Contents

[5.1: Tìm và sửa các lỗi cú pháp 2](#_Toc73999209)

[5.2: 3](#_Toc73999210)

[5.3: Check ngoặc đúng 5](#_Toc73999211)

[5.4: Bài toán người du lịch 8](#_Toc73999212)

[5.5: Năm nhuận 11](#_Toc73999213)

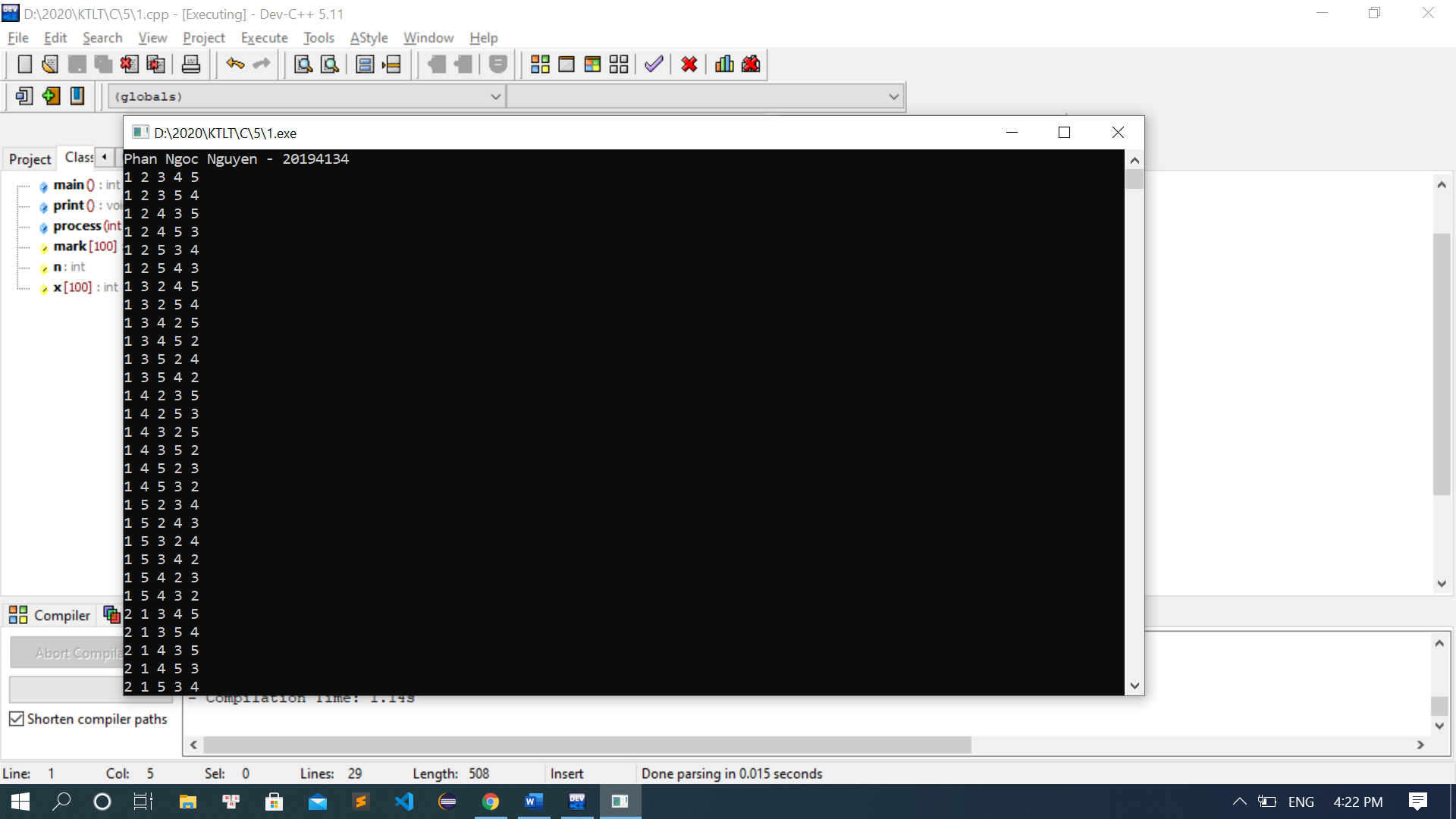
[5.6: Tổng kết 13](#_Toc73999214)

[5.7: Chia tiền 15](#_Toc73999215)

[5.8: Cắt hình chữ nhật 17](#_Toc73999216)

[5.9: Xây tháp 20](#_Toc73999217)

## 5.1: Tìm và sửa các lỗi cú pháp



Code:

#include <stdio.h>

int x[100], mark[100], n;

void print(){

for (int i = 1; i <= n; ++i) printf("%d ", x[i]);

printf("\n");

}

void process(int i) {

if (i > n){

print();

return;

}

for (int j = 1; j <= n; ++j)

if (!mark[j]){

mark[j] = 1;

x[i] = j;

process(i+1);

mark[j] = 0;

}

}

int main() {

printf("Phan Ngoc Nguyen - 20194134\n");

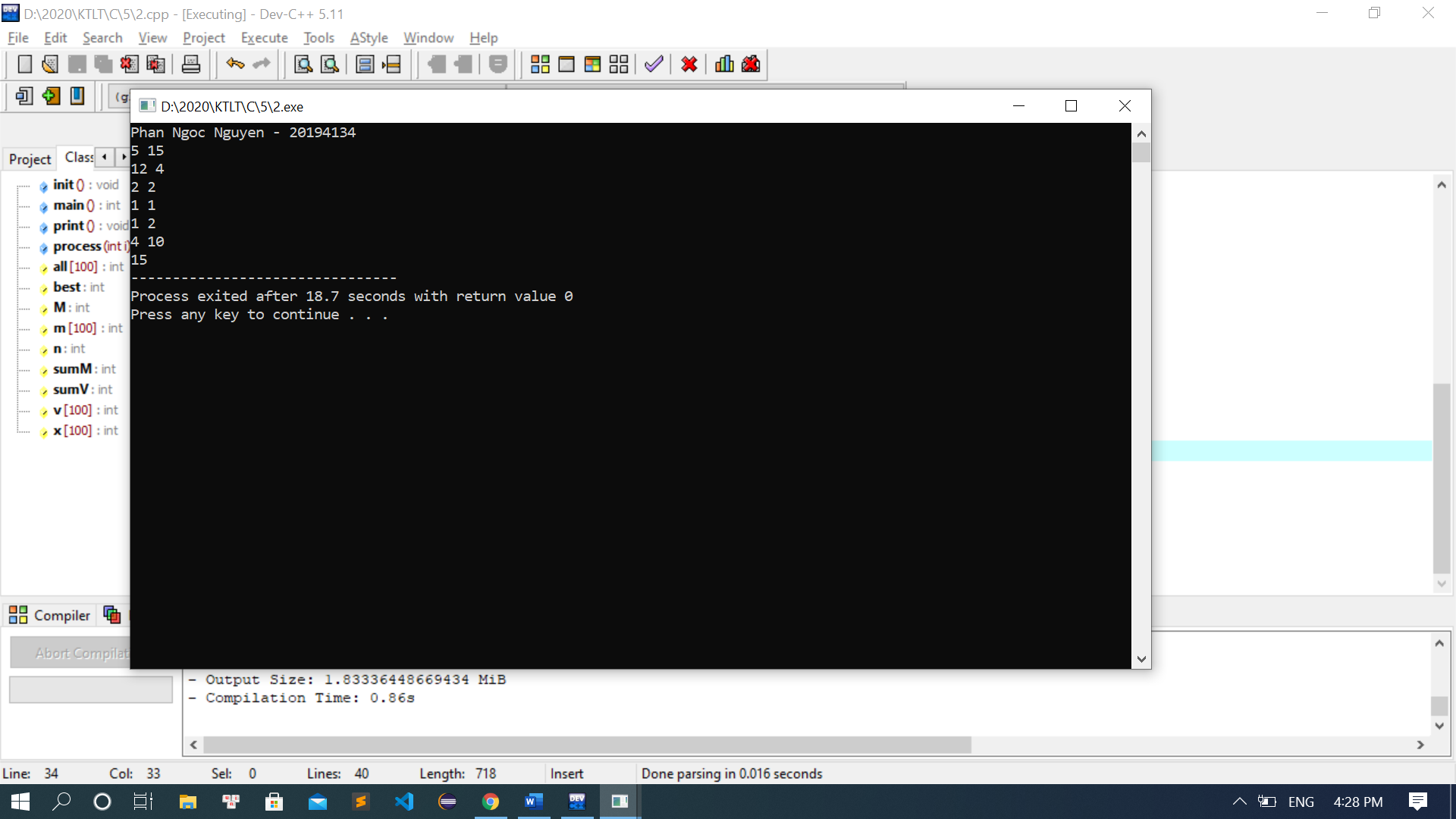
n = 5;

