Assignment 5 –

#include<bits/stdc++.h>

using namespace std;

using namespace std::chrono;

bool isSafe(int row , int col , vector<vector<char>> &v , int n){

int i = row ;

int j = col ;

// check row

while(j>=0){

if(v[i][j] == 'Q')

return false ;

j--;

}

// check upper left diagonal

i = row ;

j = col ;

while(i>=0 && j>=0){

if(v[i][j] == 'Q')

return false ;

i--;

j--;

}

// Check Lower right Diagonal

i = row ;

j = col ;

while(i<n && j>=0){

if(v[i][j] == 'Q')

return false ;

i++;

j--;

}

return true ;

}

void printOut(vector<vector<char>> &v , int n , int& cnt){

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

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

cout << v[i][j] << " ";

}

cout << endl;

}

cout << endl;

}

void solve(vector<vector<char>> &v , int col , int n , int & cnt){

if(col >= n){

cnt++;

printOut(v, n , cnt);

return ;

}

for(int row = 0 ; row < n ; row++){

if(isSafe(row , col , v , n)){

v[row][col] = 'Q';

solve(v, col+1 , n , cnt);

v[row][col] = '-';

}

}

}

int main () {

for(int i = 4 ; i < 9 ; i++){

vector<vector<char>>v(i , vector<char>(i , '-'));

int cnt = 0 ;

int col = 0 ;

auto start\_time = high\_resolution\_clock::now();

solve(v , col , i , cnt);

auto end\_time = high\_resolution\_clock::now();

auto duration = duration\_cast<nanoseconds>(end\_time - start\_time);

cout << "Time taken: " << duration.count() << " nanoseconds" << endl;

cout << cnt << endl;

}

return 0 ;

}