**BÁO CÁO BUỔI THỰC HÀNH SỐ 3 – Tuần 13**

***Bộ Môn Kỹ Thuật Lập Trình***

A blue logo with black background

Description automatically generated

A red and white logo

Description automatically generated

Sinh viên: **Nguyễn Duy Khánh**

MSSV: **20225019**

Mã lớp thực hành: **744469**

Giáo viên hướng dẫn: **ThS. Lê Thị Hoa**

Hỗ trợ giảng dạy: **Đoàn Sỹ Nguyên**

MỤC LỤC

[Hình 1.1: Code bài 1. 3](#_Toc183776477)

[Hình 1.2: Test case bài 1. 4](#_Toc183776478)

[Hình 2.1: Code bài 2. 5](#_Toc183776477)

[Hình 2.2: Code bài 2. 5](#_Toc183776478)

[Hình 2 3: Code bài 2. 6](#_Toc183776479)

[Hình 2.4: Test case bài 2 6](#_Toc183776480)

[Hình 2.5: Test case bài 2 7](#_Toc183776481)

[Hình 3.1: Code bài 3. 9](#_Toc183776526)

[Hình 3.2: Code bài 3. 10](#_Toc183776527)

[Hình 3.3: Code bài 3. 10](#_Toc183776528)

[Hình 3.4: Test case bài 3. 11](#_Toc183776529)

[Hình 4.1: Code bài 4. 14](#_Toc183776482)

[Hình 4.2: Code bài 4. 14](#_Toc183776482)

[Hình 4.3: Code bài 4. 15](#_Toc183776482)

[Hình 4.4: Test case bài 4. 15](#_Toc183776483)

[Hình 5.1: Code bài 5 18](#_Toc183776484)

[Hình 5.2: Code bài 5 18](#_Toc183776485)

[Hình 5.3: Test case 1 19](#_Toc183776486)

[Hình 5.4: Test case 2 20](#_Toc183776487)

[Hình 6.1: Code bài 6. 22](#_Toc152156742)

[Hình 6.2: Code bài 6. 22](#_Toc152156742)

[Hình 6.3: Test case bài 6. 23](#_Toc152156742)

[Hình 7.1: Code bài 7. 24](#_Toc152156792)

[Hình 7.2: Code bài 7. 24](#_Toc152156792)

[Hình 7.3: Test case bài 7. 25](#_Toc152156792)

[Hình 8.1: Code bài 8. 27](#_Toc152156742)

[Hình 8.2: Code bài 8. 27](#_Toc152156742)

[Hình 8.3: Test case bài 8. 28](#_Toc152156742)

[Hình 9.1: Code bài 9.. 30](#_Toc152156742)

[Hình 9.2: Code bài 9. 30](#_Toc152156742)

[Hình 9.3: Test case 1 bài 9. 31](#_Toc152156742)

[Hình 9.4: Test case 2 bài 9. 31](#_Toc152156825)

[Hình 9.5: Test case 3 bài 9. 32](#_Toc152156825)

[Hình 10.1: Code bài 10. 35](#_Toc152156742)

[Hình 10.2: Test case bài 10 - Ảnh 1 36](#_Toc152156742)

[Hình 10.3: Test case bài 10 - Ảnh 2 37](#_Toc152156742)

[Hình 11.1: Code bài 11. 38](#_Toc152156742)

[Hình 11.2: Code bài 11. 38](#_Toc152156742)

[Hình 11.3: Test case 1 – B11. 39](#_Toc152156742)

[Hình 11.4: Test case 2 – B11. 40](#_Toc152156742)

[Hình 12. 1 Test case 1 bài 12. 44](#_Toc152156924)

[Hình 12. 2.Test case 2 bài 12. 44](#_Toc152156925)

[Hình 12. 3.Test case 3 bài 12. 45](#_Toc152156926)

[Hình 12. 4.Test case 4 bài 12. 45](#_Toc152156927)

[Hình 12. 5 Test case 5 bài 12. 46](#_Toc152156928)

[Hình 12. 6 Test case 6 bài 12. 46](#_Toc152156929)

[Hình 12. 7.Test case 7 bài 12 47](#_Toc152156930)

[Hình 12. 8.Test case 8 bài 12. 47](#_Toc152156931)

[Hình 12. 9.Code bài 12. 48](#_Toc152156931)

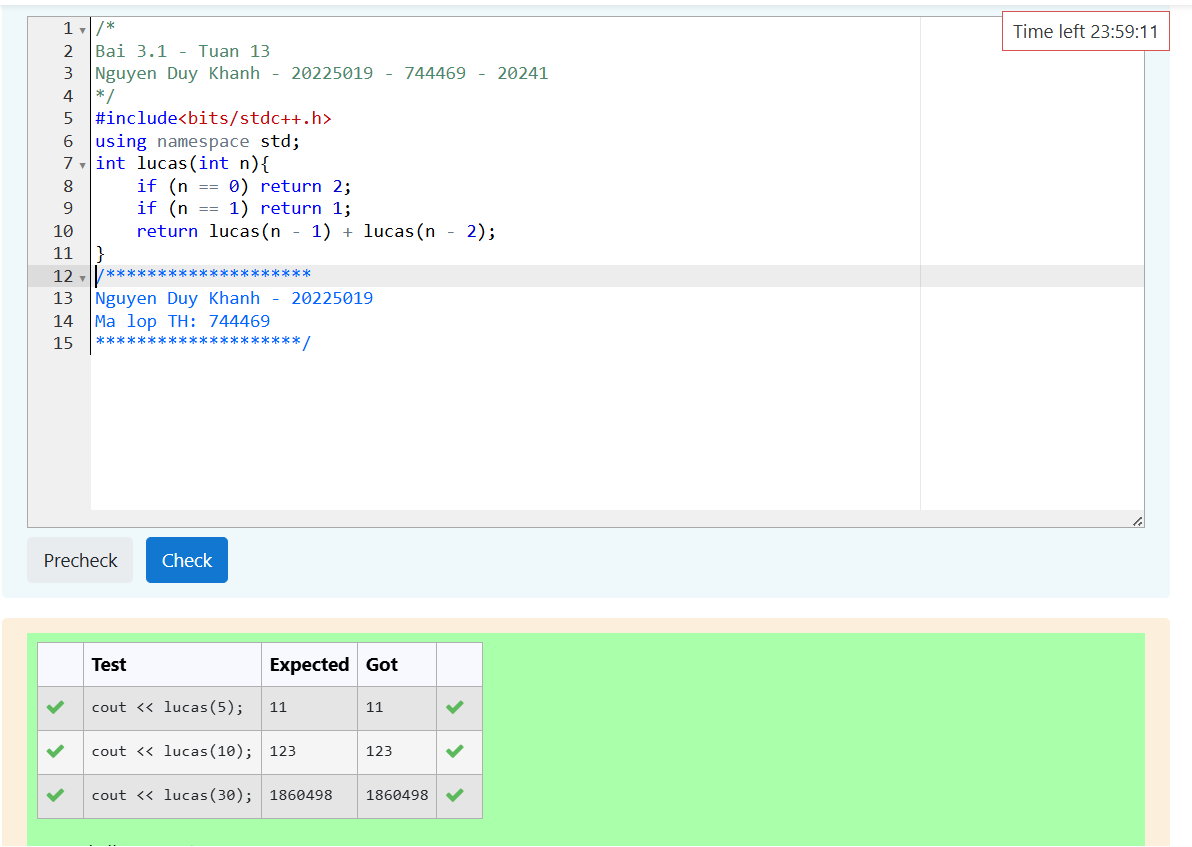
[Hình 12. 10.Code bài 12. 48](#_Toc152156931)

# 

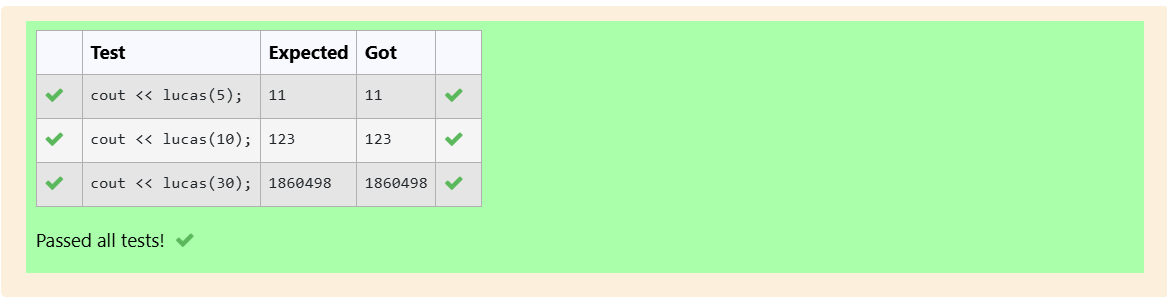
# Bài thực hành số 3 – Tuần 12

# Bài tập 1: Tính dãy Lucas

Dãy Lucas được định nghĩa bởi Ln=Ln−1+Ln−2 và bắt đầu bởi L0=2, L1=1. Viết hàm tính số Lucas thứ n.



Hình 1.1: Code bài 1.



Hình 1.2: Test case bài 1.

