



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## Experiment -10

**Student Name:**Harinder Singh

**UID:** 22BCS10280

**Branch:** BE-CSE

**Section/Group:** 605-A

**Semester:** 5th

**Date of Performance:**

**Subject Name:** Advanced Programming Lab

**Subject Code:** 22CSP-351

### **Ques 1: Pascals Triangle**

Given an integer , return the first numRows of In each number is the sum of the two numbers directly above it as shown:

#### **Code:**

```
class Solution {
public:
    vector<vector<int>> generate(int numRows) {
        vector<vector<int>> res;
        for(int i =0;i<numRows;i++){
            vector<int> a(i+1);
            res.push_back(a);
        }
        for(int i=0;i<numRows;i++){
            for(int j=0;j<=i;j++){
                if(j==0 || i==j){
                    res[i][j]=1;
                }
                else{
                    res[i][j]=res[i-1][j]+res[i-1][j-1];
                }
            }
        }
        return res;
    }
};
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

OUTPUT:

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1

Case 2

Input

numRows =  
5

Output

[[1], [1,1], [1,2,1], [1,3,3,1], [1,4,6,4,1]]

Expected

[[1], [1,1], [1,2,1], [1,3,3,1], [1,4,6,4,1]]

Description

Accepted

Editorial

Solutions

All Submissions

Accepted 30 / 30 testcases passed

Parshant Var... submitted at Apr 23, 2025 14:42

Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

9.66 MB | Beats 64.95%

100%  
50%  
0%

Code

Submit Ctrl Enter

C++ Auto

1 class Solution {  
2 public:  
3 ...vector<vector<int>> generate(int numRows) {  
4 ...vector<vector<int>> res;  
5  
6 }  
7 }

Saved

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1

Case 2

Input

numRows =  
5

Output

[[1], [1,1], [1,2,1], [1,3,3,1], [1,4,6,4,1]]

Expected



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## Ques 2 :Number of 1 bits

Given a positive integer n, write a function that returns the number of set bits in its binary representation (also known as the Hamming weight).).

CODE:

```
class Solution {
public:
    int hammingWeight(int n) {
        int ans=0;
        while(n>0){
            ans+=(n&1);
            n=n>>1;
        }
        return ans;
    }
};
```

☒ Testcase ☒ Test Result

Accepted Runtime: 0 ms

• Case 1

• Case 2

• Case 3

Input

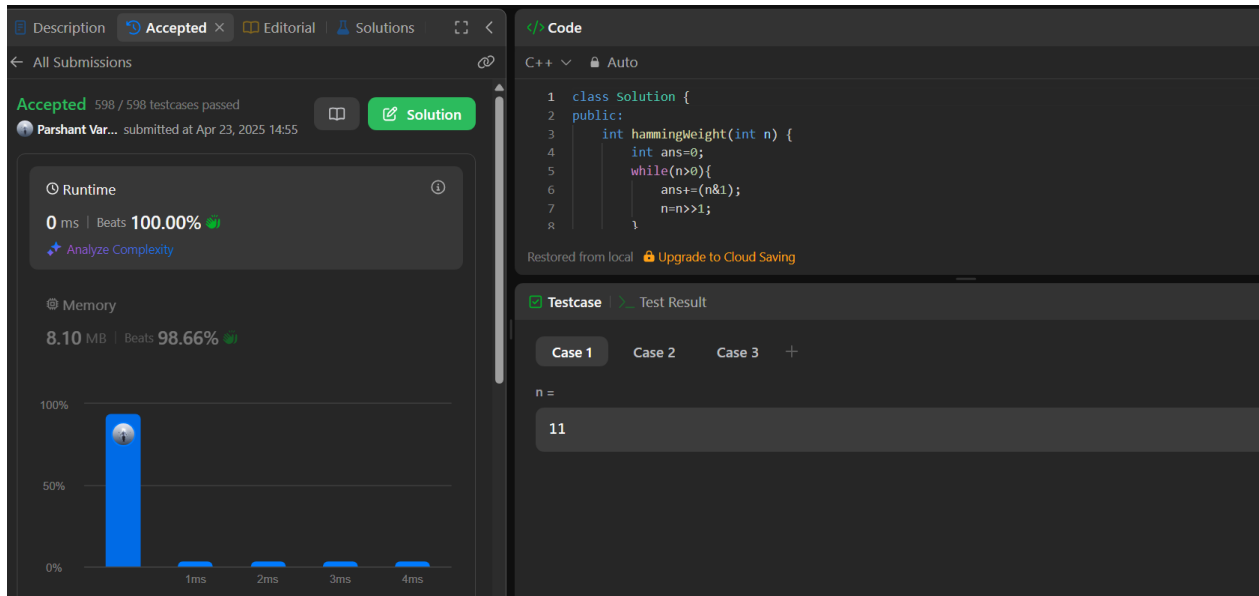
n =  
11

Output

3



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



## Ques3: Divide Two Integers

Given two integers `dividend` and `divisor`, divide two integers using multiplication, division, and mod operator.

The integer division should truncate toward zero, which means losing its fractional part. For example, `4/3` would be truncated to `1`, and `-4/3` would be truncated to `-1`.

Return the quotient after dividing `dividend` by `divisor`.

CODE:

```
class Solution {
public:
    int divide(int dividend, int divisor) {
        long long res=(long long)dividend/(long long)divisor;
        if(dividend==2147483648 && divisor ==-1) return 2147483647;
        return res;
    }
};
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

☒ Testcase | ☒ Test Result

**Accepted** Runtime: 0 ms

- Case 1
- Case 2

Input

dividend =  
10

divisor =  
3

Output  
3

Description | **Accepted** × | Editorial | Solutions | Submissions

← All Submissions

**Accepted** 994 / 994 testcases passed

Parshant Var... submitted at Apr 23, 2025 14:58

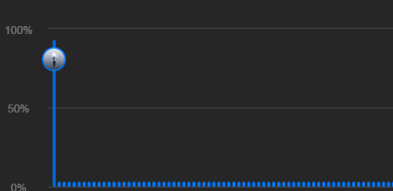
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

8.68 MB | Beats 38.46%



C++ | Auto

1 class Solution {  
2 public:  
3 int divide(int dividend, int divisor) {  
4 long long res=(long long)dividend/(long long)divisor;  
5 if(dividend==2147483648 && divisor ==-1) return 2147483647;  
}

Saved

☒ Testcase | ☒ Test Result

**Accepted** Runtime: 0 ms

- Case 1
- Case 2

Input

dividend =  
10

divisor =  
3

Output