Contents

[Bài 3.1: Dãy Lucas được định nghĩa bởi Ln = Ln-1 + Ln-2 với L0 = 2, L1 = 1. Hãy viết hàm tính số Lucas thứ n. 1](#_Toc71534154)

[Bài 3.2: Trên bàn cờ vua kích thước n\*n có một quân mã đang ở ô (1, 1). Hãy đưa ra một dãy các di chuyển của mã sao cho mỗi ô trên bàn cờ đều được đi qua đúng 1 lần (ô (1, 1) được xem là đã đi qua). 2](#_Toc71534155)

[Bài 3.3: Một người xuất phát tại thành phố 1, muốn đi thăm tất cả các thành phố khác, mỗi thành phố đúng 1 lần và quay về 1. Chi phí để đi từ thành phố i sang thành phố j là cij. Hãy tìm tổng chi phí nhỏ nhất có thể 5](#_Toc71534156)

[Bài 3.4: Cho dãy a có  n phần tử. Một dãy con của  a là dãy thu được bằng cách xóa đi một số phần tử của  a và giữ nguyên thứ tự các phần tử còn lại (có thể không xóa phần tử nào). Hãy tìm dãy con tăng dài nhất của a 8](#_Toc71534157)

[Câu 3.5: Tính hệ số tổ hợp C(n, k) 11](#_Toc71534158)

[Câu 3.6: Tìm ước chung lớn nhất của hai số nguyên a, b cho trước. 13](#_Toc71534159)

[Bài 3.7: Sử dụng phương pháp khử đệ quy bằng stack, hãy liệt kê các xâu nhị phân độ dài n không có k bit 1 nào liên tiếp 15](#_Toc71534160)

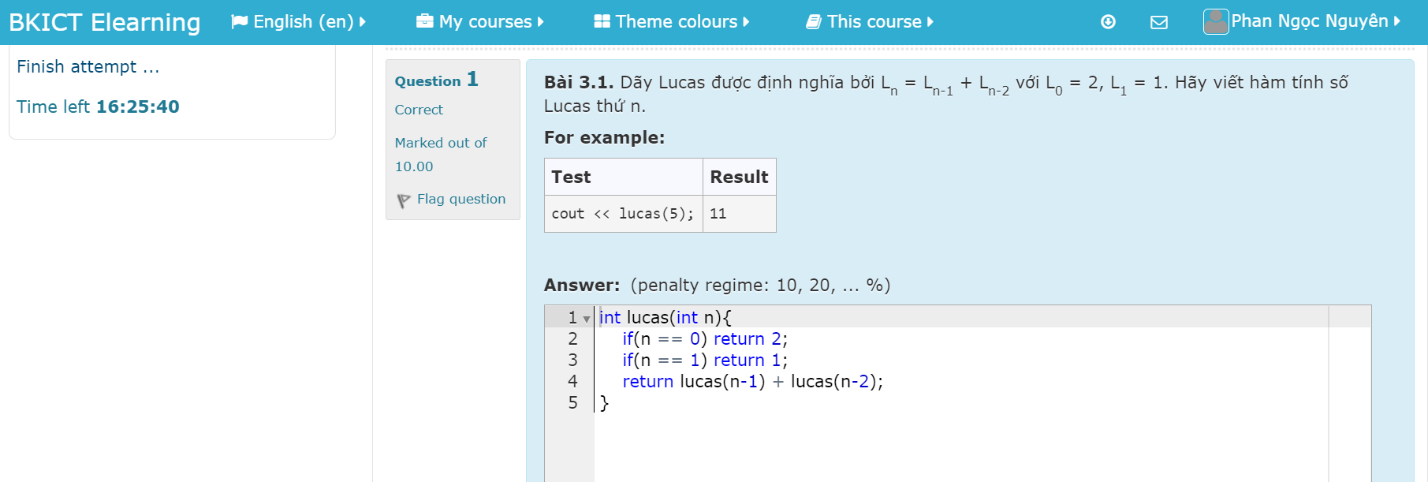
[Bài 3.9: Lập lịch cho y tá 17](#_Toc71534161)

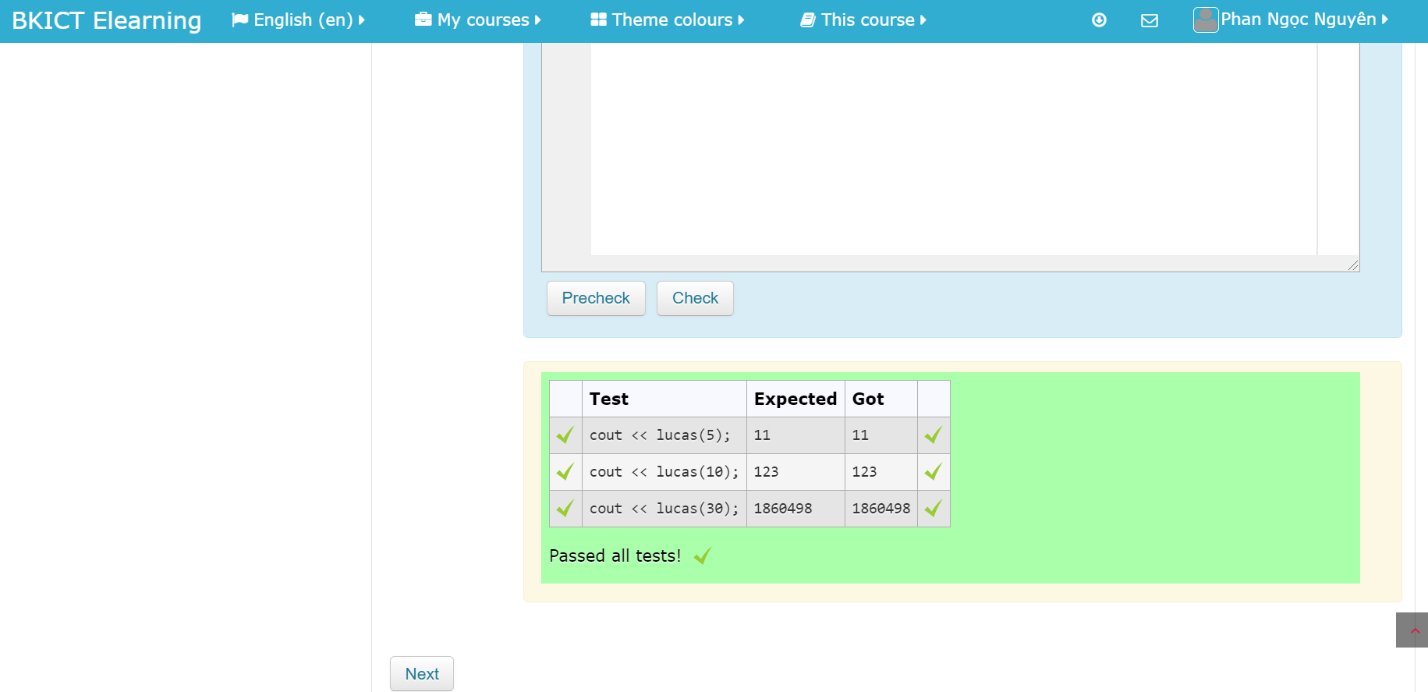
[Bài 3.10: Khoàng cách Hamming 19](#_Toc71534162)

[Bài 3.11: Lịch trình chụp ảnh: 21](#_Toc71534163)

[Bài 3.12: Đếm đường đi 23](#_Toc71534164)

## Bài 3.1: Dãy Lucas được định nghĩa bởi Ln = Ln-1 + Ln-2 với L0 = 2, L1 = 1. Hãy viết hàm tính số Lucas thứ n.





Code:

#include <iostream>

using namespace std;

int lucas(int n){

if(n == 0) return 2;

if(n == 1) return 1;

return lucas(n-1) + lucas(n-2);