/\*

Bai 3.1 - Tuan 13

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<bits/stdc++.h>

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(){

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

std::cout << lucas(5) << std::endl;

std::cout << lucas(10) << std::endl;

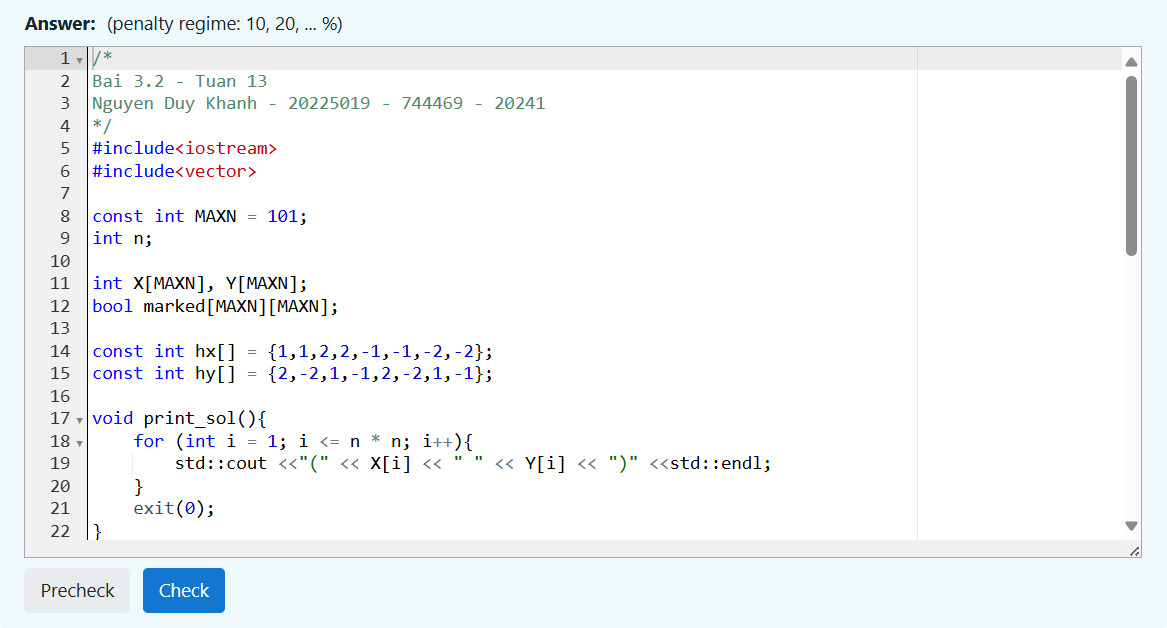
std::cout << lucas(30) << std::endl;

return 0;

}

# Bài tập 2: Quân Mã:

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).



Hình 2.1: Code bài 2

A screenshot of a computer program

Description automatically generated

Hình 2.2: Code bài 2

A screenshot of a computer code

Description automatically generated

Hình 2.3: Code bài 2

A screenshot of a computer

Description automatically generated

Hình 2.4: Test case bài 2

A screenshot of a computer

Description automatically generated

Hình 2.5: Test case bài 2

/\*

Bai 3.2 - Tuan 13

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<iostream>

#include<vector>

const int MAXN = 101;

int n;

int X[MAXN], Y[MAXN];

bool marked[MAXN][MAXN];

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

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

void print\_sol(){

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

std::cout <<"(" << X[i] << " " << Y[i] << ")" <<std::endl;

}

exit(0);

}

void Try(int k);

void Try(int k){

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

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

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

if (newX >= 1 && newX <= n && newY >= 1 && newY <= n && !marked[newX][newY]){

X[k] = newX;

Y[k] = newY;

marked[newX][newY] = true;

if (k == n \* n) print\_sol();

else {

Try(k + 1);

}

marked[newX][newY] = false;

}

}

}

int main(){

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

std::cin >> n;

marked[1][1] = true;

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

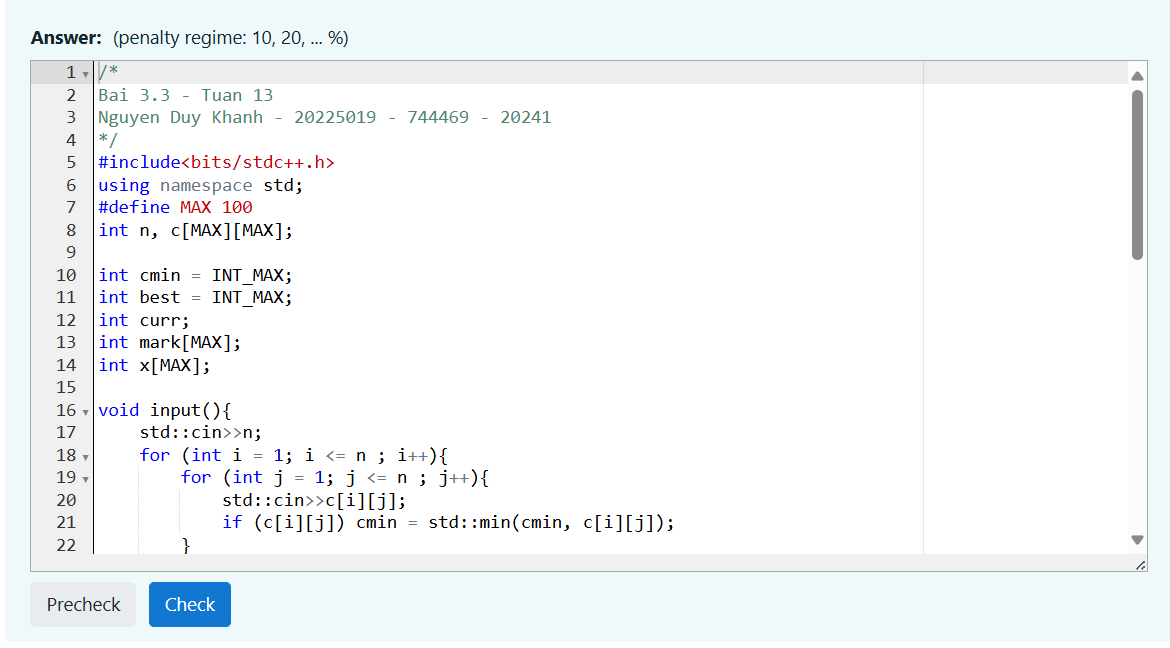
Try(2);

return 0;

}

# Bài tập 3: Bài toán người du lịch

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 ci,j. Hãy tìm tổng chi phí nhỏ nhất có thể



Hình 3.1: Code bài 3

A screenshot of a computer code

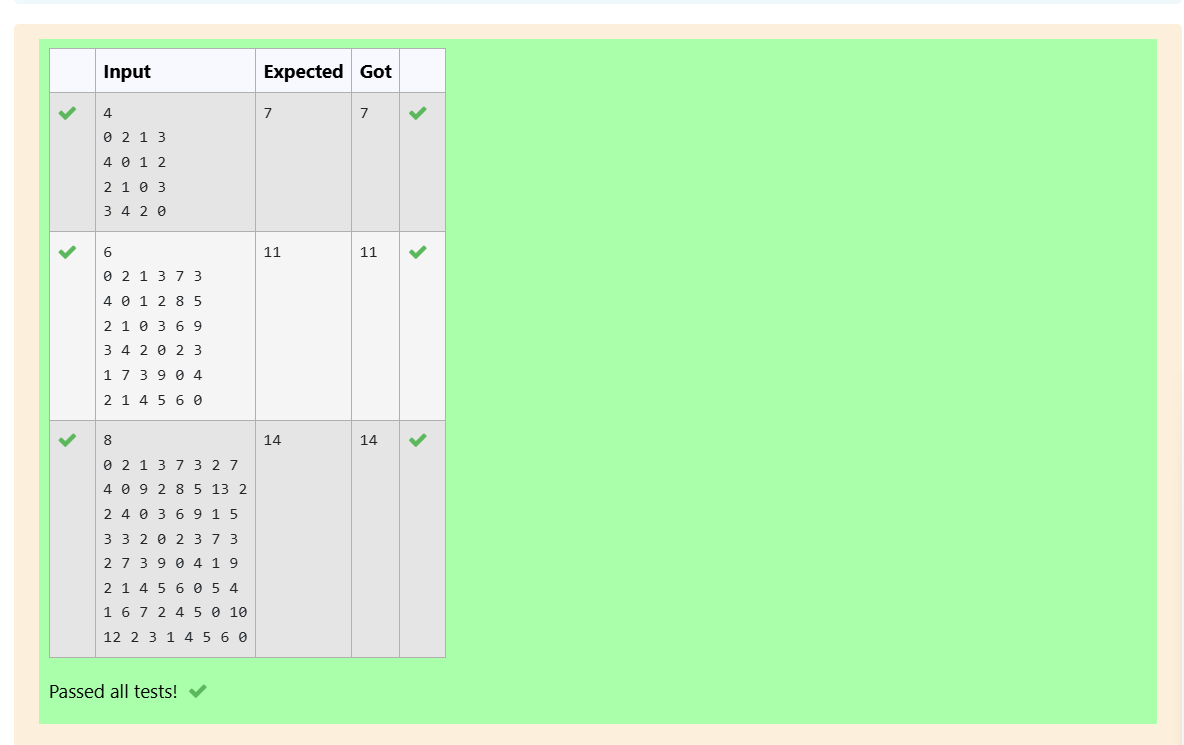
Description automatically generated

Hình 3.2: Code bài 31

A screenshot of a computer program

Description automatically generated

Hình 3.3: Code bài 3



Hình 3.4: Test case bài 3

/\*

Bai 3.3 - Tuan 13

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<iostream>

#include<cstring>

#include<climits>

using namespace std;

#define MAX 100

int n, c[MAX][MAX];

int cmin = INT\_MAX;

int best = INT\_MAX;

int curr;

int mark[MAX];

int x[MAX];

void input(){

std::cin>>n;

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

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

std::cin >> c[i][j];

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

}

}

}

void Try(int k){

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

if (!mark[i]){

mark[i] = 1;

x[k] = i;

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

if (k == n){

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

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

}

}

else {

if (curr > best) return;

else Try(k + 1);

}

mark[i] = 0;

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

}

}

}

int main(){

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

input();

x[1] = 1;

Try(2);