process(1);

return 0;

}

## 5.2:



Code:

#include <iostream>

using namespace std;

int n, M, m[100], v[100];

int x[100], best, sumV, sumM, all[100];

void init(){

for (int i = n; i >= 1; --i){

all[i] = all[i+1] + v[i];

}

}

void print() {

cout << best;

}

void process(int i){

if (sumV + all[i] <= best || sumM > M) return ;

if (i > n){

best = sumV;

return ;

}

process(i+1);

sumM += m[i];

sumV += v[i];

process(i+1);

sumM -= m[i];

sumV -= v[i];

}

int main() {

cout << "Phan Ngoc Nguyen - 20194134" << endl;

cin >> n >> M;

for (int i = 1; i <= n; ++i)

cin >> m[i] >> v[i];

init();

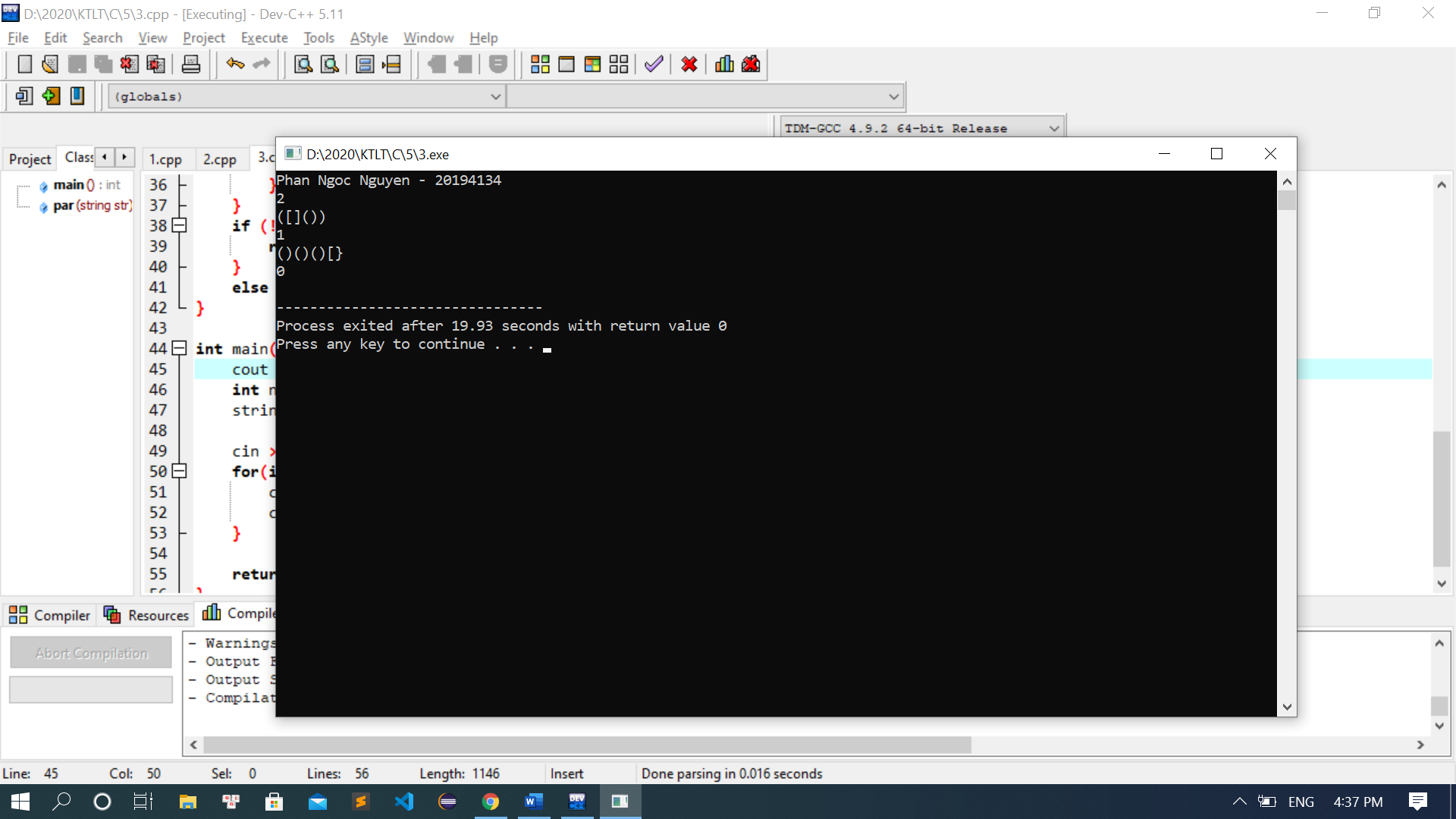
process(1);

print();

return 0;

}

## 5.3: Check ngoặc đúng



Code:

#include <iostream>

using namespace std;

#include <string.h>

#include <stack>

int par(string str){

int a = str.length();

stack<char> S;

char x, y;

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

x = str[i];

if (x == '(' || x == '[' || x == '{'){

S.push(x);

}

else {

if (x == ')') {

if (S.top() == '('){

S.pop();

