



Shri Vile Parle Kelavani Mandal's

INSTITUTE OF TECHNOLOGY

DHULE (M.S.)

DEPARMENT OF COMPUTER ENGINEERING

Subject: Competitive Programming Lab (BTCOL606)

Name : Mohammad Anas Aarif Baig Mirza

Roll No. : 31

Class : T.Y Comp

Batch : T2

Division: T

Expt. No. :12

Date :

Remark

Signature

**Title : Problem 12: Write a Program to implement Combinatorics
Pascal's Triangle of Death**

Code:

```
// MOHAMMAD_ANAS_31_TY_COMP
```

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
int main() {
```

```
    const unsigned long long LIMIT = 1e60; // 10^60
```

```
    vector<unsigned long long> row = {1}; // Initialize first row
```

```
    int line_count = 0;
```

```
    while (line_count < 7) {
```

```
        // Print current row
```

```
        for (size_t i = 0; i < row.size(); ++i) {
```

```
            cout << row[i];
```

```
            if (i < row.size() - 1) cout << " ";
```

```
        }
```

```

cout << endl;
line_count++;
// Generate next row
vector<unsigned long long> next_row = {1}; // Start with 1
bool exceed_limit = false;
// Calculate middle elements
for (size_t i = 1; i < row.size(); ++i) {
    unsigned long long next_val = row[i-1] + row[i];
    if (next_val >= LIMIT) {
        exceed_limit = true;
    }
    next_row.push_back(next_val);
}
next_row.push_back(1); // End with 1
// Check if limit exceeded
if (exceed_limit) {
    // Print the row that caused exceed
    for (size_t i = 0; i < next_row.size(); ++i) {
        cout << next_row[i];
        if (i < next_row.size() - 1) cout << " ";
    }
    cout << endl;
    break;
}
// Update row for next iteration

```

```

row = next_row;

}

return 0;

```

}Output:

The screenshot shows a C++ IDE with two windows. The left window displays the source code for a program that generates Pascal's triangle. The right window shows the output of the program, which is a 6x6 Pascal's triangle. The code includes comments and uses a limit of 10 for the sum of elements.

```

22 bool exceed_limit = false;
23
24 // Calculate middle elements
25 for (size_t i = 1; i < row.size(); ++i) {
26     unsigned long long next_val = row[i-1] + row[i];
27     if (next_val >= LIMIT) {
28         exceed_limit = true;
29     }
30     next_row.push_back(next_val);
31 }
32 next_row.push_back(1); // End with 1
33
34 // Check if limit exceeded
35 if (exceed_limit) {
36     // Print the row that caused exceed
37     for (size_t i = 0; i < next_row.size(); ++i) {
38         cout << next_row[i];
39         if (i < next_row.size() - 1) cout << " ";
40     }
41     cout << endl;
42     break;
43 }
44
45 // Update row for next iteration
46 row = next_row;
47 }
48
49 return 0;
50 }
51

```

```

1
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

Process returned 0 (0x0)   execution time : 0.098 s
Press any key to continue.

```