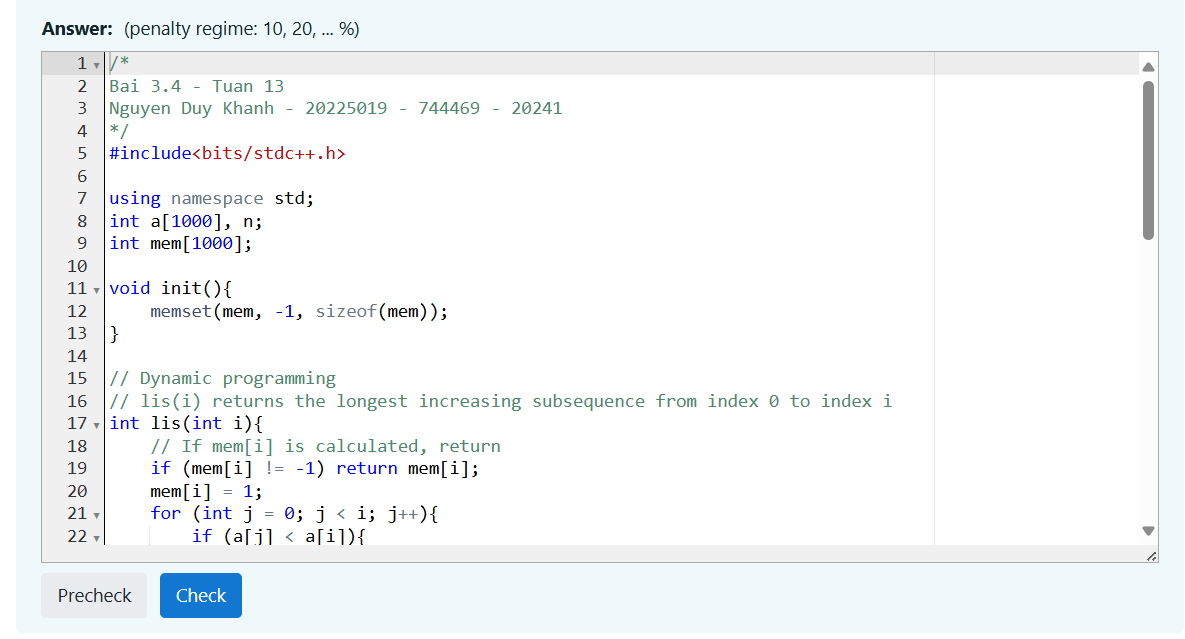
std::cout<<best;

return 0;

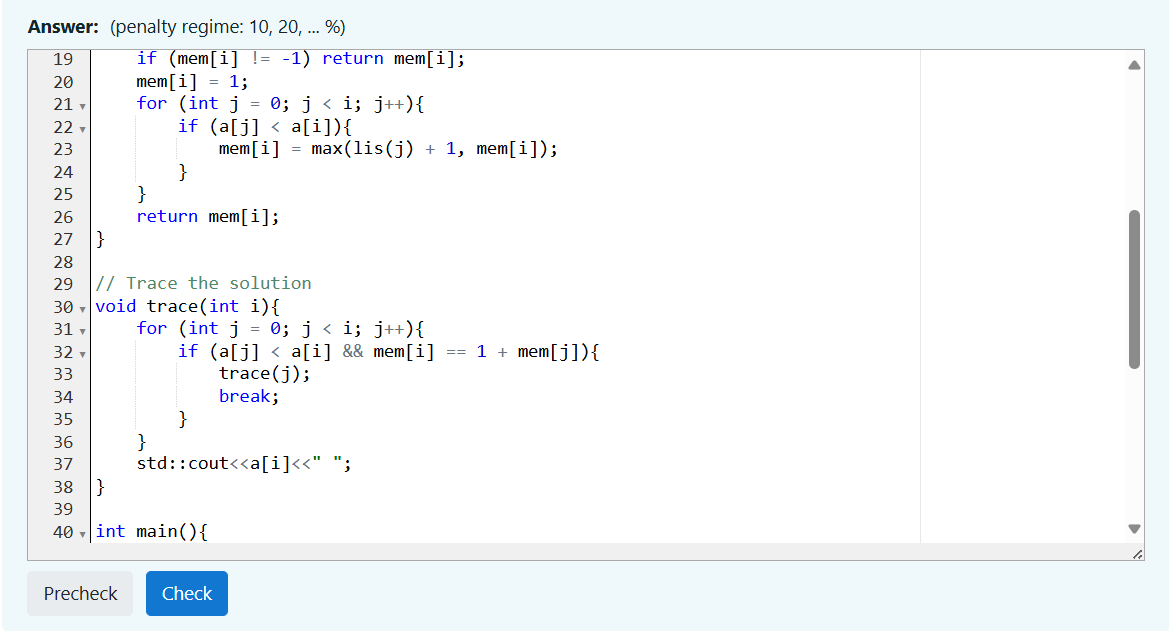
}

# Bài tập 4: LIS

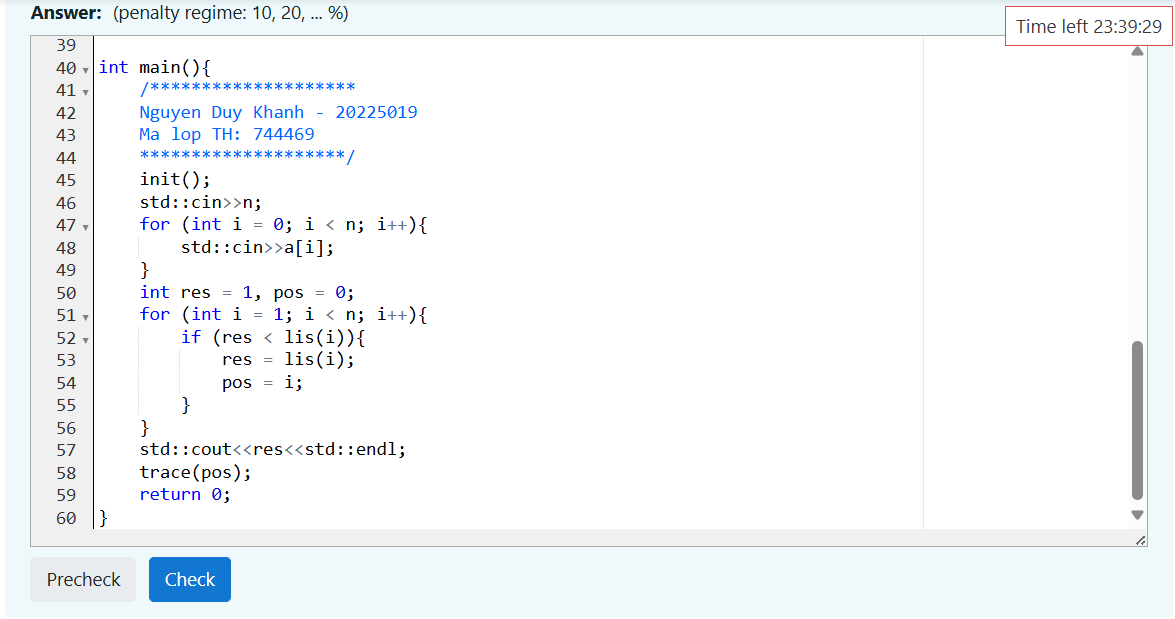
Cho dãy a có n phần tử. Một dãy con của a là dãy thu được bằng cách xóa đi một số phần tử của a và giữ nguyên thứ tự các phần tử còn lại (có thể không xóa phần tử nào). Hãy tìm dãy con tăng dài nhất của a



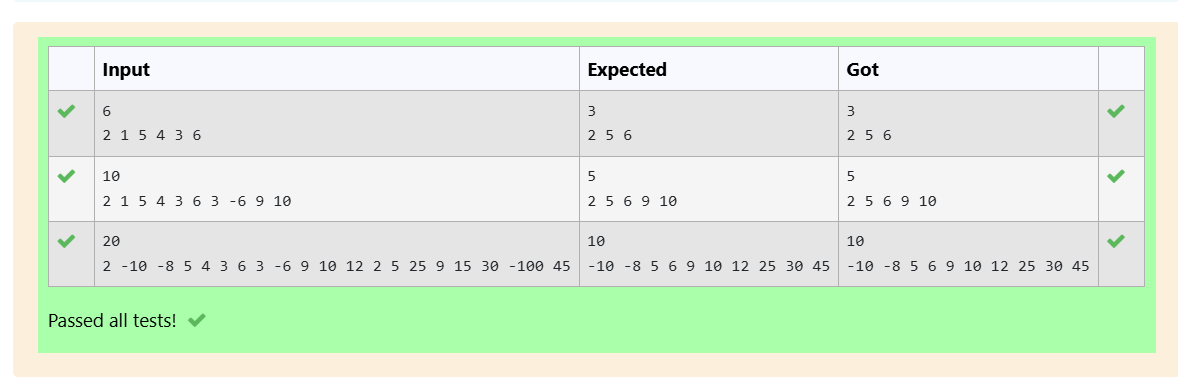
Hình 4. 1. Code bài 4.