}

else return 0;

}

else if (x == ']') {

if (S.top() == '['){

S.pop();

}

else return 0;

}

else if (x == '}') {

if (S.top() == '{'){

S.pop();

}

else return 0;

}

}

}

if (!S.empty()){

return 0;

}

else return 1;

}

int main(){

cout <<"Phan Ngoc Nguyen - 20194134" << endl;

int n;

string str;

cin >> n;

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

cin >> str;

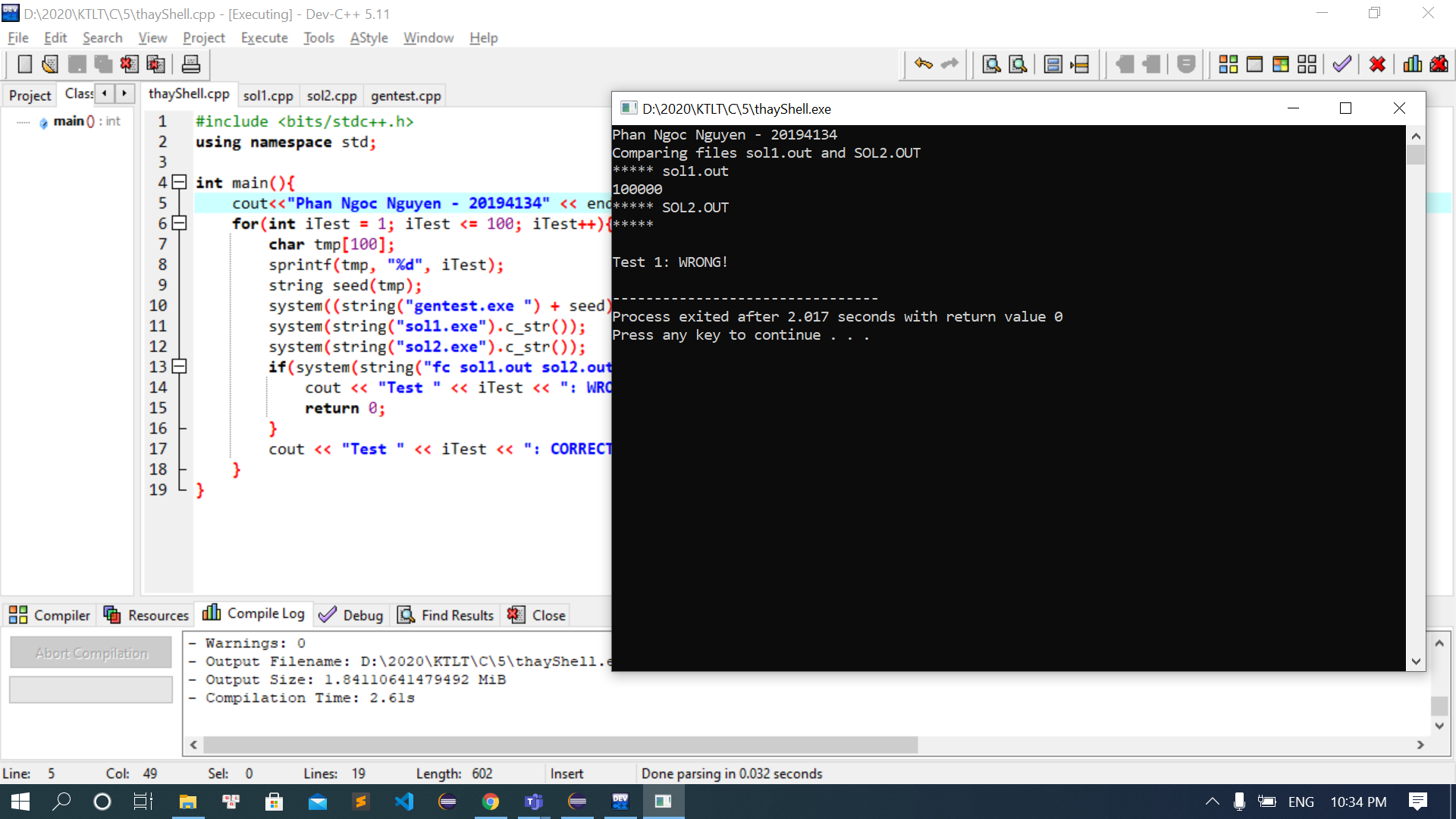
cout << par(str) << endl;

}

return 0;