}

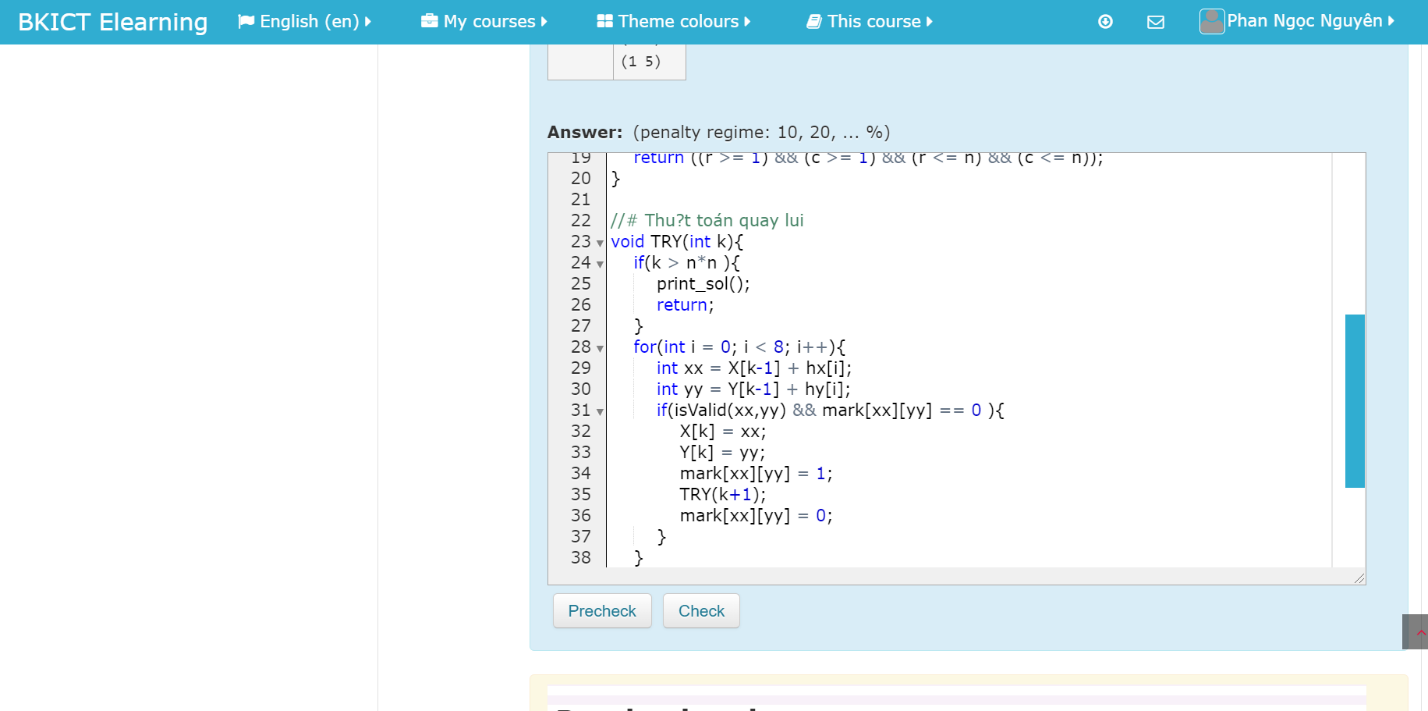
int main(){

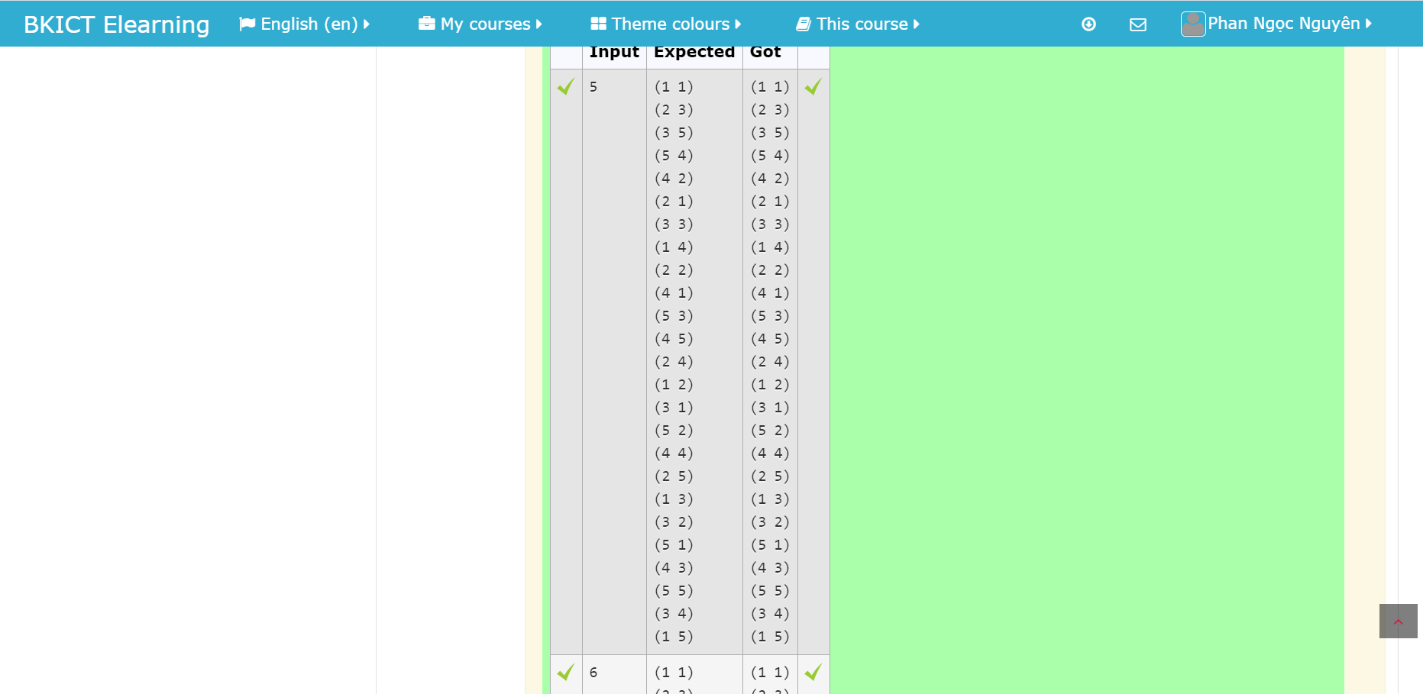
cout << "Phan Ngoc Nguyen - 20194134" << endl;

cout << lucas(5) << endl;

}

## Bài 3.2: Trên bàn cờ vua kích thước n\*n có một quân mã đang ở ô (1, 1). Hãy đưa ra một dãy các di chuyển của mã sao cho mỗi ô trên bàn cờ đều được đi qua đúng 1 lần (ô (1, 1) được xem là đã đi qua).





Code:

#include <iostream>

using namespace std;

int n;

int X[100], Y[100]; //# Luu t?a d? các bu?c di chuy?n c?a quân mã

int mark[100][100]; //# Ðánh d?u v? trí các ô mà quân mã dã di chuy?n qua

//# M?ng hx, hy mô t? 8 v? trí quân mã có th? di chuy?n k? t? v? trí hi?n t?i

const int hx[] = {1, 1, 2, 2, -1, -1, -2, -2};

const int hy[] = {2, -2, 1, -1, 2, -2, 1, -1};

//# In ra dãy các di chuy?n tìm du?c

void print\_sol(){

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

printf("(%d %d)\n", X[j], Y[j]);

exit(0);

}

bool isValid(int r ,int c){

return ((r >= 1) && (c >= 1) && (r <= n) && (c <= n));

}

//# Thu?t toán quay lui

void TRY(int k){

if(k > n\*n ){

print\_sol();

return;

}

for(int i = 0; i < 8; i++){

int xx = X[k-1] + hx[i];

int yy = Y[k-1] + hy[i];

if(isValid(xx,yy) && mark[xx][yy] == 0 ){

X[k] = xx;

Y[k] = yy;

mark[xx][yy] = 1;

TRY(k+1);

mark[xx][yy] = 0;

}

}

}

int main(){

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

cin >> n;

mark[1][1] = 1;

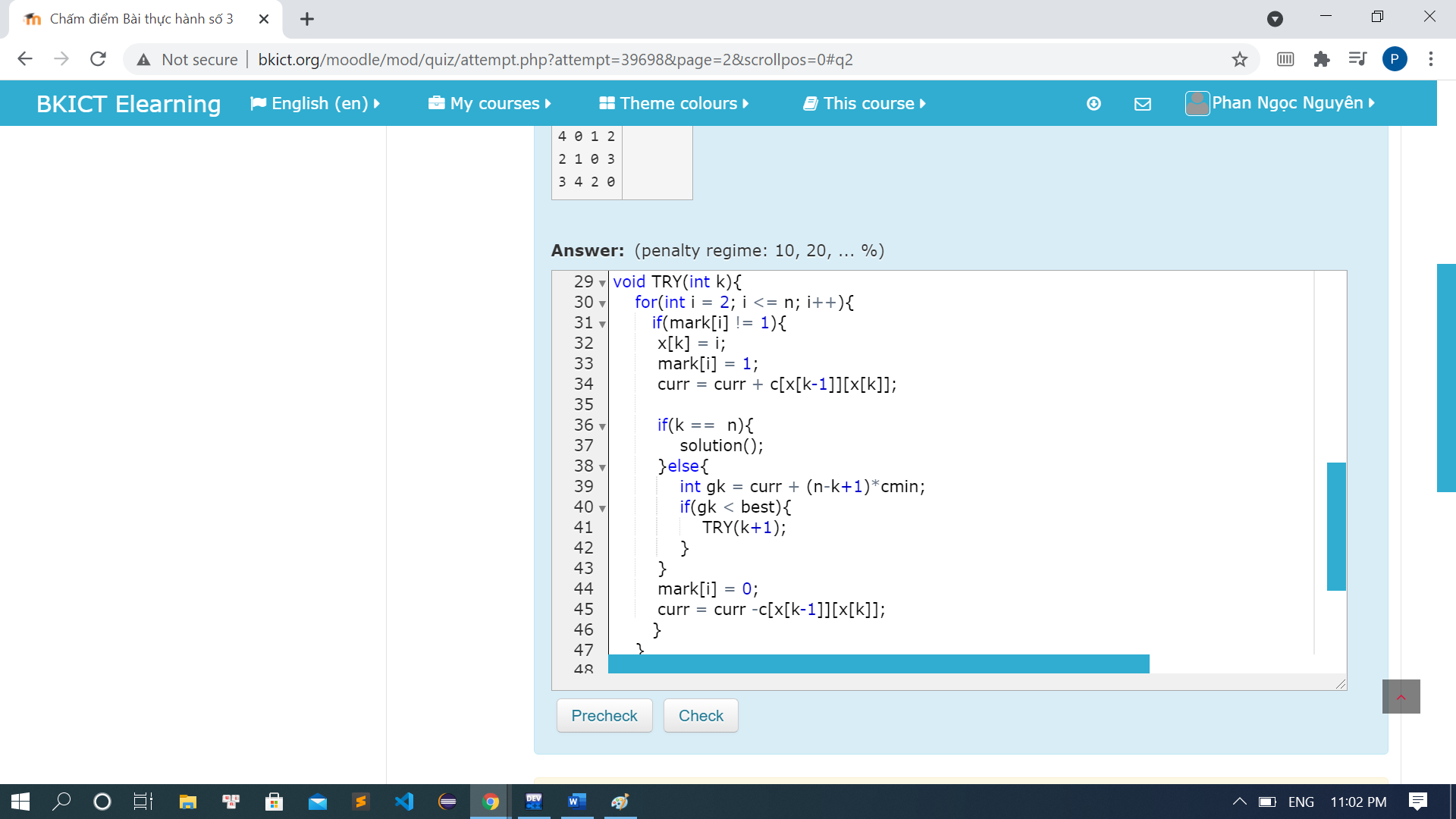
X[1] = Y[1] = 1;