Hình 4.2: Code bài 4



Hình 4.3: Code bài 4



Hình 4.4: Test case bài 4

/\*

Bai 3.4 - Tuan 13

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<bits/stdc++.h>

using namespace std;

int a[1000], n;

int mem[1000];

void init(){

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

}

// Dynamic programming

// lis(i) returns the longest increasing subsequence from index 0 to index i

int lis(int i){

// If mem[i] is calculated, return

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

}

}

return mem[i];

}

// Trace the solution

void trace(int i){

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

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

trace(j);

break;

}

}

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

}

int main(){

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

init();

std::cin>>n;

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

std::cin>>a[i];

}

int res = 1, pos = 0;

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

if (res < lis(i)){

res = lis(i);

pos = i;

}

}

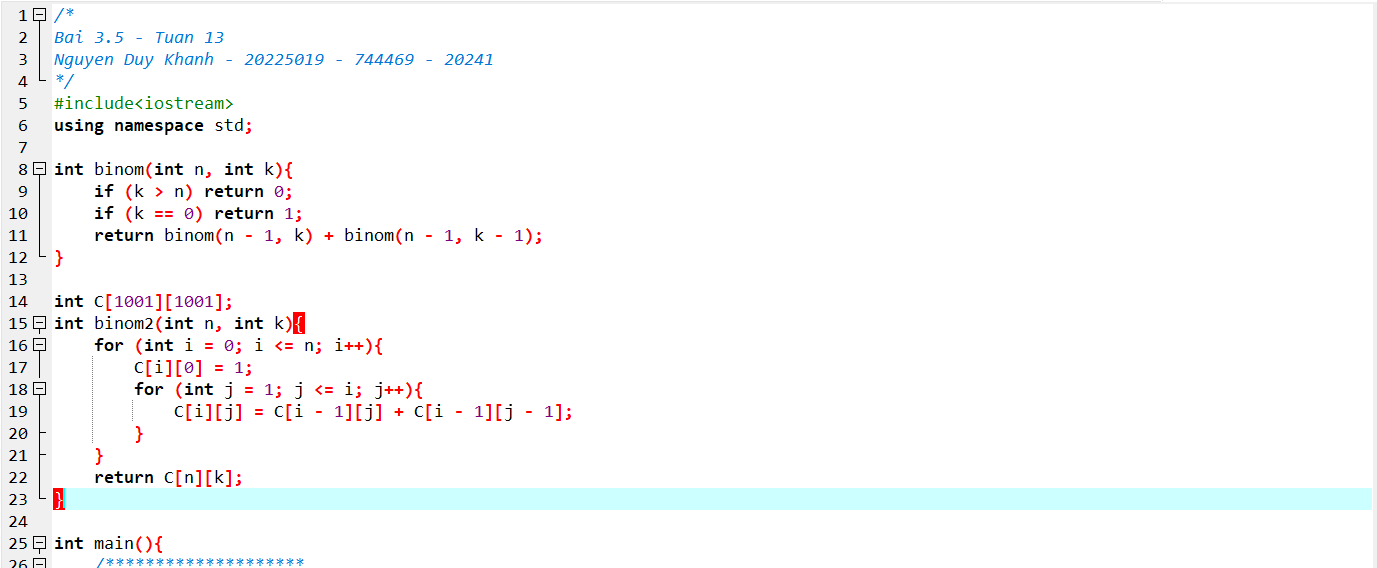
std::cout<<res<<std::endl;

trace(pos);

return 0;

}

# Bài tập 5: Tính tổ hợp



Hình 5.1: Code bài 5

A screenshot of a computer

Description automatically generated

Hình 5.2: Code bài 5

A screenshot of a computer

Description automatically generated

Hình 5.3: Test case 1

A computer screen shot of a number

Description automatically generated

Hình 5.4: Test Case 2

/\*

Bai 3.5 - Tuan 13

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#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 C[1001][1001];

int binom2(int n, int k){

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

C[i][0] = 1;

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

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

}

}

return C[n][k];

}

int main(){

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int m;

std::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));

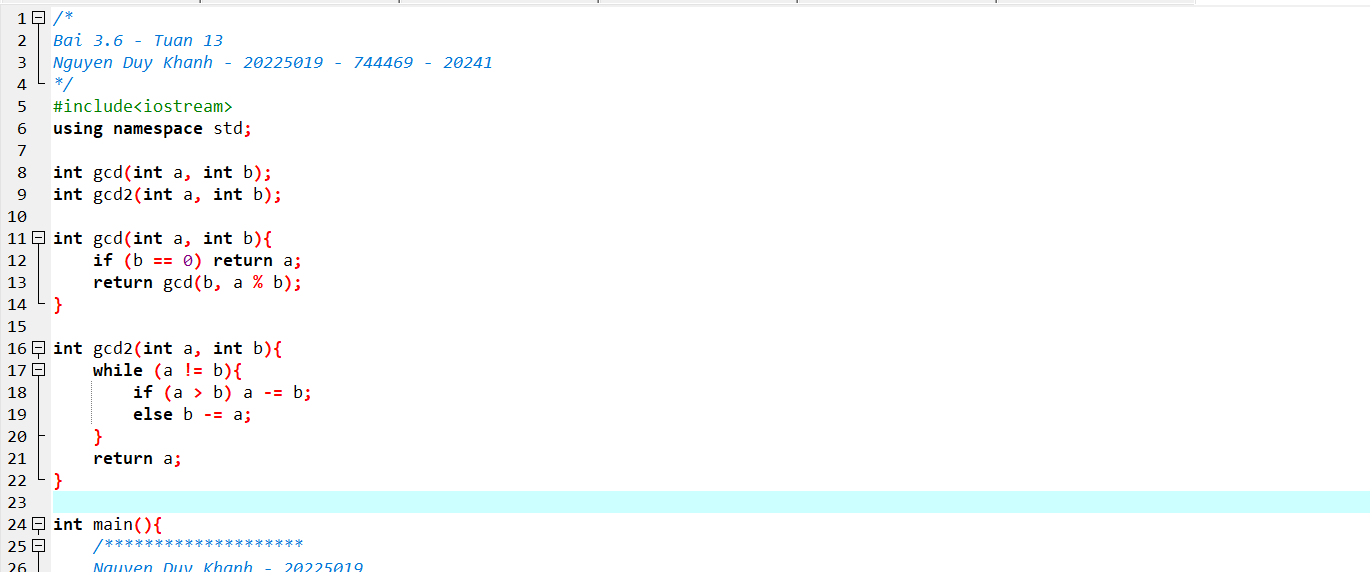
}

printf("\n");

}

}

# Bài tập 6: Tìm ước chung lớn nhất



Hình 6.1: Code bài 6

A screenshot of a computer

Description automatically generated

Hình 6.2: Code bài 6

A green rectangle with a white border

Description automatically generated

Hình 6.3: Test case bài 6

/\*

Bai 3.6 - Tuan 13

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<iostream>

using namespace std;

int gcd(int a, int b);

int gcd2(int a, int b);

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){

if (a > b) a -= b;

else b -= a;

}

return a;

}

int main(){

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int a, b;

std::cin >> a >> b;

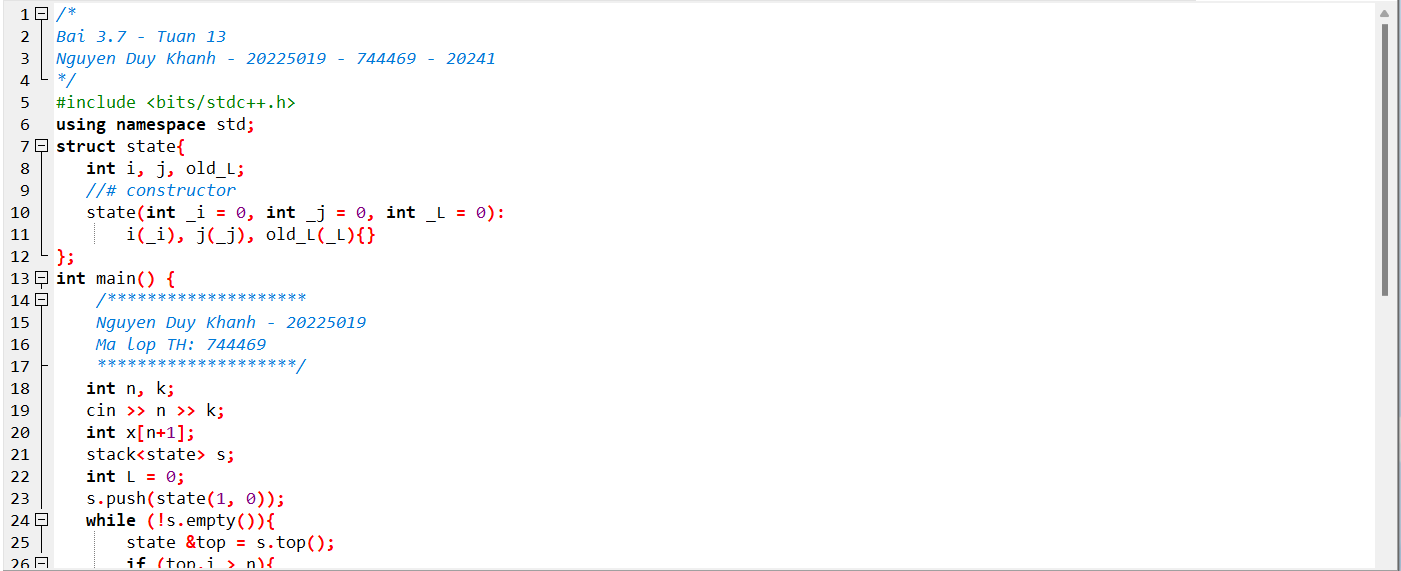
std::cout<<gcd(a, b)<<std::endl;

std::cout<<gcd2(a, b)<<std::endl;

