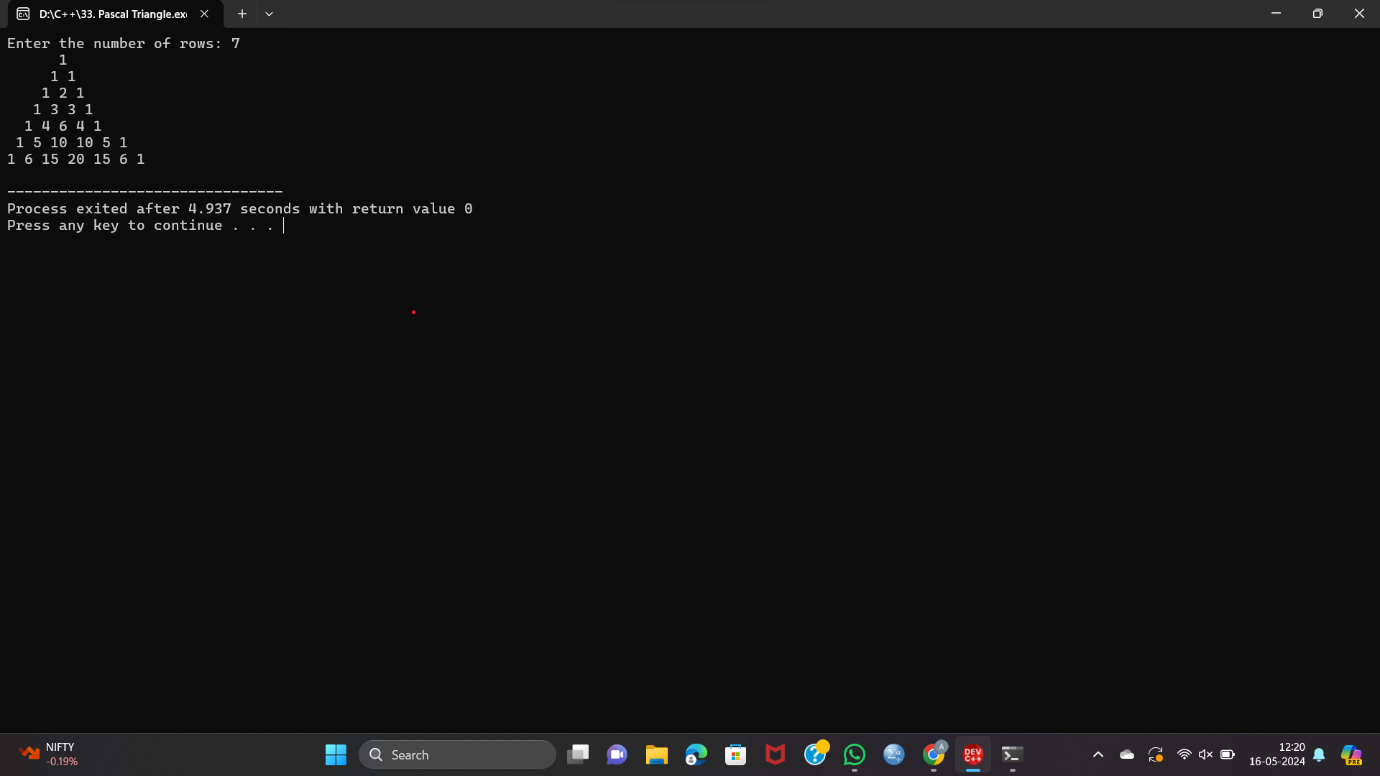
}

cout << endl;

}

return 0;

}



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

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter the number of rows for the upper half (including middle row): ";

cin >> n;

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

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

cout << " ";

}

for (int j = 1; j <= (2 \* i - 1); j++) {

cout << "\*";

}

cout << endl;

}

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

for (int j = n; j > i; j--) {

cout << " ";

}

for (int j = 1; j <= (2 \* i - 1); j++) {

cout << "\*";

}

cout << endl;

}

return 0;

}