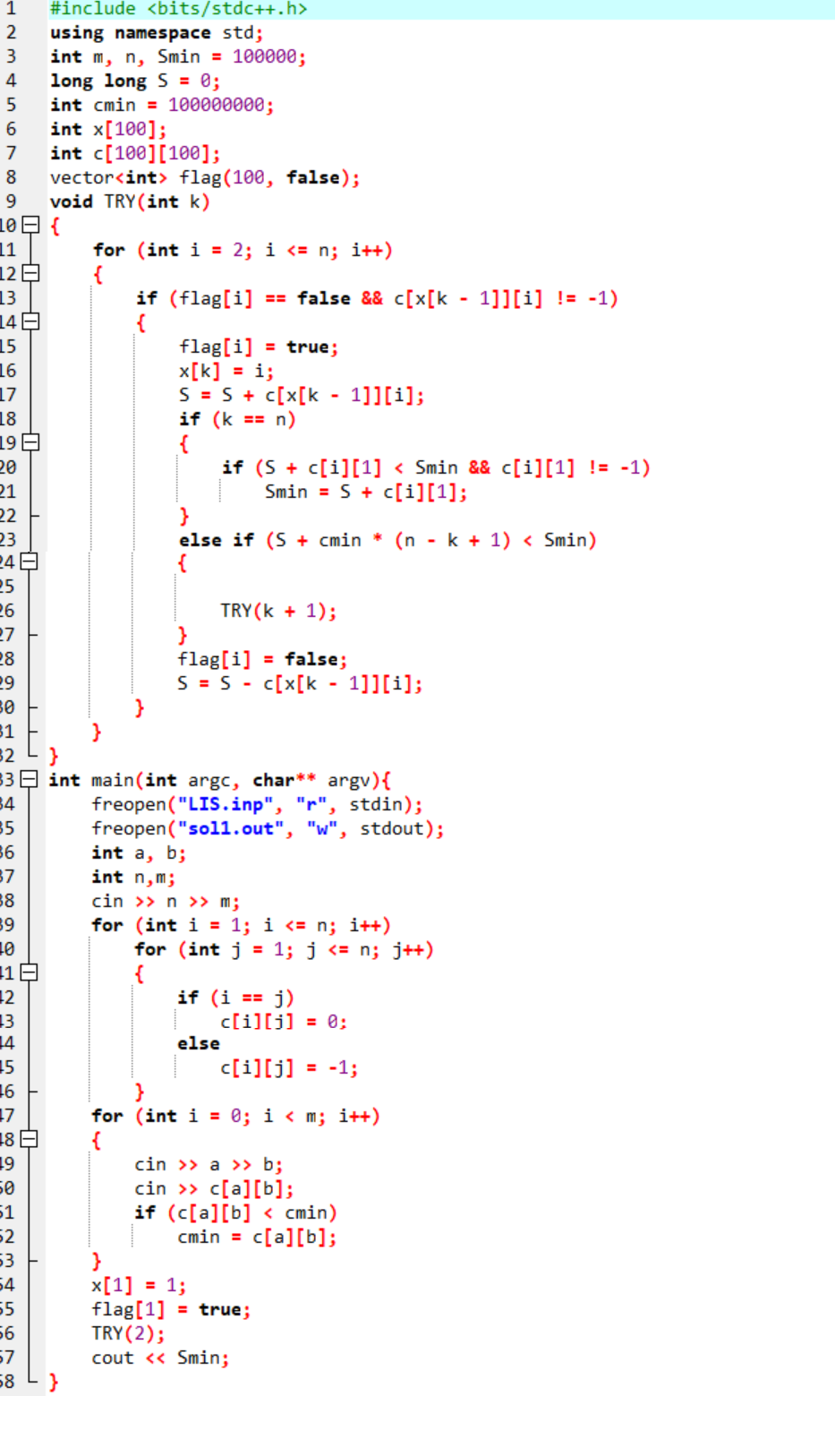
}

## 5.4: Bài toán người du lịch

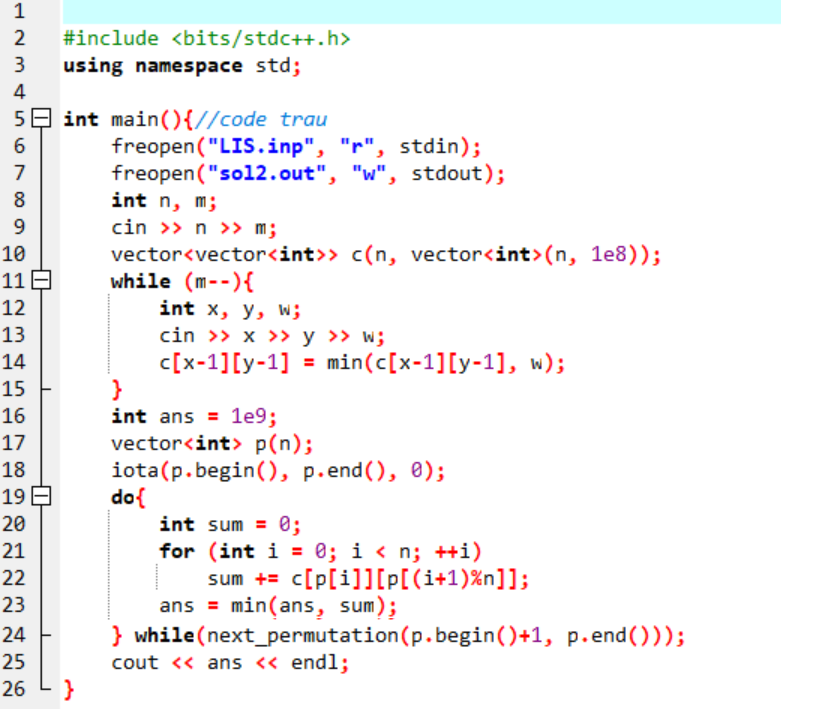


Code:

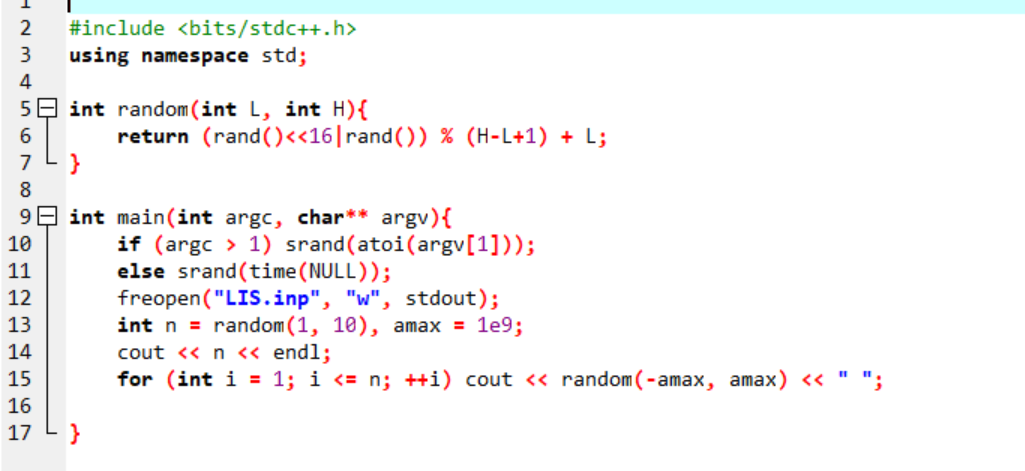
Sol1:



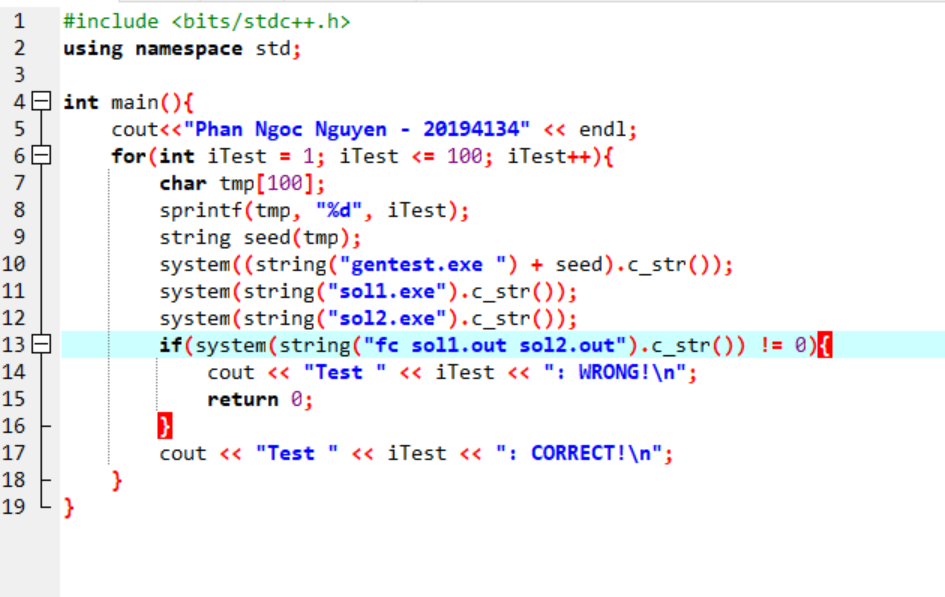
Sol2:



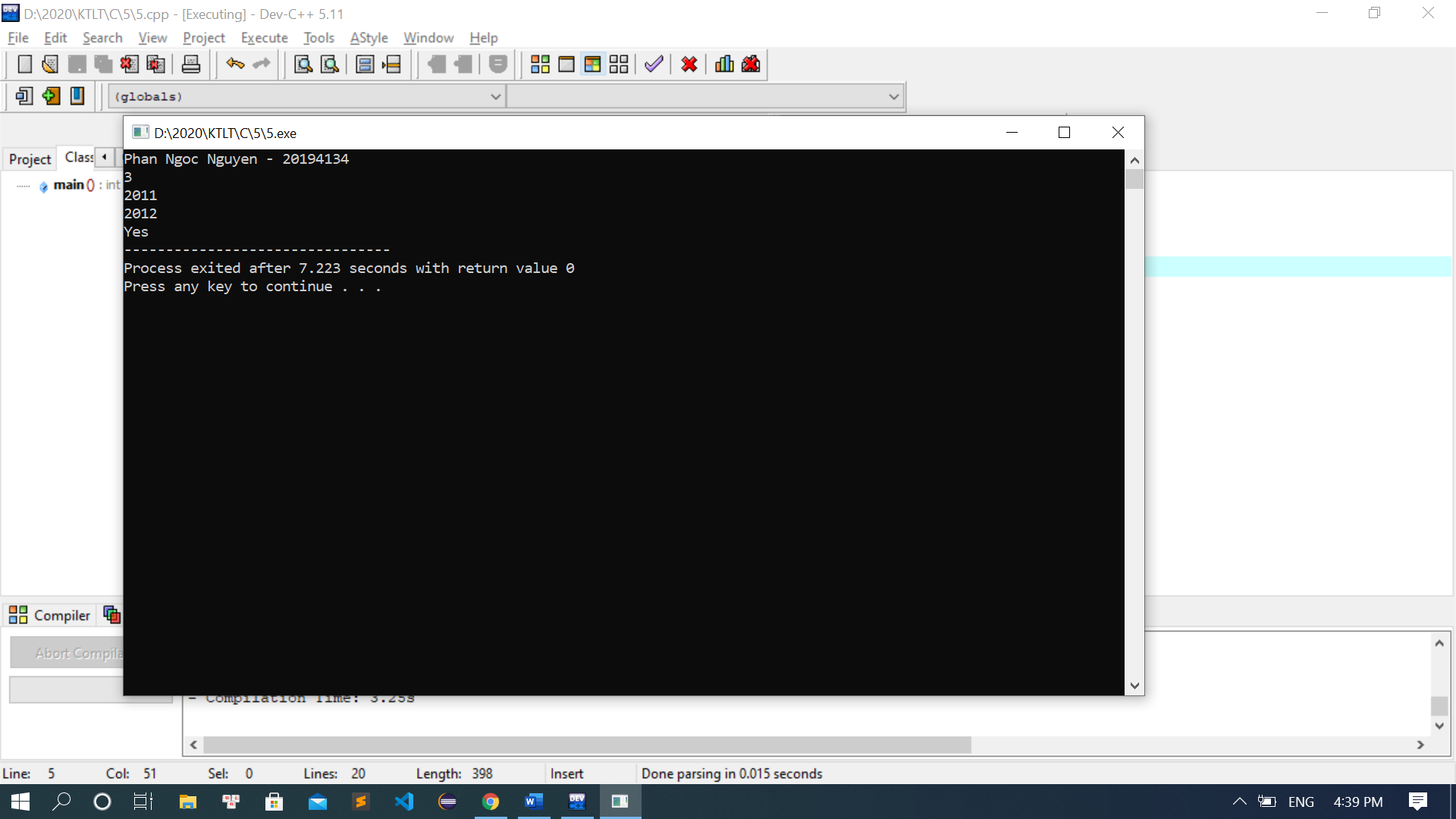
Gentest:



Shell:



## 5.5: Năm nhuận

Code:

#include <bits/stdc++.h>

using namespace std;

int main(){

cout << "Phan Ngoc Nguyen - 20194134" << endl;

int n;

cin >> n;

bool found = false;

while(n--){

int a;

cin >> a;

if ((a % 4 == 0 && a % 100 != 0) || (a % 100 == 0)){

found = true;

break;

}

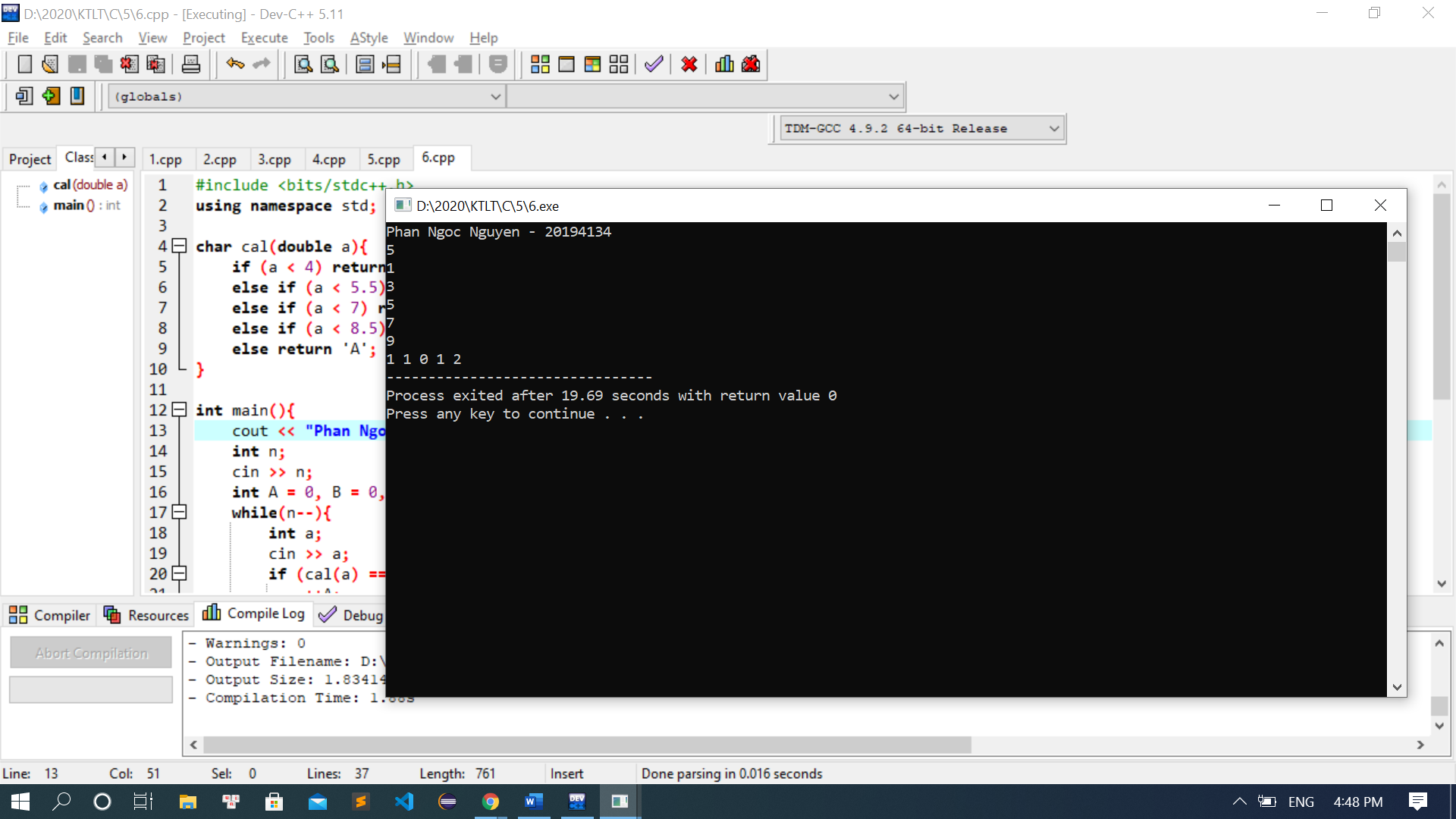
}

if (found) cout << "Yes";

else cout << "No";