}

# Bài tập 7: Liệt kê xâu nhị phân

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

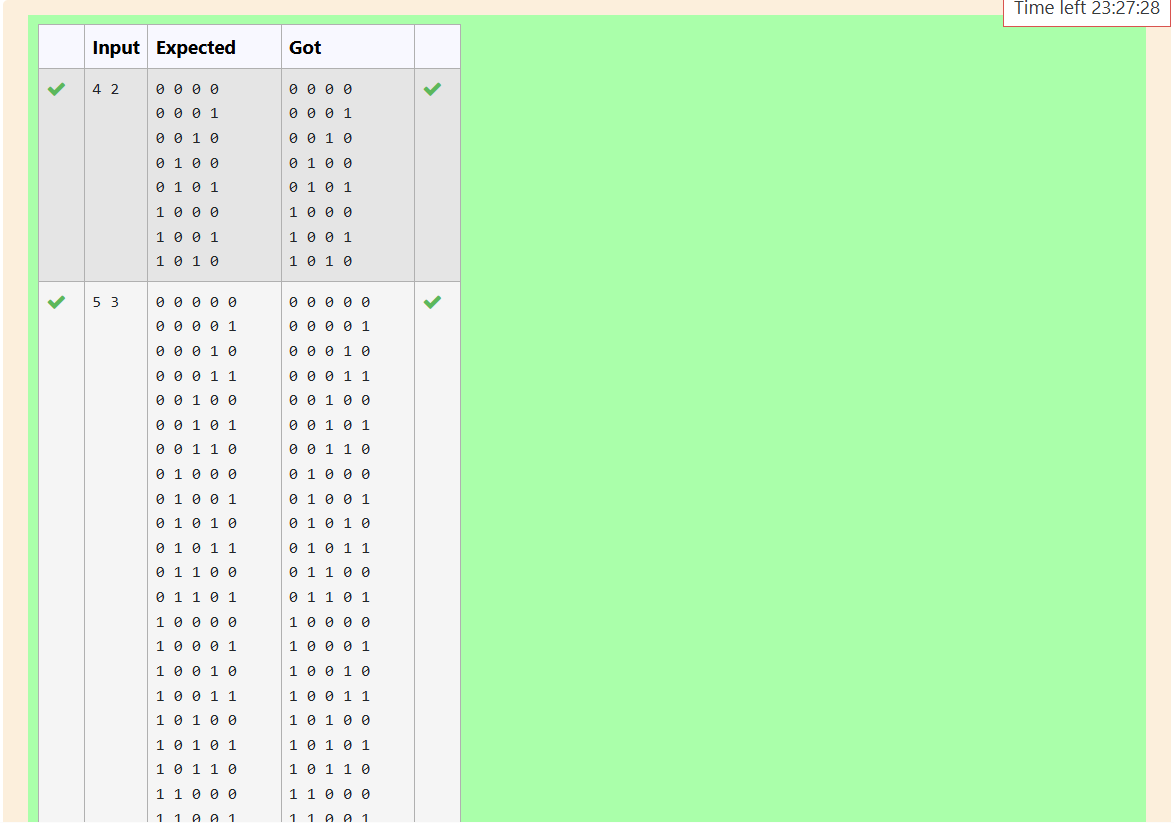


Hình 7.1. Code bài 7.

A white background with red and blue text

Description automatically generated

Hình 7.2: Code bài 7



Hình 7.3: Test case bài 7.

/\*

Bai 3.7 - Tuan 13

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#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() {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int n, k;

cin >> n >> k;

int x[n+1];

stack<state> s;

int L = 0;

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

while (!s.empty()){

state &top = s.top();

if (top.i > n){

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

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

s.pop();

continue;

}

if(top.j > 0)

L = top.old\_L;

if(top.j >1){

s.pop();

continue;

}

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;

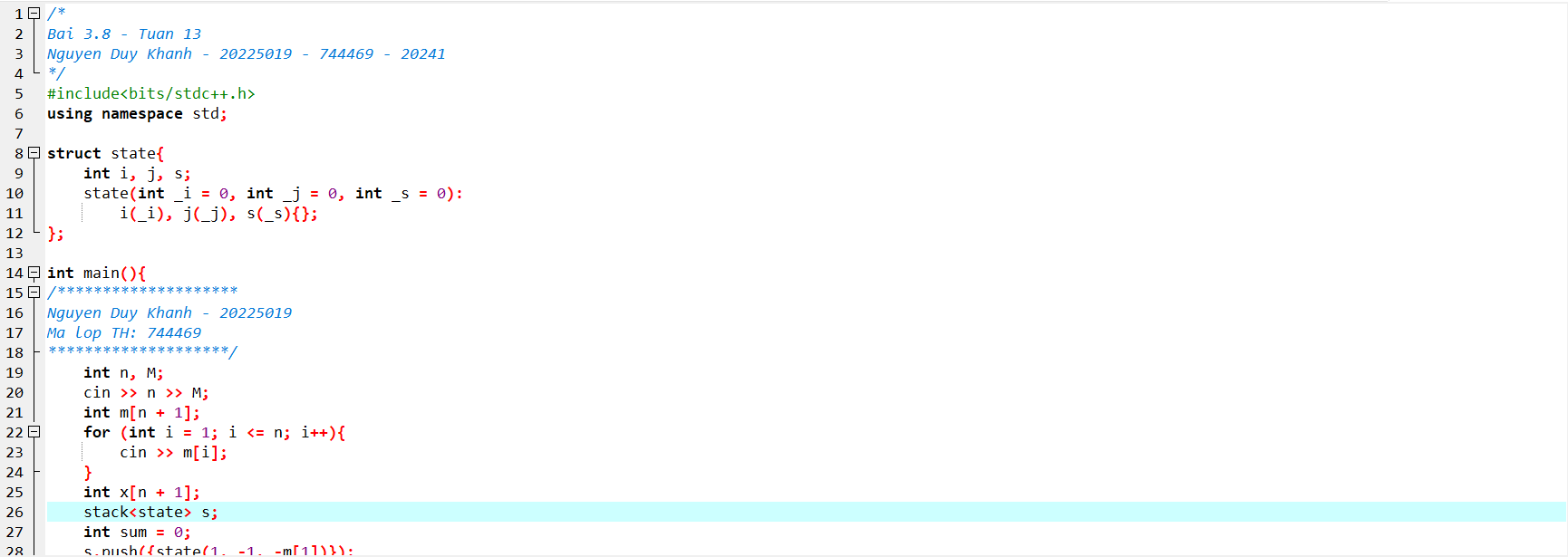
}

return 0;

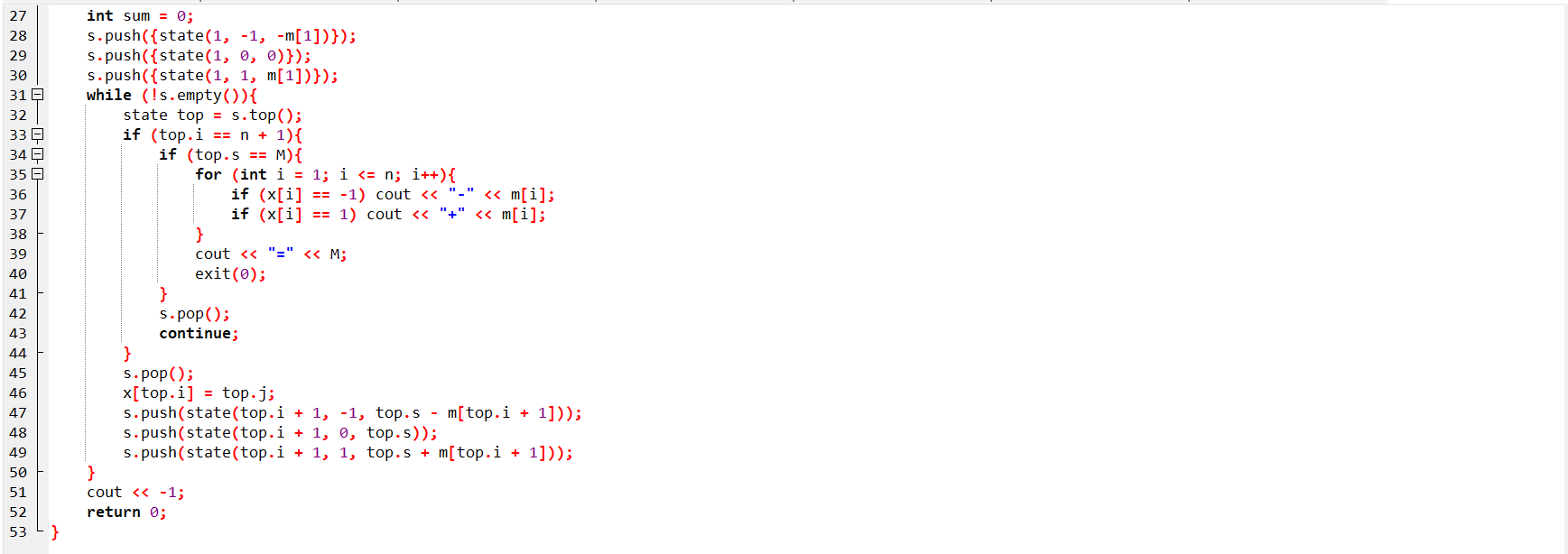
}

# Bài tập 8: Cân đĩa

Bạn đang muốn kiểm tra xem một vật cho trước có đúng nặng M như người ta nói hay không. Có một cân thăng bằng và n quả cân. Quả thứ i nặng mi. Hãy chỉ ra một cách cân thỏa mãn. Quy cách in ra đã được tích hợp trong mã nguồn dưới.



Hình 8.1: Code bài 8



Hình 8.2: Code bài 8

A black screen with white text

Description automatically generated

Hình 8.3: Test case bài 8.

