Shri Vile Parle Kelavani Mandal's

INSTITUTE OF TECHNOLOGY

DHULE (M.S.)

DEPARMENT OF COMPUTER ENGINEERING

Subject : Competitive Programming Lab

Name: Mohammed Meraj Mohammed Ashfaque **Roll No. :** 32

Class : TY. Comp. Engg. **Batch : T2** **Division: T**

Expt. No. :12 **Date : 24/03/2025**

Title : Pascal's Triangle

Remark

Signature

Language: C++

```
//  Pascal's Triangle Problem by Meraj 32 T2
#include<iostream>
using namespace std;

int main()
{
   cout << "========== PASCAL'S TRIANGLE OF DEATH ==========" << endl;
   cout << "\nINPUT: No input required for this program\n" << endl;
   cout << "OUTPUT:" << endl;

   // Define the limit (10^60 is very large, so we'll use a smaller practical limit)
   // For demonstration, using 10^6 instead of 10^60 to see the program work
   long long LIMIT = 1000000; // 10^6 (you can change this to test)

   // Maximum possible rows (assuming we won't exceed 100 rows)
   int MAX_ROWS = 100;

   // Current row and next row arrays
   long long current_row[MAX_ROWS];
   long long next_row[MAX_ROWS];

   // Initialize first row
   current_row[0] = 1;
```

```
int current_size = 1;

// Main loop to generate Pascal's Triangle
while(true)
{
   // Print current row
   for(int i = 0; i < current_size; i++)
   {
      cout << current_row[i];
      if(i < current_size - 1)  // Add space between numbers, but not after last number
      {
         cout << " ";
      }
   }
   cout << endl;

   // Generate next row
   next_row[0] = 1;  // First element is always 1
   int next_size = current_size + 1;

   // Calculate middle elements
   for(int i = 1; i < current_size; i++)
   {
      next_row[i] = current_row[i-1] + current_row[i];
   }

   next_row[current_size] = 1;  // Last element is always 1

   // Check if any number in next row exceeds limit
   bool exceeds_limit = false;
   for(int i = 0; i < next_size; i++)
   {
      if(next_row[i] >= LIMIT)
      {
         exceeds_limit = true;
         break;
      }
   }

   // If limit exceeded, print the row and exit
   if(exceeds_limit)
   {
      // Print the final row
```