}

## 5.6: Tổng kết



Code:

#include <bits/stdc++.h>

using namespace std;

char cal(double a){

if (a < 4) return 'F';

else if (a < 5.5) return 'D';

else if (a < 7) return 'C';

else if (a < 8.5) return 'B';

else return 'A';

}

int main(){

cout << "Phan Ngoc Nguyen - 20194134" << endl;

int n;

cin >> n;

int A = 0, B = 0, C = 0, D = 0, F = 0;

while(n--){

int a;

cin >> a;

if (cal(a) == 'A') {

++A;

}

else if (cal(a) == 'B'){

++B;

}

else if (cal(a) == 'C') {

++C;

}

else if (cal(a) == 'D') {

++D;

}

else if (cal(a) == 'F') {

++F;

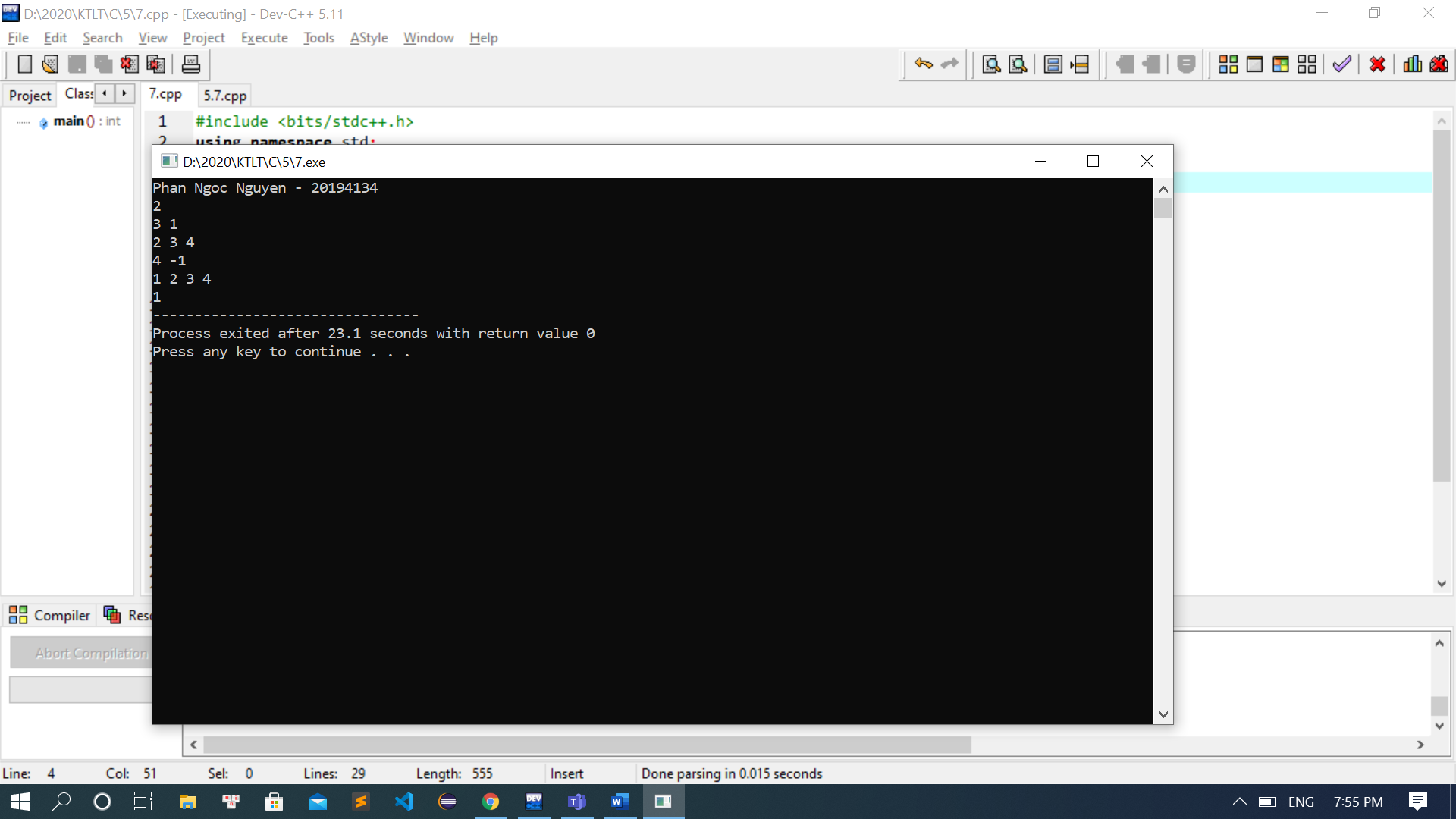
}

}

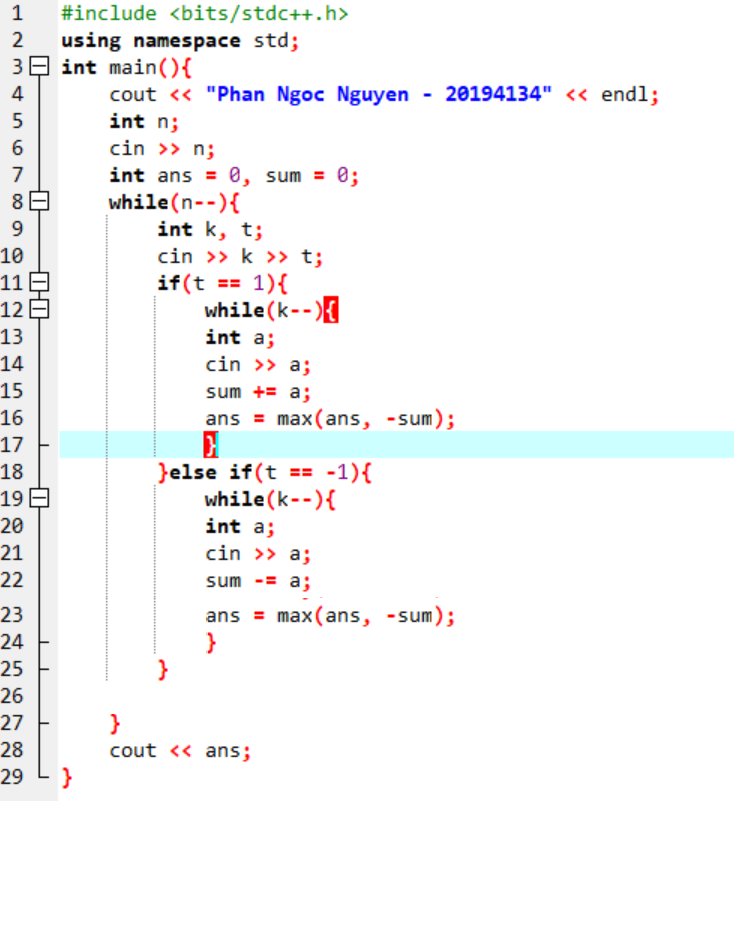
cout << A << " " << B << " " << C << " " << D << " " << F;