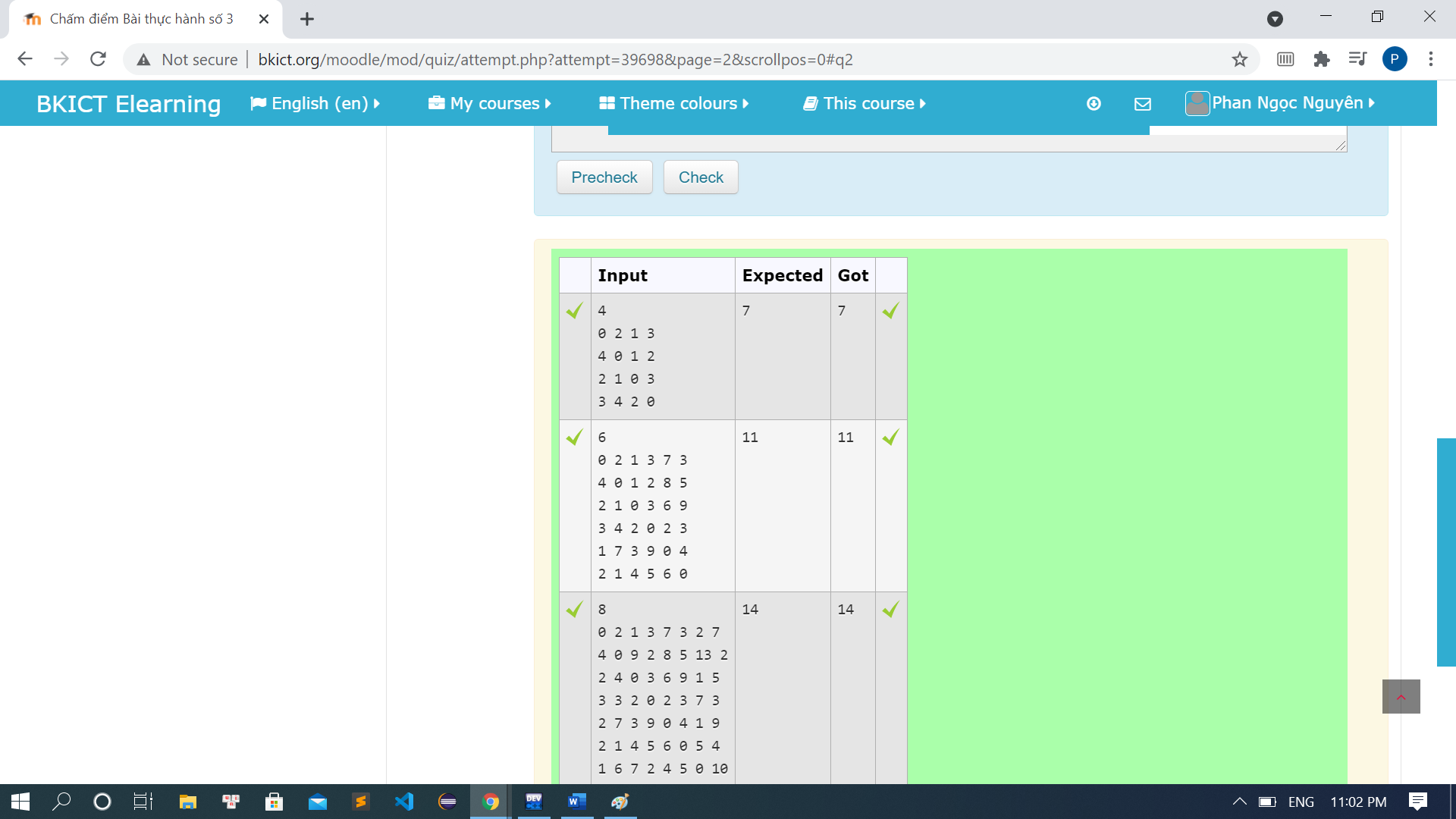
TRY(2);

return 0;

}

## Bài 3.3: Một người xuất phát tại thành phố 1, muốn đi thăm tất cả các thành phố khác, mỗi thành phố đúng 1 lần và quay về 1. Chi phí để đi từ thành phố i sang thành phố j là cij. Hãy tìm tổng chi phí nhỏ nhất có thể





Code:

#include <bits/stdc++.h>

using namespace std;

#define MAX 100

int n, c[MAX][MAX]; //# s? thành ph? và ma tr?n chi phí

int cmin = INT\_MAX; //# chi phí di l?i nh? nh?t gi?a hai thành ph? khác nhau

int best = INT\_MAX; //# t?ng chi phí nh? nh?t c?n tìm, ban d?u d?t b?ng giá tr? vô cùng l?n INT\_MAX = 2^31-1

int curr; //# t?ng chi phí t?i th?i di?m hi?n t?i

int mark[MAX]; //# dánh d?u nh?ng thành ph? dã di

int x[MAX]; //# luu gi? các thành ph? dã di

void input(){

cin >> n;

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

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

cin >> c[i][j];

if (c[i][j] > 0) cmin = min(cmin, c[i][j]);

}

}

void solution(){

if((curr + c[x[n]][x[1]]) < best){

best = curr + c[x[n]][x[1]];

}

}

//# Thu?t toán quay

void TRY(int k){

for(int v = 2; v <= n; v++){

if(mark[v] != 1){

x[k] = v;

mark[v] = 1;

curr = curr + c[x[k-1]][x[k]];

if(k == n){

solution();

}else{

int gk = curr + (n-k+1)\*cmin;

if(gk < best)

TRY(k+1);

}

mark[v] = 0;

curr = curr - c[x[k-1]][x[k]];

}

}

}

int main() {

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

input();

x[1] = 1;

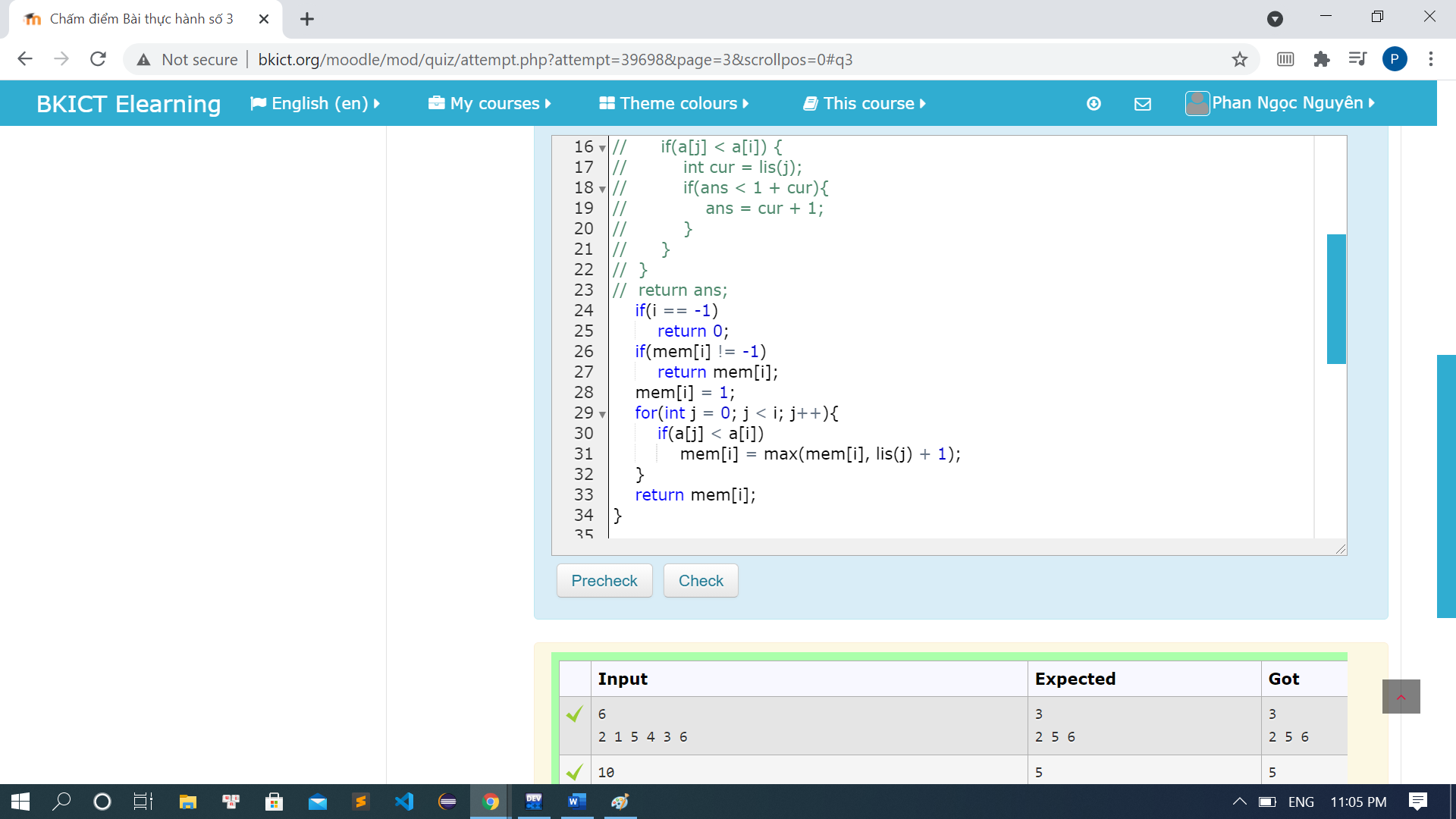
TRY(2);