/\*

Bai 3.8 - Tuan 13

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<bits/stdc++.h>

using namespace std;

struct state{

int i, j, s;

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

i(\_i), j(\_j), s(\_s){};

};

int main(){

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int n, M;

cin >> n >> M;

int m[n + 1];

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

cin >> m[i];

}

int x[n + 1];

stack<state> s;

int sum = 0;

s.push({state(1, -1, -m[1])});

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

s.push({state(1, 1, m[1])});

while (!s.empty()){

state top = s.top();

if (top.i == n + 1){

if (top.s == M){

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

if (x[i] == -1) cout << "-" << m[i];

if (x[i] == 1) cout << "+" << m[i];

}

cout << "=" << M;

exit(0);

}

s.pop();

continue;

}

s.pop();

x[top.i] = top.j;

s.push(state(top.i + 1, -1, top.s - m[top.i + 1]));

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

s.push(state(top.i + 1, 1, top.s + m[top.i + 1]));

}

cout << -1;

return 0;

}

# Bài tập 9: Lập lịch cho y tá

Một y tá cần lập lịch làm việc trong N ngày, mỗi ngày chỉ có thể là làm việc hay nghỉ ngơi. Một lịch làm việc là tốt nếu không có hai ngày nghỉ nào liên tiếp và mọi chuỗi ngày tối đại làm việc liên tiếp đều có số ngày thuộc đoạn [K1,K2][1,2]. Hãy liệt kê tất cả các cách lập lịch tốt, với mỗi lịch in ra trên một dòng một xâu nhị phân độ dài n với bit 0/1 tương ứng là nghỉ/làm việc. Các xâu phải được in ra theo thứ tự từ điển

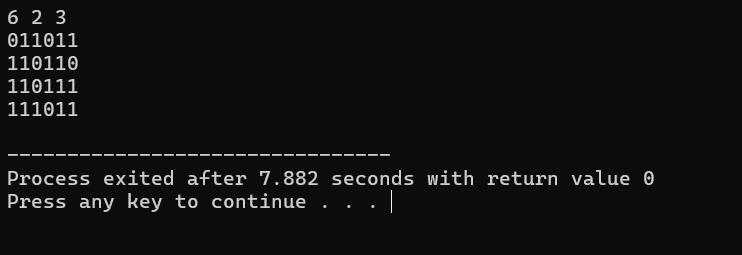
A screenshot of a computer

Description automatically generated

Hình 9.1: Code bài 9



Hình 9.2: Code bài 9

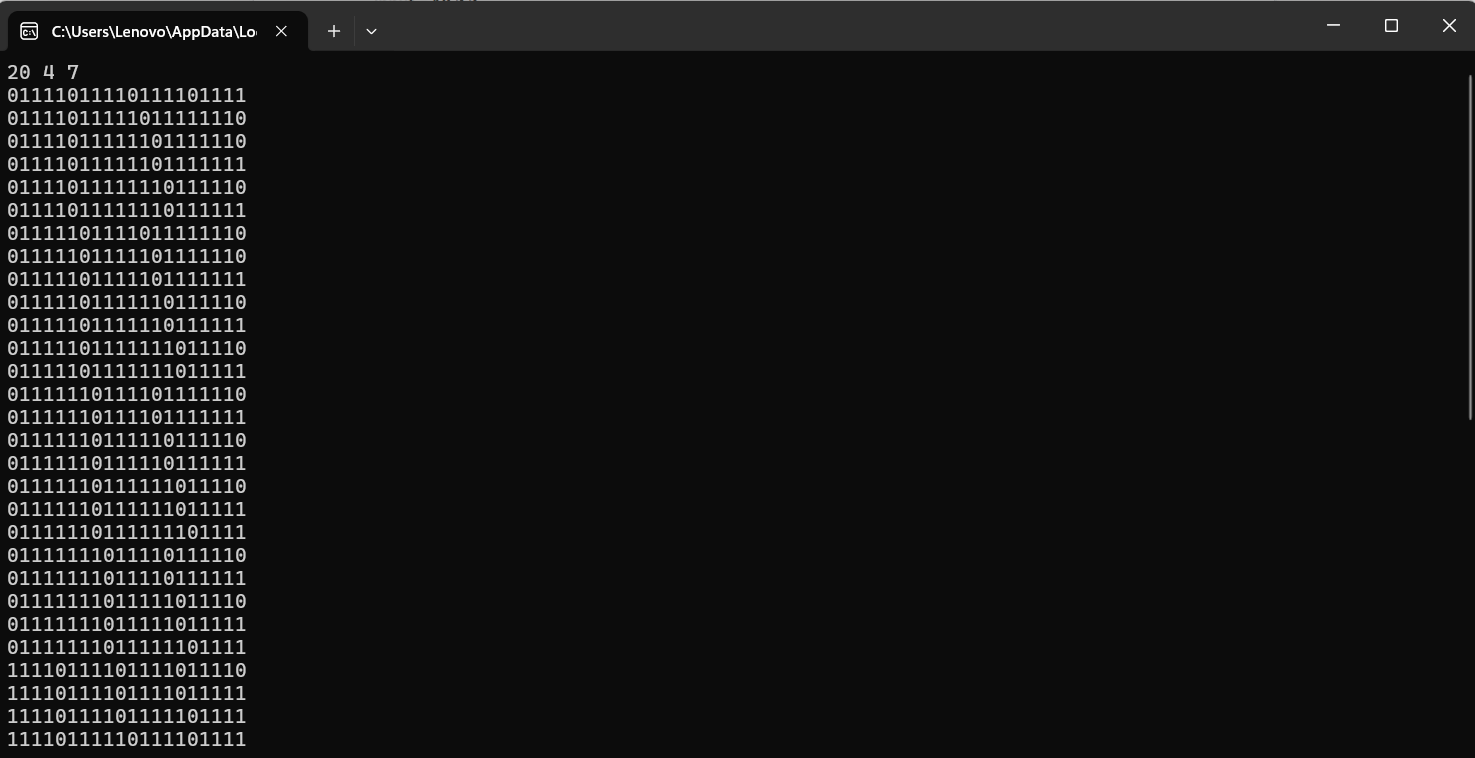


Hình 9.3: Test case 1 bài 9

A screen shot of a computer

Description automatically generated

Hình 9.4: Test case 2 bài 9.



Hình 9.5: Test case 3 bài 9.

/\*

Bai 3.9 - Tuan 13

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<bits/stdc++.h>

using namespace std;

const int MAX = 1000;

const int du = 1000000000 + 7;

int n, k1, k2;

int x[MAX];

int cnt = 0;

int so0 = 0, so1 = 0;

void inputData(){

cin >> n >> k1 >> k2;

}

bool check(int a, int i){

if(a==1) return true;

else {

if(i==0){

if(x[a-1] == 0) return false;

else {

if(so1<k1) return false;

}

} else {

if(x[a-1] == 0){

if(n-a+1 < k1) return false;

} else {

if(so1>=k2) return false;

}

}

return true;

}

}

void solution(){

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

cout << x[i];

cout << endl;

cnt++;

}

void TRY(int a){

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

if(check(a,i)){

x[a] = i;

int pre = so1;

if(i == 1){

if(x[a-1] == 1) so1++;

else so1 = 1;

} else {

so1 = 0;

}

if(a==n) solution();

else TRY(a+1);

so1 = pre;

}

}

}

int main(){

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

inputData();

TRY(1);

return 0;

}

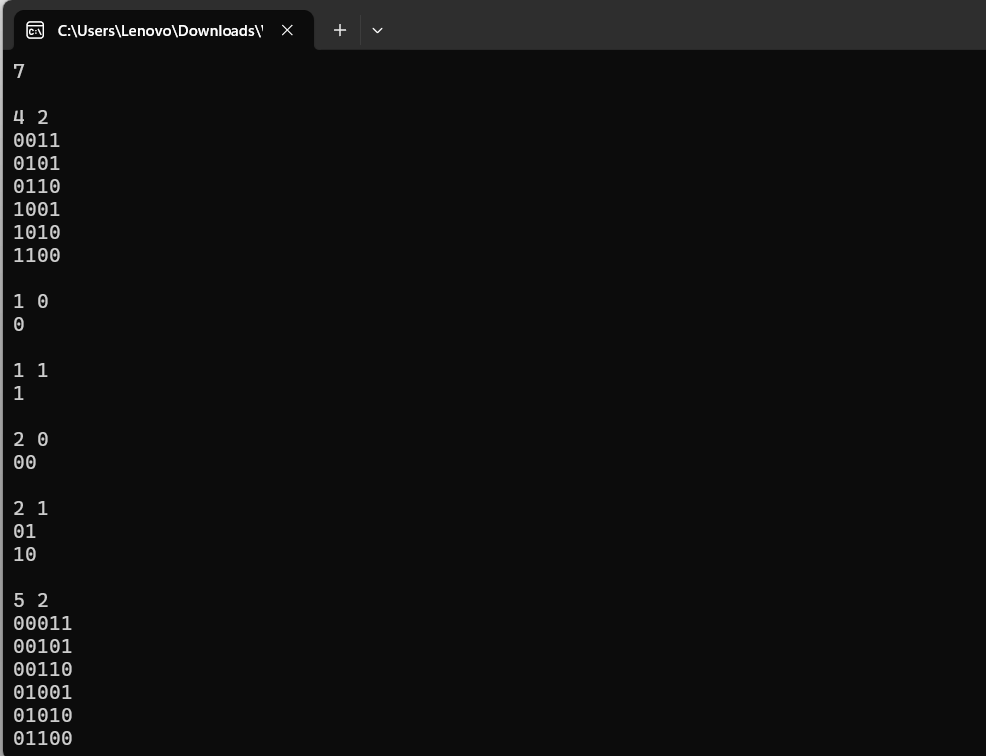
# Bài tập 10: Khoảng cách Hamming

Khoảng cách Hamming giữa hai xâu cùng độ dài là số vị trí mà ký tự tại vị trí đó là khác nhau trên hai xâu. Cho S là xâu gồm n ký tự 0. Hãy liệt kê tất cả các xâu nhị phân độ dài n, có khoảng cách Hamming với S bằng H. Các xâu phải được liệt kê theo thứ tự từ điển.

A white background with black and white clouds

Description automatically generated with medium confidence

Hình 10.1: Code bài 10



Hình 10.2: Test case bài 10 – Ảnh 1

A screenshot of a computer

Description automatically generated

Hình 10.3: Test Case Bài 10 - Ảnh 2

# Bài tập 11: Lịch trình chụp ảnh

Superior là một hòn đảo tuyệt đẹp với n địa điểm chụp ảnh và các đường một chiều nối các điểm chụp ảnh với nhau. Đoàn khách tham quan có r người với sở thích chụp ảnh khác nhau. Theo đó, mỗi người sẽ đưa ra danh sách các địa điểm mà họ muốn chụp. Bạn cần giúp mỗi người trong đoàn lập lịch di chuyển sao cho đi qua các điểm họ yêu cầu đúng một lần, không đi qua điểm nào khác, bắt đầu tại điểm đầu tiên và kết thúc tại điểm cuối cùng trong danh sách mà họ đưa ra, và có tổng khoảng cách đi lại là nhỏ nhất.