}

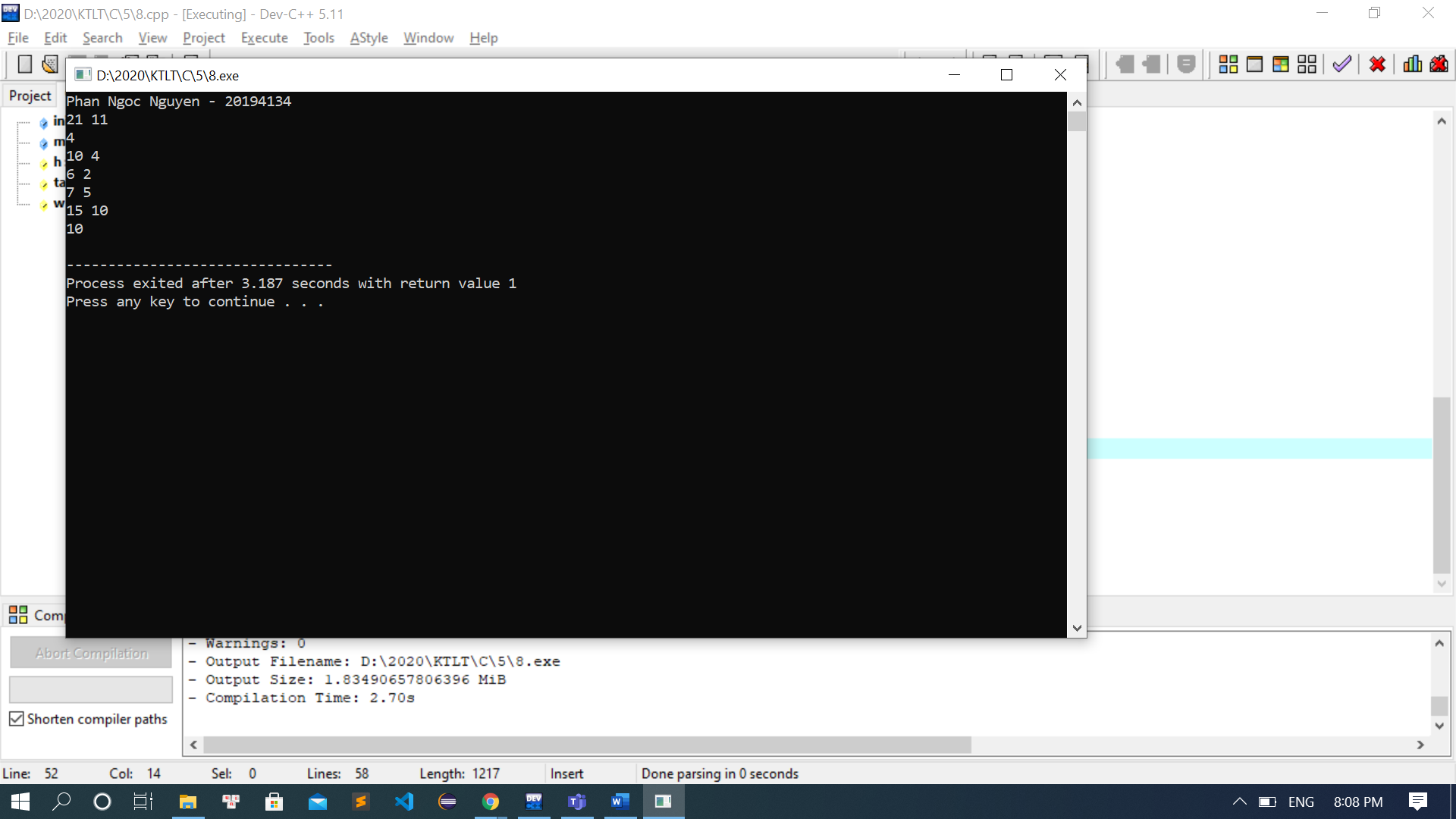
## 5.7: Chia tiền

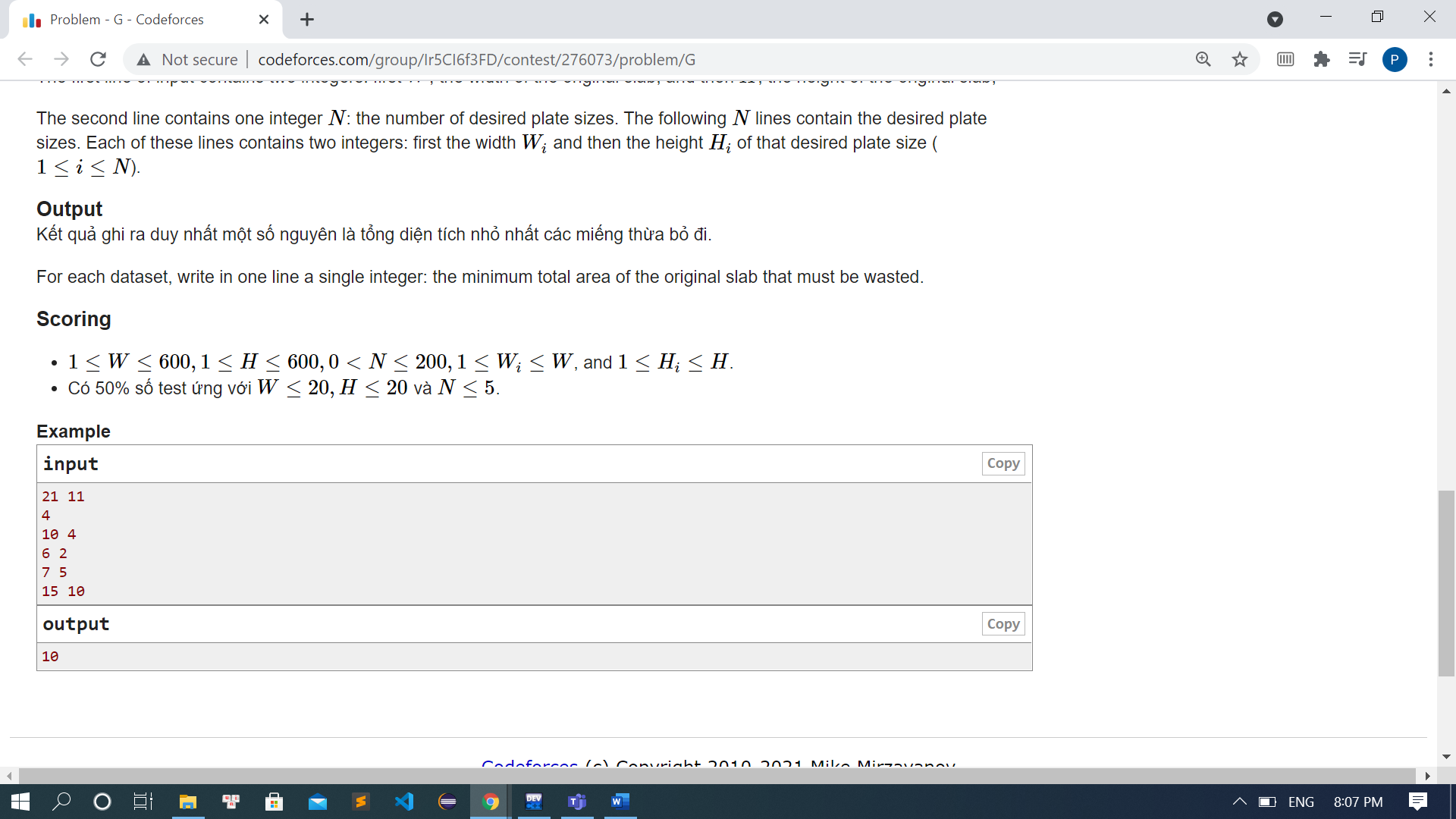


Code:



## 5.8: Cắt hình chữ nhật





#include <bits/stdc++.h>

using namespace std;

int w, h;

int table[601][601] ;

void init() {

for (int i=1; i<=h; i++) {

for (int j=1; j<=w; j++) {

table[i][j] = i\*j;

}

}

}

int main()

{

cout << "Phan Ngoc Nguyen - 20194134" <<endl;

ios::sync\_with\_stdio(false);

cin.tie();

int w, h, m;

cin >> w >> h;

cin >> m;

//init();

for (int i=1; i<=h; i++) {

for (int j=1; j<=w; j++) {

table[i][j] = i\*j;

}