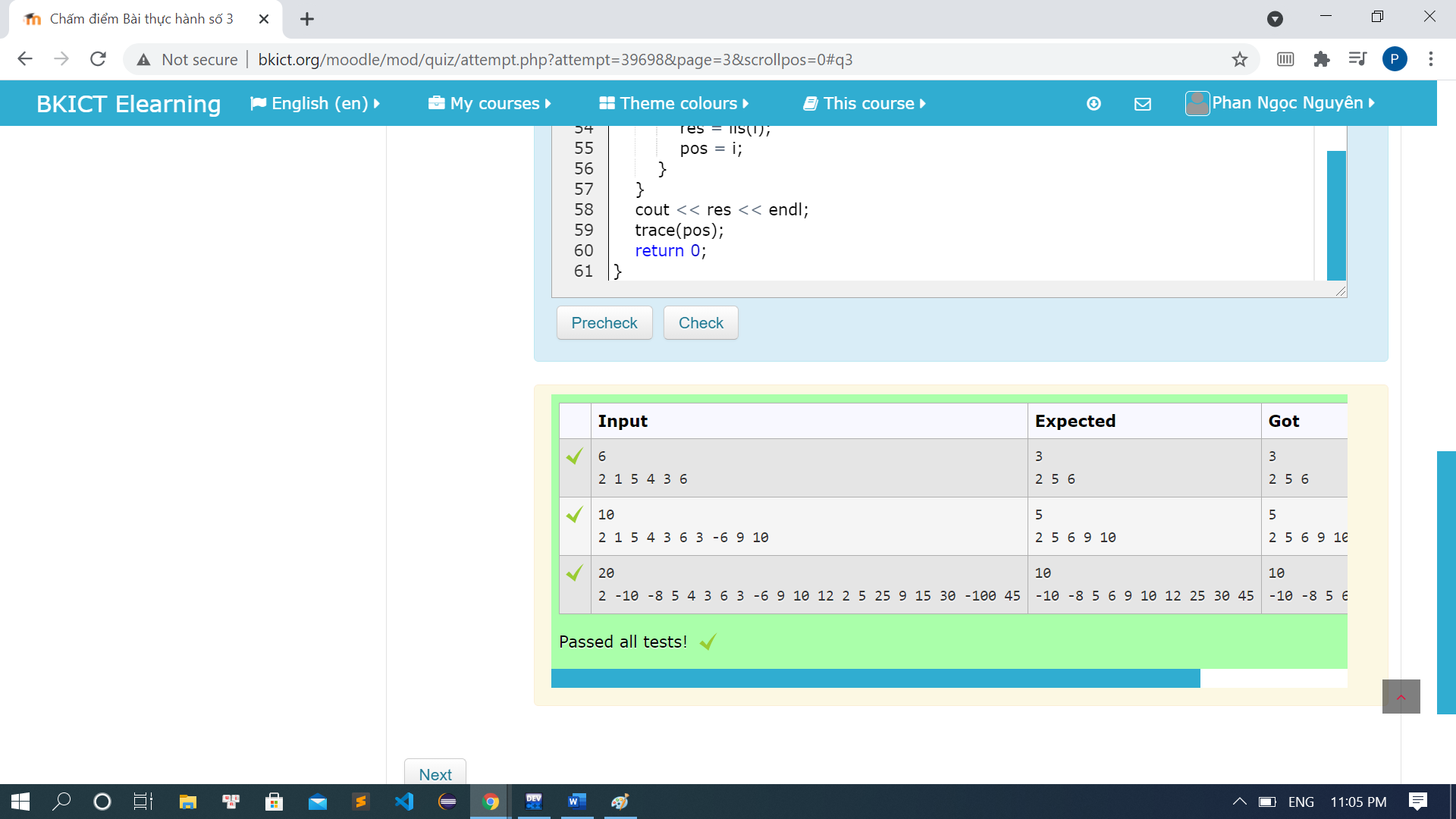
cout << best;

return 0;

}

## Bài 3.4: Cho dãy a có  n phần tử. Một dãy con của  a là dãy thu được bằng cách xóa đi một số phần tử của  a và giữ nguyên thứ tự các phần tử còn lại (có thể không xóa phần tử nào). Hãy tìm dãy con tăng dài nhất của a





Code:

#include <bits/stdc++.h>

using namespace std;

int a[1000], n;

int mem[1000]; //# m?ng ghi nh? l?i gi?i các bài toán con dã du?c gi?i

void init(){

memset(mem, -1, sizeof(mem));

}

//# Quy ho?ch d?ng,

//# Hàm lis(i) tr? v? d? dài dãy con tang dài nh?t k?t thúc b?i a[i]

int lis(int i) {

if(i == -1){

return 0;

}

if(mem[i] != -1){

return mem[i];

}

mem[i] = 1;

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

if(a[j] < a[i]){

mem[i] = max(mem[i], lis(j) + 1);

}

}

return mem[i];

}

//# Truy vet loi giai

void trace(int i){

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

if (a[j] < a[i] && mem[i] == 1 + mem[j]){

trace(j);

break;

}

}

cout << a[i] << " ";

}

int main(){

init();

cin >> n;

for(int i = 0; i < n; i++) cin >> a[i];

int res = 1, pos = 0;

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

if (res < lis(i)){

res = lis(i);

pos = i;

}

}

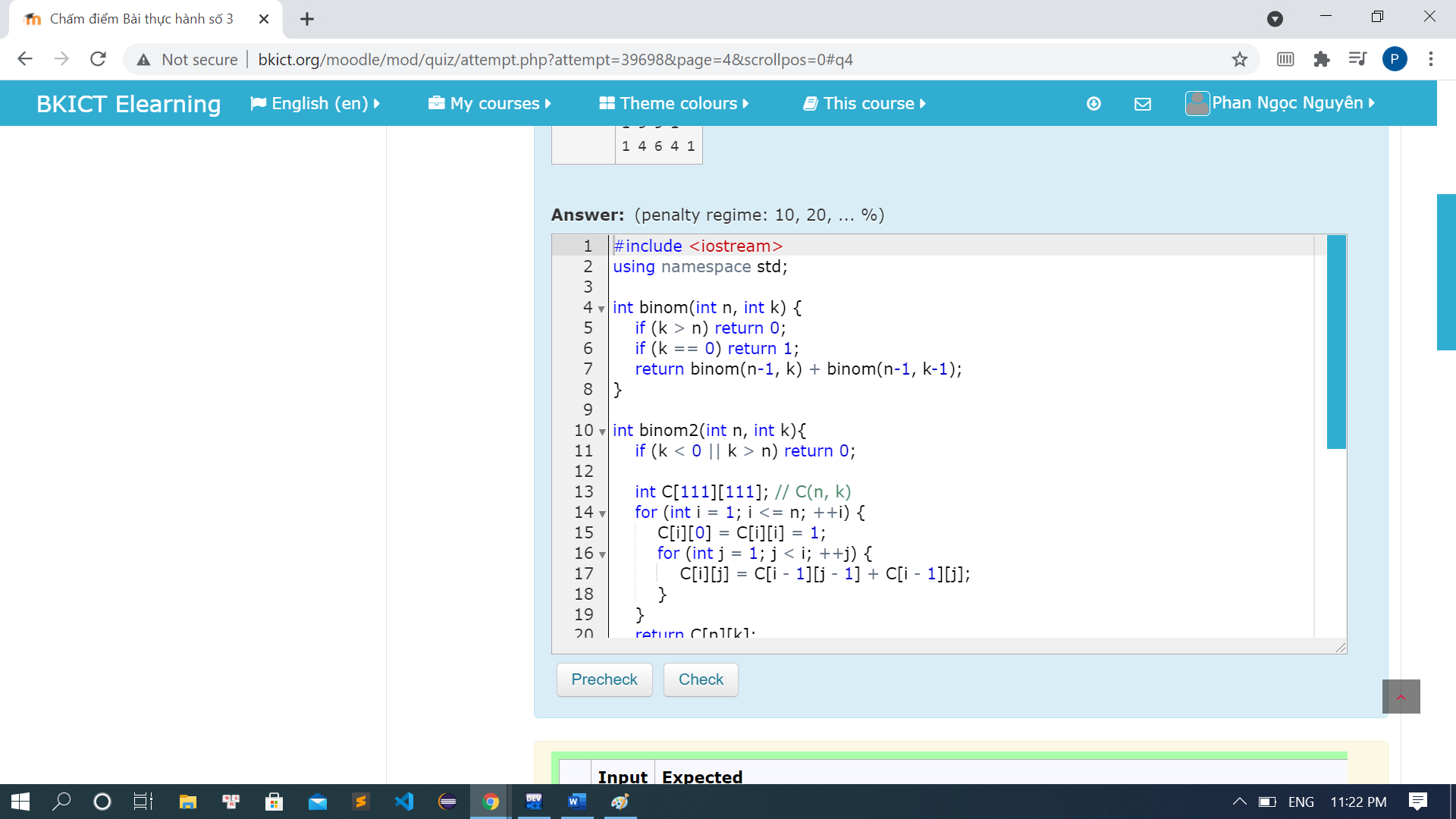
cout << res << endl;