A white background with colorful text

Description automatically generated

Hình 11.1: Code bài 11

A screenshot of a computer

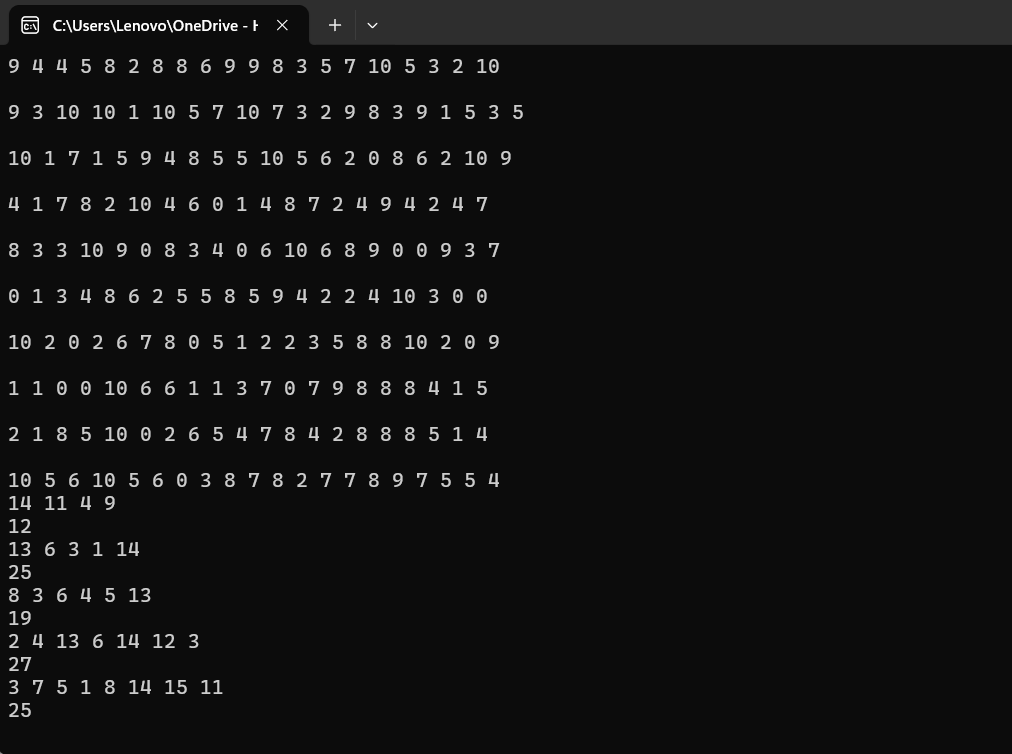
Description automatically generated

Hình 11.2: Code bài 11

A computer screen with white text

Description automatically generated

Hình 11.3: Test case 1 – B11



Hình 11.4: Test case 2 – B11

/\*

Bai 3.11 - Tuan 13

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<bits/stdc++.h>

using namespace std;

const int MAX = 10000;

int n, r;

int price[MAX][MAX];

int x[MAX];

bool visited[MAX];

vector<int> vt;

int best, sum\_price, start, des, numOfPoint;

void inputData(){

cin >> n >> r;

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

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

cin >> price[i][j];

}

}

}

bool check(int a, int i){

if(visited[vt[i]]) return false;

if(price[x[a-1]][vt[i]] == 0) return false;

return true;

}

void solution(){

if (price[x[numOfPoint-2]][des] == 0) return;

best = min(best, sum\_price + price[x[numOfPoint-2]][des]);

}

void TRY(int k){

for (int i = 1; i < numOfPoint - 1; i++){

if (check(k, i)){

visited[vt[i]] = true;

sum\_price += price[x[k-1]][vt[i]];

x[k] = vt[i];

if (k == numOfPoint - 2) solution();

else TRY(k+1);

visited[vt[i]] = false;

sum\_price -= price[x[k-1]][vt[i]];

}

}

}

int main(){

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Nguyen Duy Khanh - 20225019

Ma lop TH: 744469

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

string str;

inputData();

getline(cin,str);

while(r > 0){

best = INT\_MAX;

sum\_price = 0;

getline(cin, str);

while (!str.empty()){

stringstream convert(str.substr(0, str.find(" ")));

int tmp = 0;

convert >> tmp;

vt.push\_back(tmp - 1);

if (str.find(" ") > str.size()) break;

else str.erase(0, str.find(" ") + 1);

}

start = vt[0];

des = vt[vt.size() - 1];

numOfPoint = vt.size();

x[0] = start; x[numOfPoint-1] = des;

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

visited[i] = false;

}

TRY(1);

if (best == INT\_MAX) cout << "0" << endl;

else cout << best << endl;

vt.erase(vt.begin(), vt.end());

r--;

}

}

# Bài tập 12: Đếm đường đi

Cho đồ thị vô hướng G, hãy đếm số đường đi đi qua k cạnh và không đi qua đỉnh nào quá một lần.

A screenshot of a computer

Description automatically generated

Hình 12.1: Test case 1 bài 12

A screenshot of a computer

Description automatically generated

Hình 12.2: Test case 2 bài 12

A screenshot of a computer

Description automatically generated

Hình 12.3: Test case 3 bài 12.

A screenshot of a computer

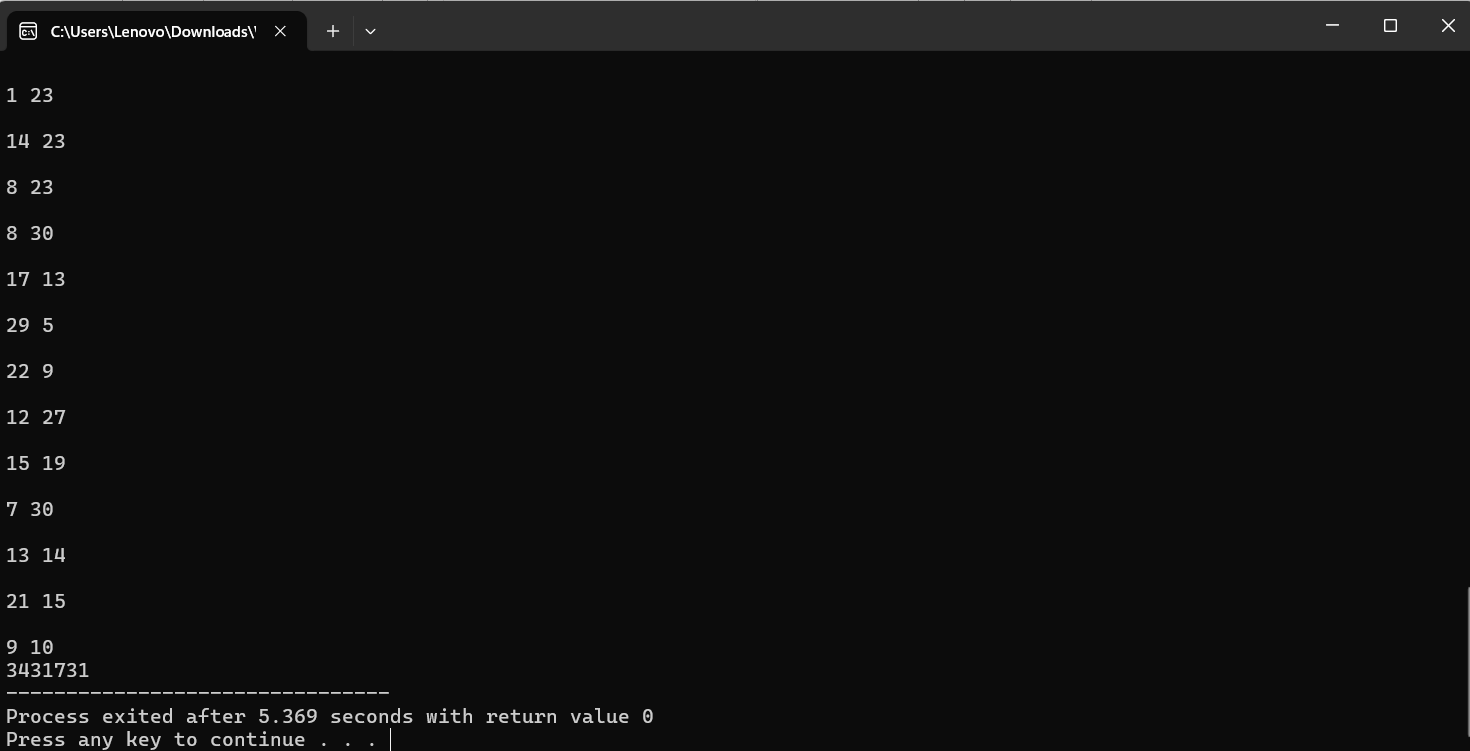
Description automatically generated

Hình 12.4: Test case 4 bài 12.