}

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

int tmp1, tmp2;

cin >> tmp1 >> tmp2;

table[tmp2][tmp1] = 0;

}

//dp

for (int i=1; i<=h; i++) {

for (int j=1; j<=w; j++) {

int minWaste = table[i][j];

// horizonal cut

for(int k=1; k<=i; k++) {

minWaste = min(minWaste, table[k][j] + table[i-k][j]);

}

// vertical cut

for (int k=1; k<=j; k++) {

minWaste = min(minWaste, table[i][k] + table[i][j-k]);

}

table[i][j] = minWaste;

}

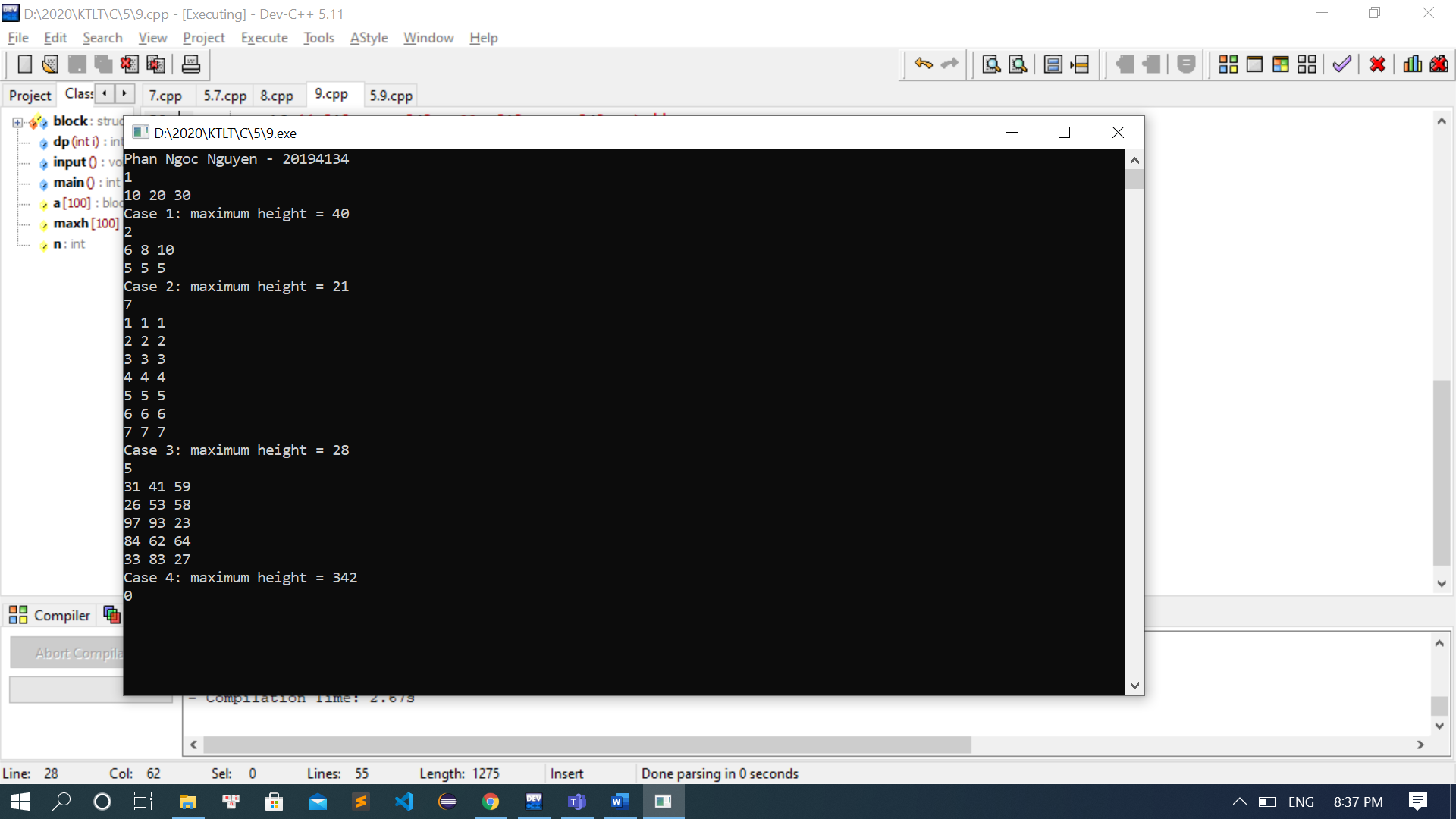
}

cout << table[h][w] << endl;

return 1;

}

## 5.9: Xây tháp



Code :