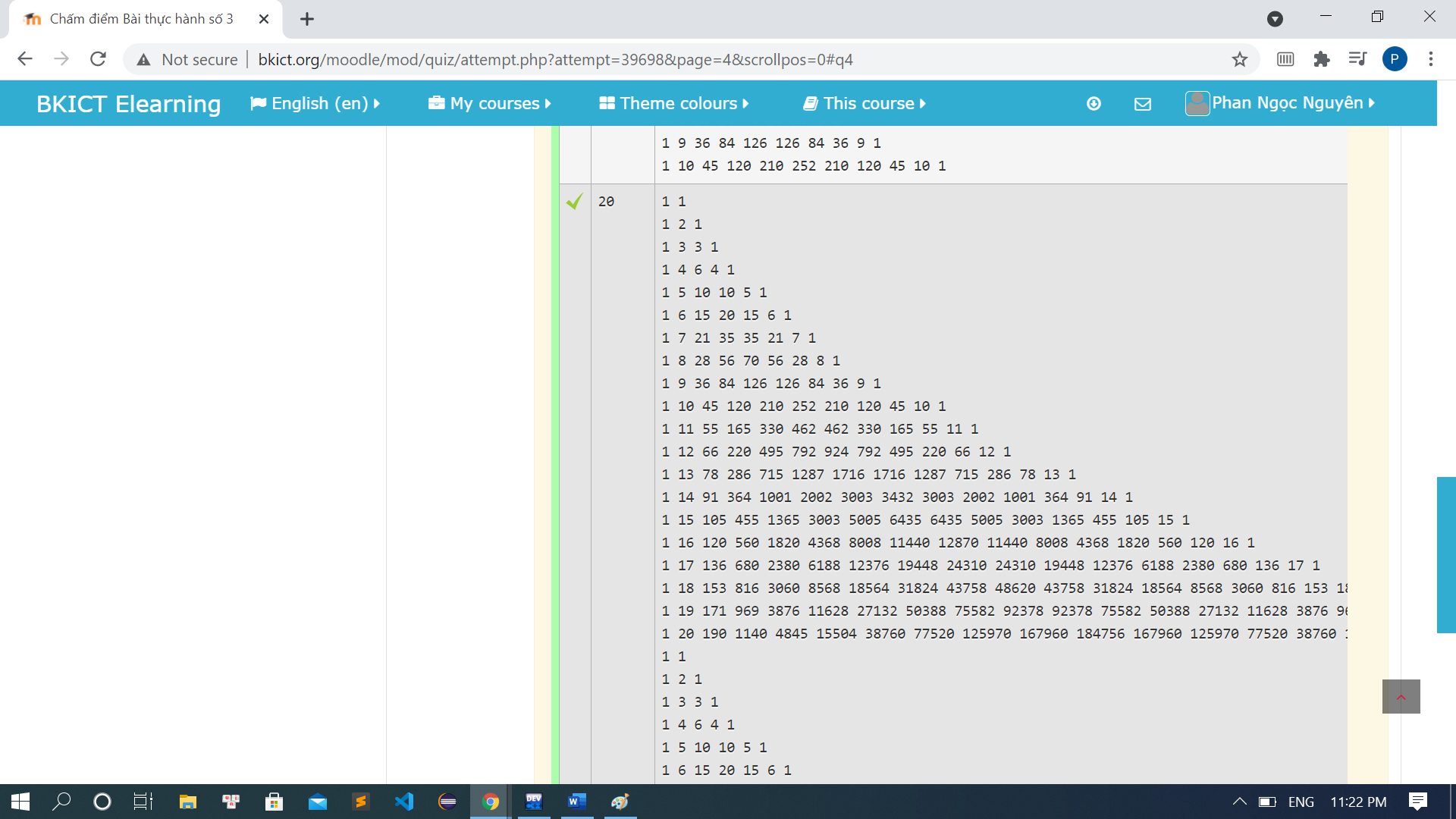
trace(pos);

return 0;

}

## Câu 3.5: Tính hệ số tổ hợp C(n, k)





Code:

#include <iostream>

using namespace std;

int binom(int n, int k) {

if (k > n) return 0;

if (k == 0) return 1;

return binom(n-1, k) + binom(n-1, k-1);

}

int binom2(int n, int k){

if (k < 0 || k > n) return 0;

int C[111][111]; // C(n, k)

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

C[i][0] = C[i][i] = 1;

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

C[i][j] = C[i - 1][j - 1] + C[i - 1][j];

}

}

return C[n][k];

}

int main() {

int m;

cin >> m;

for (int n = 1; n <= m; ++n){

for (int k = 0; k <= n; ++k)

printf("%d ", binom(n, k));

printf("\n");

}

for (int n = 1; n <= m; ++n){

for (int k = 0; k <= n; ++k)

printf("%d ", binom2(n, k));

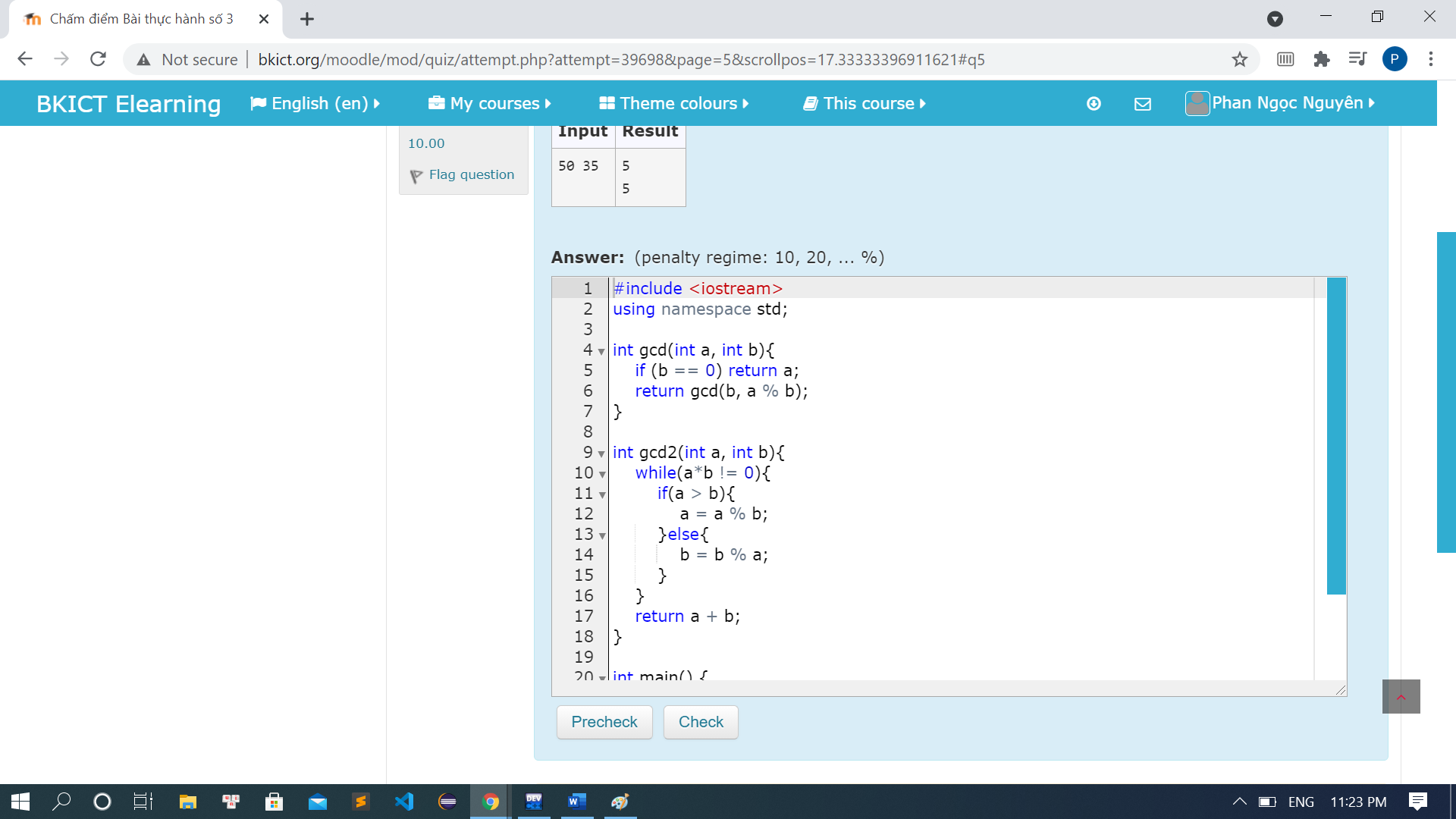
printf("\n");

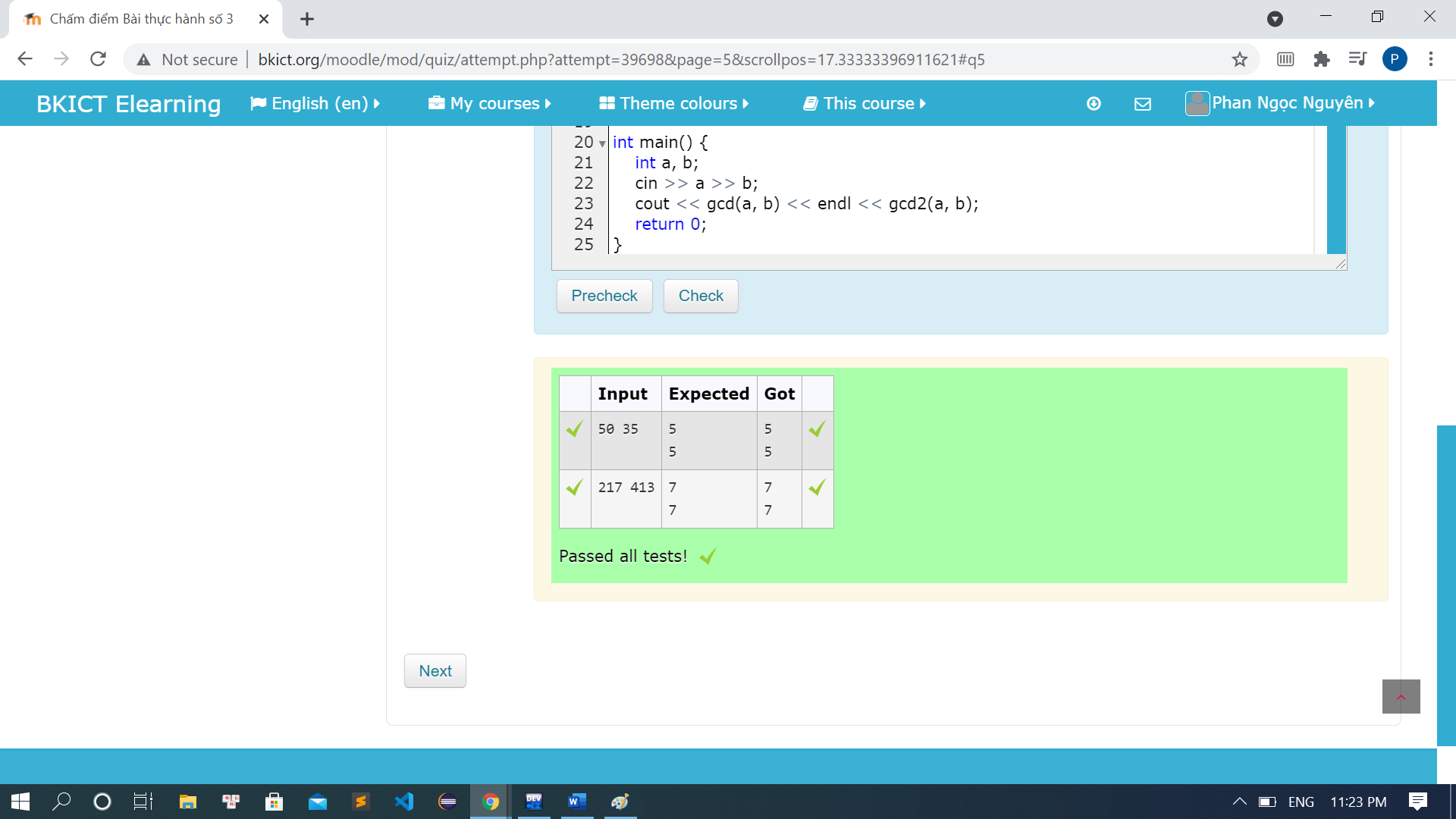
}

return 0;