A screenshot of a computer

Description automatically generated

Hình 12.5: Test case 5 bài 12

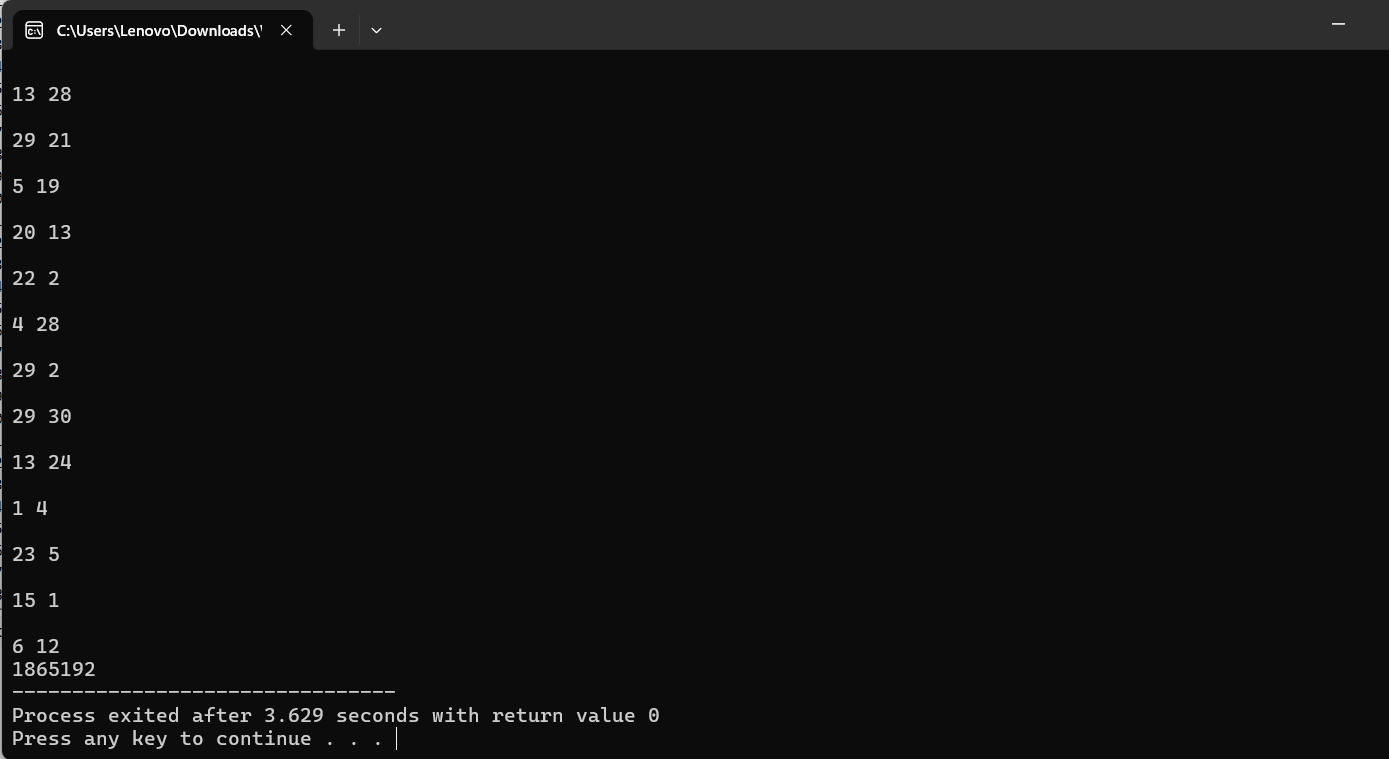


Hình 12.6: Test case 6 Bài 12

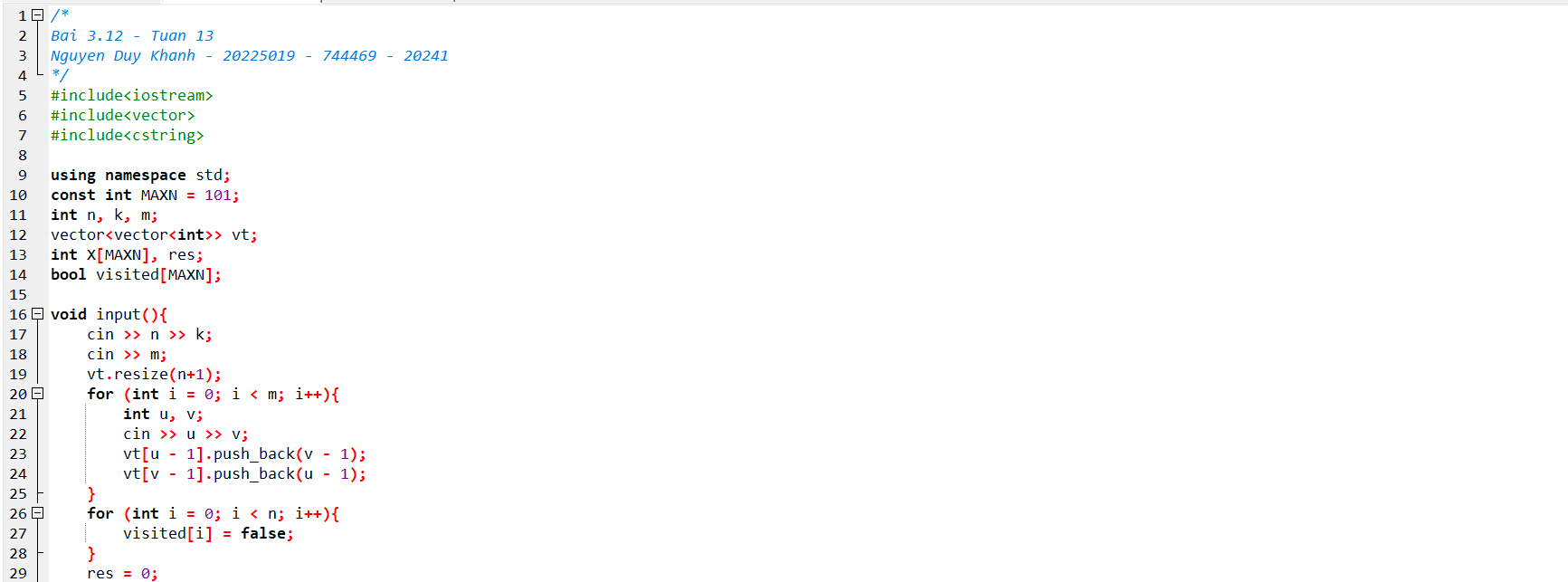
A screenshot of a computer

Description automatically generated

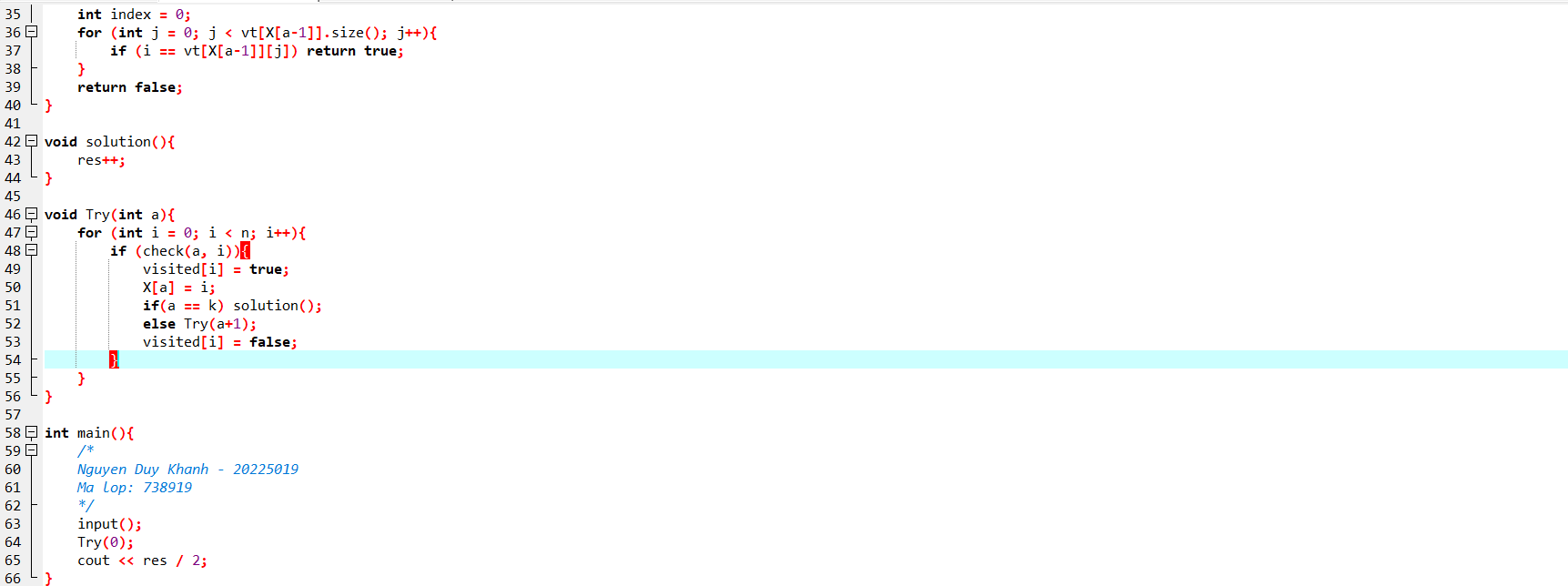
Hình 12.7: Test case 7 bài 12



Hình 12.8: Test case 8 bài 12



Hình 12.9: Code bài 12 – Ảnh 1



Hình 12.10: Code bài 12 – Ảnh 2

/\*

Bai 3.12 - Tuan 13

Nguyen Duy Khanh - 20225019 - 744469 - 20241

\*/

#include<iostream>

#include<vector>

#include<cstring>

using namespace std;

const int MAXN = 101;

int n, k, m;

vector<vector<int>> vt;

int X[MAXN], res;

bool visited[MAXN];

void input(){

cin >> n >> k;

cin >> m;

vt.resize(n+1);

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

int u, v;

cin >> u >> v;

vt[u - 1].push\_back(v - 1);

vt[v - 1].push\_back(u - 1);

}

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

visited[i] = false;

}

res = 0;

}

bool check(int a, int i){

if (a == 0) return true;

if (visited[i] == true) return false;

int index = 0;

for (int j = 0; j < vt[X[a-1]].size(); j++){

if (i == vt[X[a-1]][j]) return true;

}

return false;

}

void solution(){

res++;

}

void Try(int a){

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

if (check(a, i)){

visited[i] = true;

X[a] = i;

if(a == k) solution();

else Try(a+1);

visited[i] = false;

}

}

}

int main(){

/\*

Nguyen Duy Khanh - 20225019

Ma lop: 738919

\*/

input();

Try(0);

cout << res / 2;

}