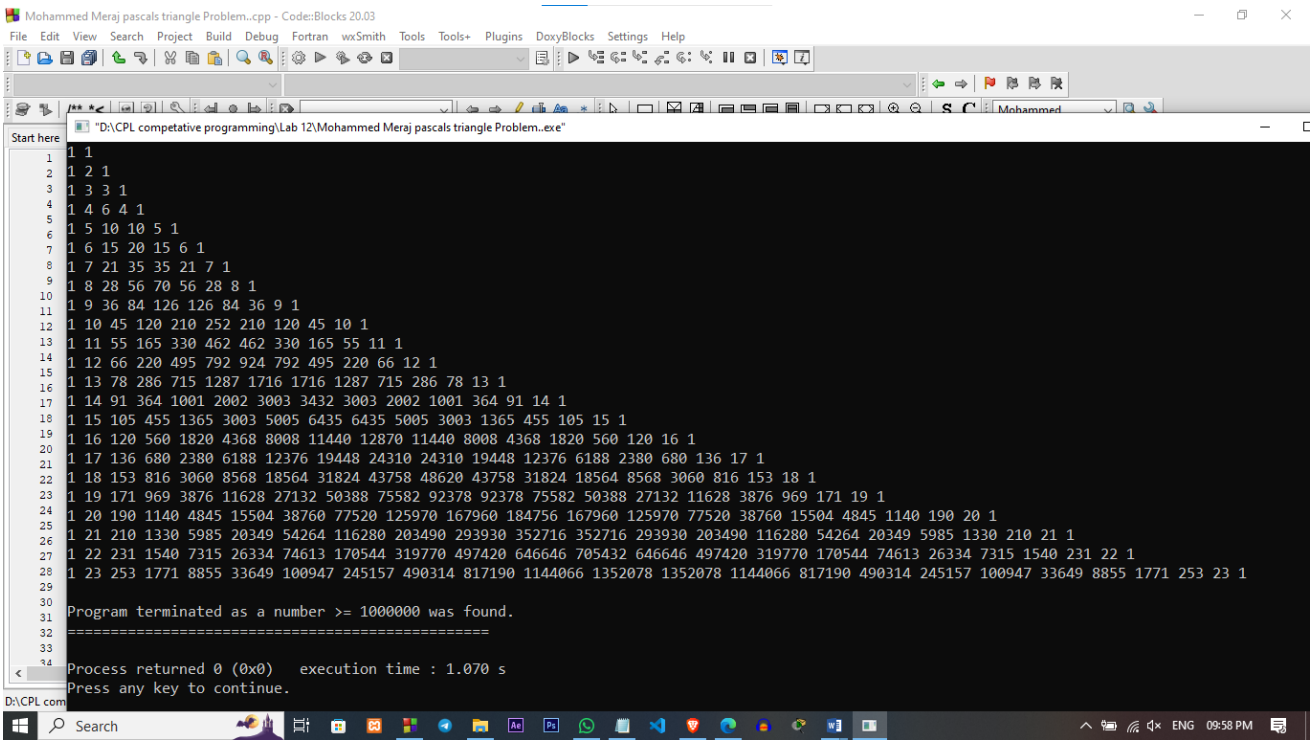
```
            for(int i = 0; i < next_size; i++)
            {
                cout << next_row[i];
                if(i < next_size - 1)
                {
                    cout << " ";
                }
            }
            cout << endl;
            break;  // Exit the loop
        }

        // Copy next_row to current_row for next iteration
        for(int i = 0; i < next_size; i++)
        {
            current_row[i] = next_row[i];
        }
        current_size = next_size;
    }

    cout << "\nProgram terminated as a number >= " << LIMIT << " was found." << endl;
    cout << "================================================" << endl;

    return 0;
}
```

Output :


Mohammed Meraj pascals triangle Problem..cpp - Code::Blocks 20.03
"D:\CPL competative programming\Lab 12\Mohammed Meraj pascals triangle Problem..exe"
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
1 12 66 220 495 792 924 792 495 220 66 12 1
1 13 78 286 715 1287 1716 1716 1287 715 286 78 13 1
1 14 91 364 1001 2002 3003 3432 3003 2002 1001 364 91 14 1
1 15 105 455 1365 3003 5005 6435 6435 5005 3003 1365 455 105 15 1
1 16 120 560 1820 4368 8008 11440 12870 11440 8008 4368 1820 560 120 16 1
1 17 136 680 2380 6188 12376 19448 24310 24310 19448 12376 6188 2380 680 136 17 1
1 18 153 816 3060 8568 18564 31824 43758 48620 43758 31824 18564 8568 3060 816 153 18 1
1 19 171 969 3876 11628 27132 50388 75582 92378 92378 75582 50388 27132 11628 3876 969 171 19 1
1 20 190 1140 4845 15504 38760 77520 125970 167960 184756 167960 125970 77520 38760 15504 4845 1140 190 20 1
1 21 210 1330 5985 20349 54264 116280 203490 293930 352716 352716 293930 203490 116280 54264 20349 5985 1330 210 21 1
1 22 231 1540 7315 26334 74613 170544 319770 497420 646646 705432 646646 497420 319770 170544 74613 26334 7315 1540 231 22 1
1 23 253 1771 8855 33649 100947 245157 490314 817190 1144066 1352078 1352078 1144066 817190 490314 245157 100947 33649 8855 1771 253 23 1
Program terminated as a number >= 1000000 was found.
Process returned 0 (0x0) execution time : 1.070 s
Press any key to continue.
Search
ENG
09:58 PM