}

## Câu 3.6: Tìm ước chung lớn nhất của hai số nguyên a, b cho trước.





Code:

#include <iostream>

using namespace std;

int gcd(int a, int b){

if (b == 0) return a;

return gcd(b, a % b);

}

int gcd2(int a, int b){

while(a\*b != 0){

if(a > b){

a = a % b;

}else{

b = b % a;

}

}

return a + b;

}

int main() {

int a, b;

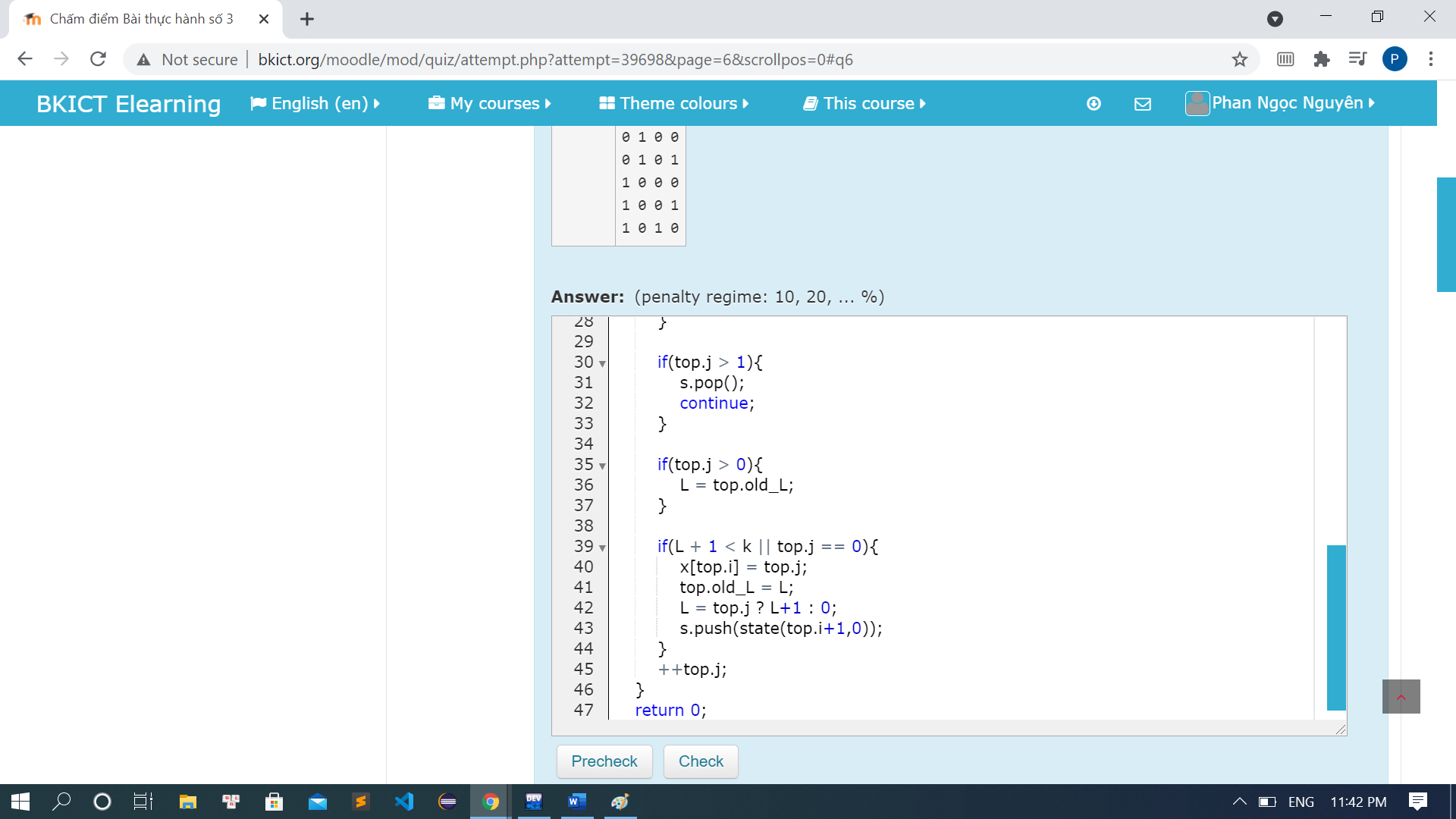
cin >> a >> b;

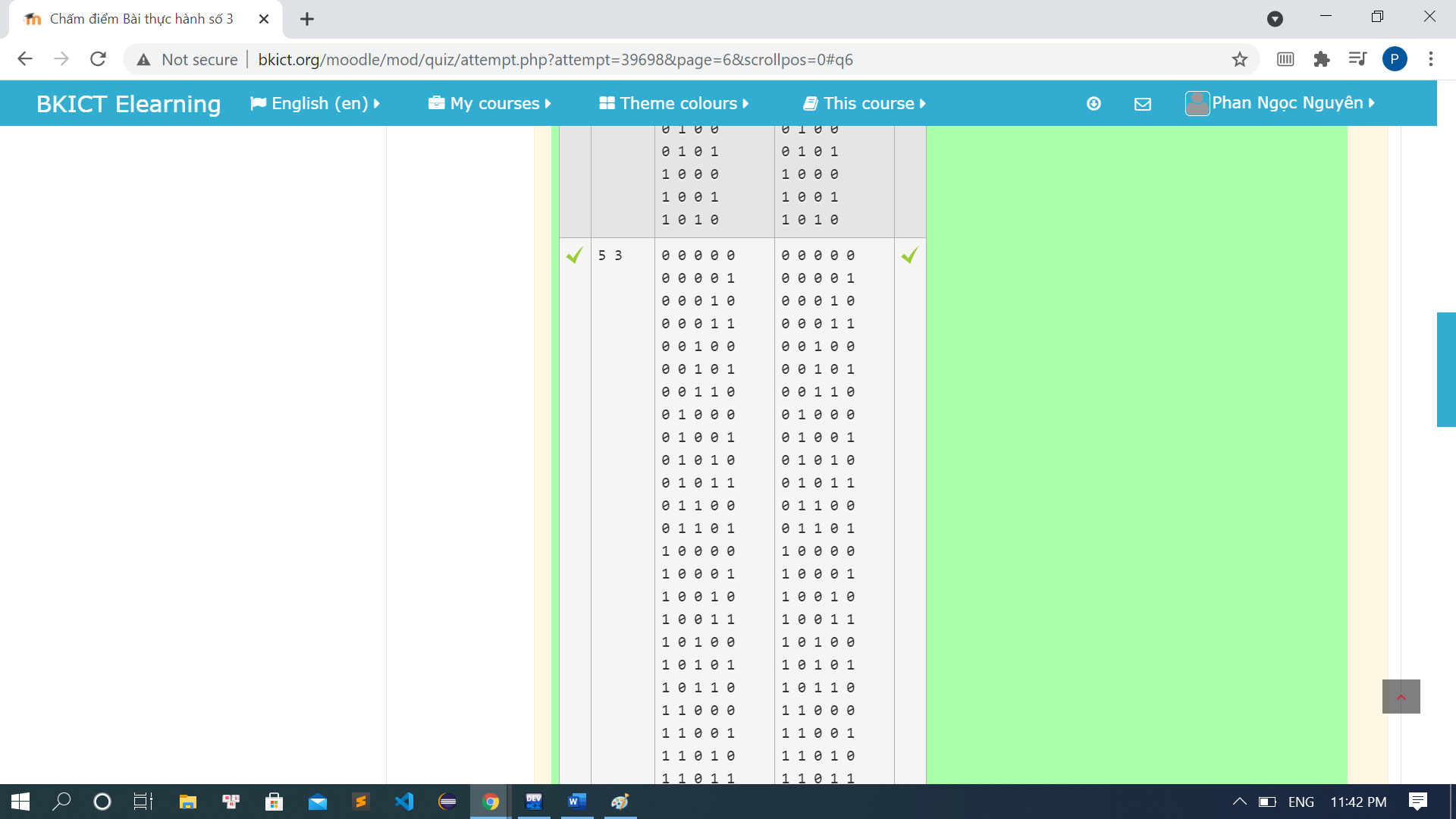
cout << gcd(a, b) << endl << gcd2(a, b);

return 0;

}

## Bài 3.7: Sử dụng phương pháp khử đệ quy bằng stack, hãy liệt kê các xâu nhị phân độ dài n không có k bit 1 nào liên tiếp





#include <bits/stdc++.h>

using namespace std;

struct state{

int i, j, old\_L;

//# constructor

state(int \_i = 0, int \_j = 0, int \_L = 0):

i(\_i), j(\_j), old\_L(\_L){}

};

int main() {

int n, k;

cin >> n >> k;

int x[n+1];

stack<state> s;

