

```

    putchar(x % 10 + '0');
}
string __int128toString(__int128 num)
{
    auto tenPow18 = 10000000000000000000;
    string str;
    do
    {
        long long digits = num % tenPow18;
        auto digitsStr = to_string(digits);
        auto leading0s = (digits != num) ? string(18 -
digitsStr.length(), '0') : "";
        str = leading0s + digitsStr + str;
        num = (num - digits) / tenPow18;
    } while (num != 0);
    return str;
}
bool cmp(__int128 x, __int128 y) { return x > y; }

// To find the rectangular grid sum in a range
with complexity O(1)

class NumMatrix {
private:
    vector<vector<ll>> prefixSum;

public:
    NumMatrix(vector<vector<int>> &matrix) {
        int m = matrix.size();
        int n = matrix[0].size();

        prefixSum = vector<vector<ll>>(m + 1,
vector<ll>(n + 1, 0));

        for (int i = 1; i <= m; i++) {
            for (int j = 1; j <= n; j++) {
                prefixSum[i][j] = matrix[i - 1][j - 1] +
prefixSum[i - 1][j] + prefixSum[i][j - 1] -
prefixSum[i - 1][j - 1];
            }
        }
    }

    ll sumRegion(int row1, int col1, int row2, int
col2) {
        return prefixSum[row2 + 1][col2 + 1] -
prefixSum[row1][col2 + 1] - prefixSum[row2 +
1][col1] + prefixSum[row1][col1];
    }
};

```