//# number of consecutive suffix 1

int L = 0;

s.push(state(1, 0));

while (!s.empty()){

state &top = s.top();

//# if a new binary sequence is found

if (top.i > n){

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

cout << x[i] << " \n"[i == n];

s.pop();

continue;

}

if(top.j > 1){

s.pop();

continue;

}

if(top.j > 0){

L = top.old\_L;

}

if(L + 1 < k || top.j == 0){

x[top.i] = top.j;

top.old\_L = L;

L = top.j ? L+1 : 0;

s.push(state(top.i+1,0));

}

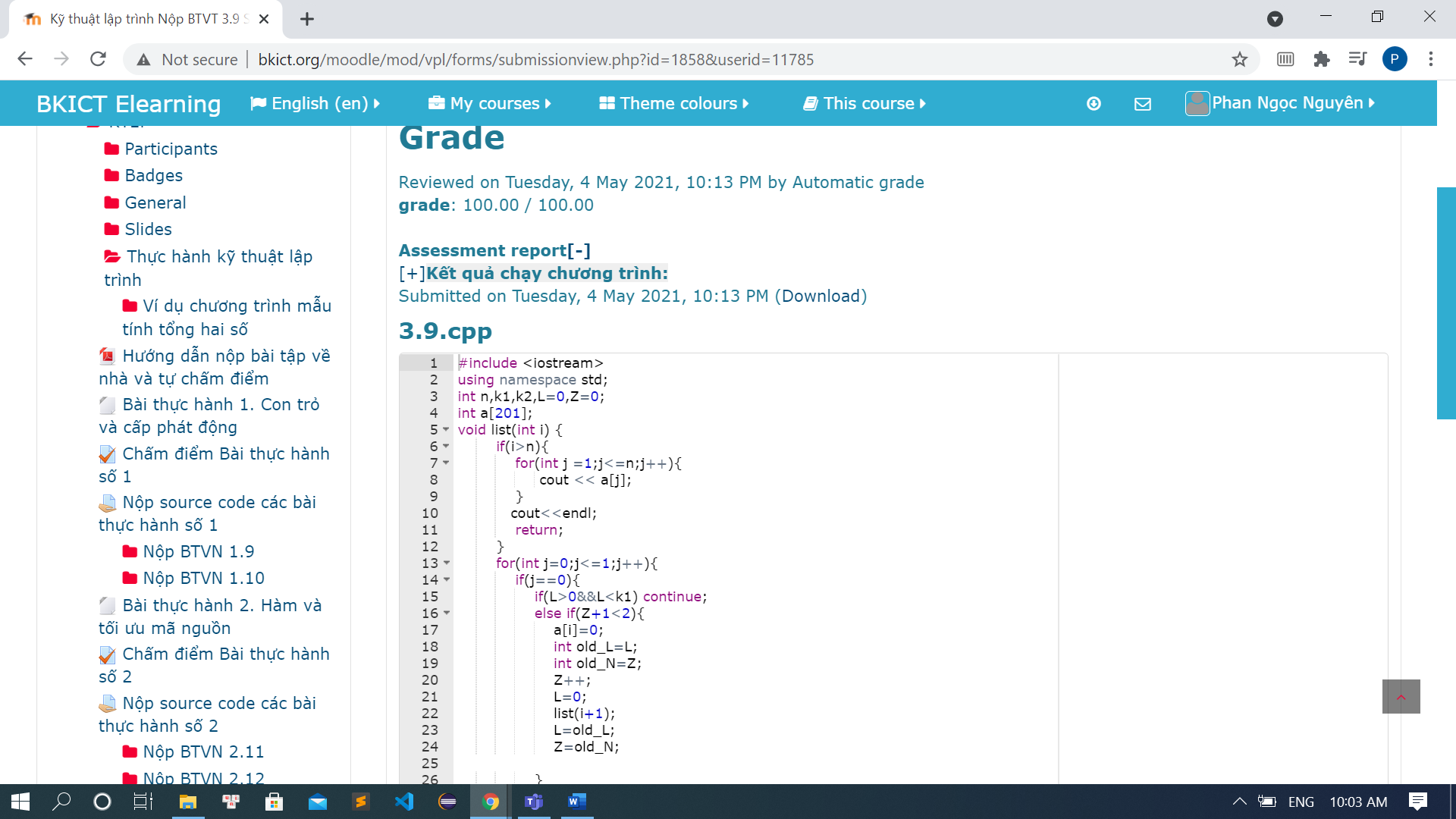
++top.j;

}

return 0;

}

## Bài 3.9: Lập lịch cho y tá



Code:

#include <iostream>

using namespace std;

int n,k1,k2,L=0,Z=0;

int a[201];

void list(int i) {

if(i>n){

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

cout << a[j];

}

cout<<endl;

return;

}

for(int j=0;j<=1;j++){

if(j==0){

if(L>0&&L<k1) continue;

else if(Z+1<2){

a[i]=0;

int old\_L=L;

int old\_N=Z;

Z++;

L=0;

list(i+1);

L=old\_L;

Z=old\_N;

}

}else{

if(i==n&&L+1<k1) continue;

else if(L<k2){

a[i]=1;

int old\_L=L;

int old\_N=Z;

L++;

Z=0;

list(i+1);

L=old\_L;

Z=old\_N;

}

}

}

}

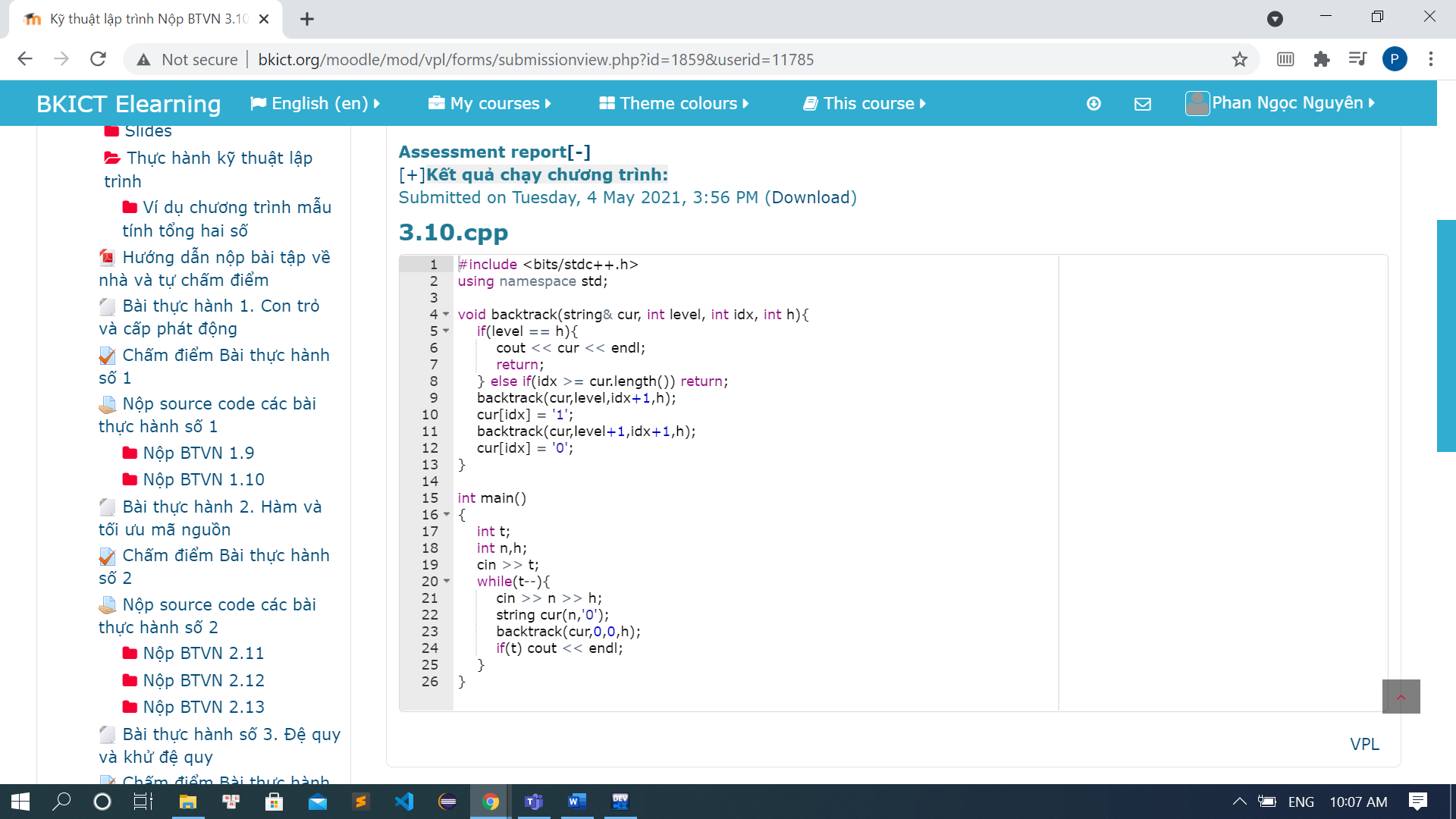
int main(){

cin >> n >> k1>>k2;

list(1);

}

## Bài 3.10: Khoàng cách Hamming



Code:

#include <bits/stdc++.h>

using namespace std;

void backtrack(string& cur, int level, int idx, int h){

if(level == h){

cout << cur << endl;

return;

} else if(idx >= cur.length()) return;

backtrack(cur,level,idx+1,h);

cur[idx] = '1';

backtrack(cur,level+1,idx+1,h);

cur[idx] = '0';

}

int main()

{

int t;

int n,h;

cin >> t;

while(t--){

cin >> n >> h;

string cur(n,'0');

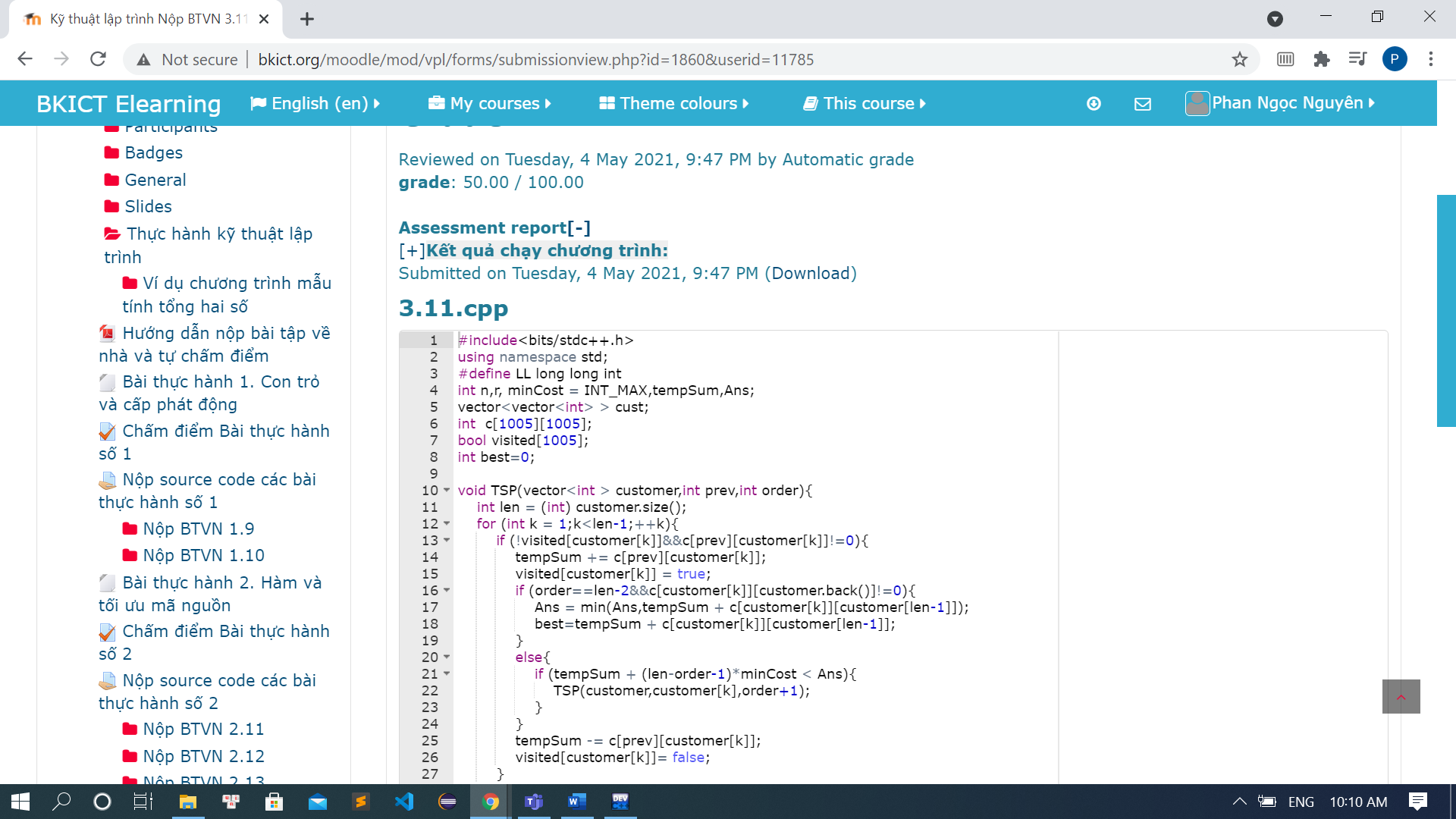
backtrack(cur,0,0,h);

if(t) cout << endl;

}

}

## Bài 3.11: Lịch trình chụp ảnh:



Code:

#include<bits/stdc++.h>

using namespace std;

#define LL long long int

int n,r, minCost = INT\_MAX,tempSum,Ans;

vector<vector<int> > cust;

int c[1005][1005];

bool visited[1005];

int best=0;

void TSP(vector<int > customer,int prev,int order){

int len = (int) customer.size();

for (int k = 1;k<len-1;++k){

if (!visited[customer[k]]&&c[prev][customer[k]]!=0){

tempSum += c[prev][customer[k]];

visited[customer[k]] = true;

if (order==len-2&&c[customer[k]][customer.back()]!=0){

Ans = min(Ans,tempSum + c[customer[k]][customer[len-1]]);

best=tempSum + c[customer[k]][customer[len-1]];

}

else{

if (tempSum + (len-order-1)\*minCost < Ans){

TSP(customer,customer[k],order+1);

}

}

tempSum -= c[prev][customer[k]];

visited[customer[k]]= false;

}

}

}

main(){

cin >> n >> r;

cust.resize(r);

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

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

cin >> c[i][j];

if (c[i][j]!=0) minCost = min(minCost,c[i][j]);

}

}

getchar();

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

string line;

getline(cin,line);

istringstream is(line);

cust[i] = (vector<int>(istream\_iterator<int>(is), istream\_iterator<int>()));

}

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

tempSum = 0; Ans = INT\_MAX;

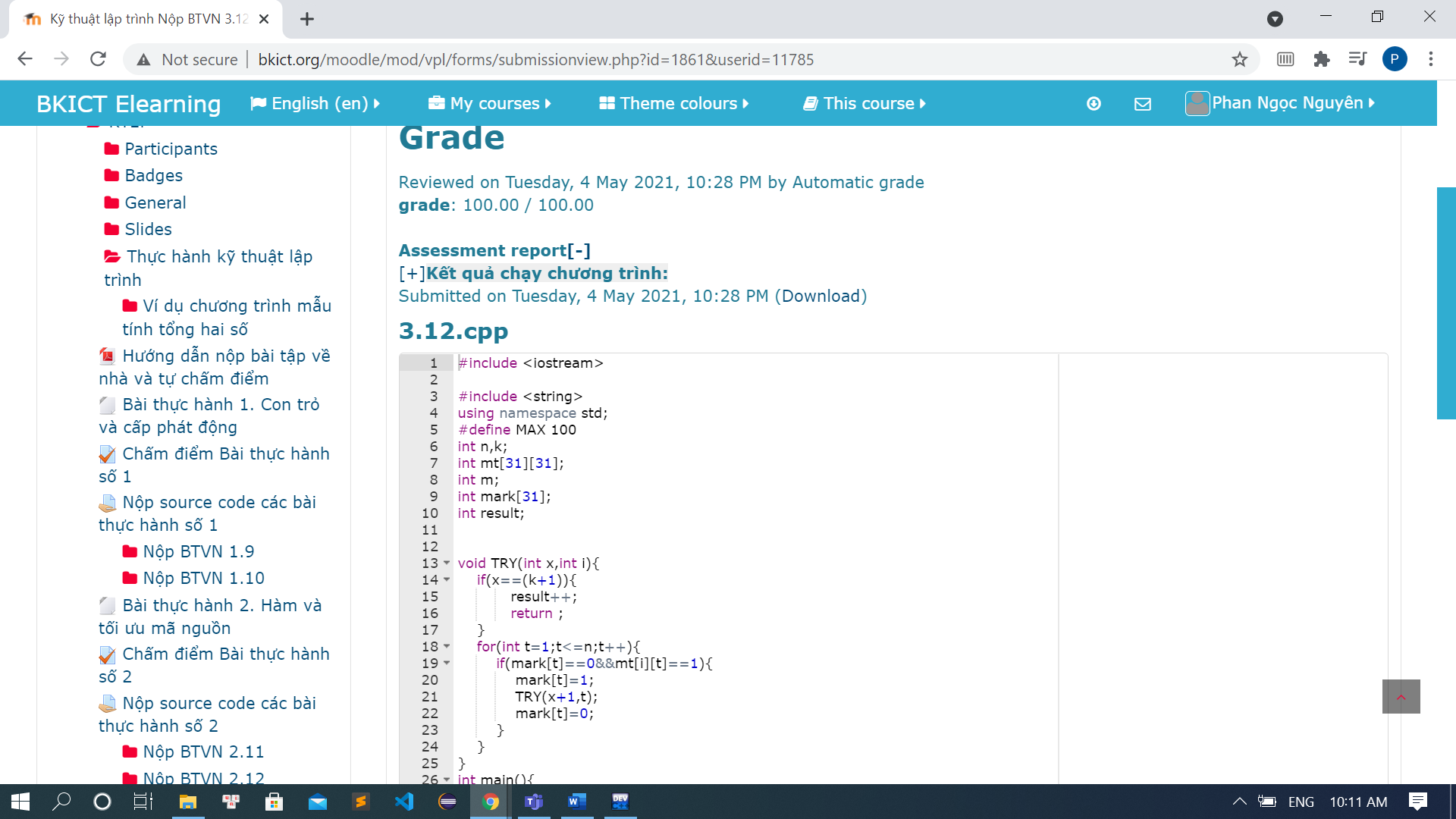
TSP(cust[i],(cust[i]).at(0),1);

cout << (Ans%INT\_MAX) << endl;

}

}

## Bài 3.12: Đếm đường đi



Code:

#include <iostream>

#include <string>

using namespace std;

#define MAX 100

int n,k;

int mt[31][31];

int m;

int mark[31];

int result;

void TRY(int x,int i){

if(x==(k+1)){

result++;

return ;

}

for(int t=1;t<=n;t++){

if(mark[t]==0&&mt[i][t]==1){

mark[t]=1;

TRY(x+1,t);

mark[t]=0;

}

}

}

int main(){

int d\_dau;

int d\_dich ;

cin>>n>>k;

cin>> m;

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

cin>> d\_dau>> d\_dich;

mt[d\_dau][d\_dich]=1;

mt[d\_dich][d\_dau]=1;

}

result=0;

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

for(int t=1;t<=n;t++) mark[t]=0;

mark[i]=1;

TRY(1,i);

}

cout<< result/2 ;

}