# Bài thực hành 3: Đệ quy và khử đệ quy để giải quyết một số bài toán

Contents

[Bài thực hành 3: Đệ quy và khử đệ quy để giải quyết một số bài toán 1](#_Toc71543787)

[Phần 1. Thực hành về đệ quy 3](#_Toc71543788)

[1.1 Đệ quy - quay lui 3](#_Toc71543789)

[Bài tập 1: Tính dãy Lucas 3](#_Toc71543790)

[Dãy Lucas được định nghĩa bởi Ln=Ln−1+Ln−2Ln=Ln−1+Ln−2 và bắt đầu bởi L0=2L0=2, L1=1L1=1. Viết hàm tính số Lucas thứ nn. 3](#_Toc71543791)

[Bài tập 2: Quân mã đi tuần 4](#_Toc71543792)

[Trên bàn cờ vua kích thước n×nn×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) 4](#_Toc71543793)

[1.2 Kỹ thuật nhánh cận 6](#_Toc71543794)

[Bài tập 3: Bài toán người du lịch 6](#_Toc71543795)

[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 ii sang thành phố jj là ci,jci,j. Hãy tìm tổng chi phí nhỏ nhất có thể 6](#_Toc71543796)

[1.3 Đệ quy có nhớ 9](#_Toc71543797)

[Bài tập 4: LIS 9](#_Toc71543798)

[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 9](#_Toc71543799)

[Phần 2. Khử đệ quy 11](#_Toc71543800)

[Bài tập 5: Tính tổ hợp 11](#_Toc71543801)

[Tính Ckn 11](#_Toc71543802)

[Bài tập 6: Tìm ước chung lớn nhất 13](#_Toc71543803)

[Tính ước chung lớn nhất của hai số cho trước 13](#_Toc71543804)

[Bài tập 7: Liệt kê xâu nhị phân 15](#_Toc71543805)

[Sử dụng phương pháp khử đệ quy bằng stack, hãy liệt kê các xâu nhị phân độ dài nn không có kk bit 1 nào liên tiếp 15](#_Toc71543806)

[Bài tập 8: Cân đĩa 18](#_Toc71543807)

[Bạn đang muốn kiểm tra xem một vật cho trước có đúng nặng MM như người ta nói hay không. Có một cân thăng bằng và nn quả cân. Quả thứ ii nặng mimi. Hãy chỉ ra một cách cân thỏa mãn. Quy cách in ra đã được tích hợp trong mã nguồn dưới. 18](#_Toc71543808)

[Phần 3. Bài tập về nhà 20](#_Toc71543809)

[Bài tập 9: Lập lịch cho y tá 20](#_Toc71543810)

[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][K1,K2]. Hãy liệt kê tất cả các cách lập lịch tốt, với mỗi lịch in ra trên một dòng một xâu nhị phân độ dài nn với bit 0/1 tương ứng là nghỉ/làm việc. Các xâu phải được in ra theo thứ tự từ điển 20](#_Toc71543811)

[Bài tập 10: Khoảng cách Hamming 22](#_Toc71543812)

[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 SS là xâu gồm nn ký tự 0. Hãy liệt kê tất cả các xâu nhị phân độ dài nn, có khoảng cách Hamming với SS bằng HH. Các xâu phải được liệt kê theo thứ tự từ điển 22](#_Toc71543813)

[Bài tập 11: Lịch trình chụp ảnh 24](#_Toc71543814)

[Superior là một hòn đảo tuyệt đẹp với nn địa điểm chụp ảnh và các đường một chiều nối các điểm chụp ảnh với nhau. Đoàn khách tham quan có rr 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. 24](#_Toc71543815)

[Bài tập 12: Đếm đường đi 28](#_Toc71543816)

[Cho đồ thị vô hướng G, hãy đếm số đường đi đi qua k cạnh và không đi qua đỉnh nào quá một lần. 28](#_Toc71543817)

# Phần 1. Thực hành về đệ quy

## 1.1 Đệ quy - quay lui

### Bài tập 1: Tính dãy Lucas

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

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap1\_Buoi3.cpp

#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() {

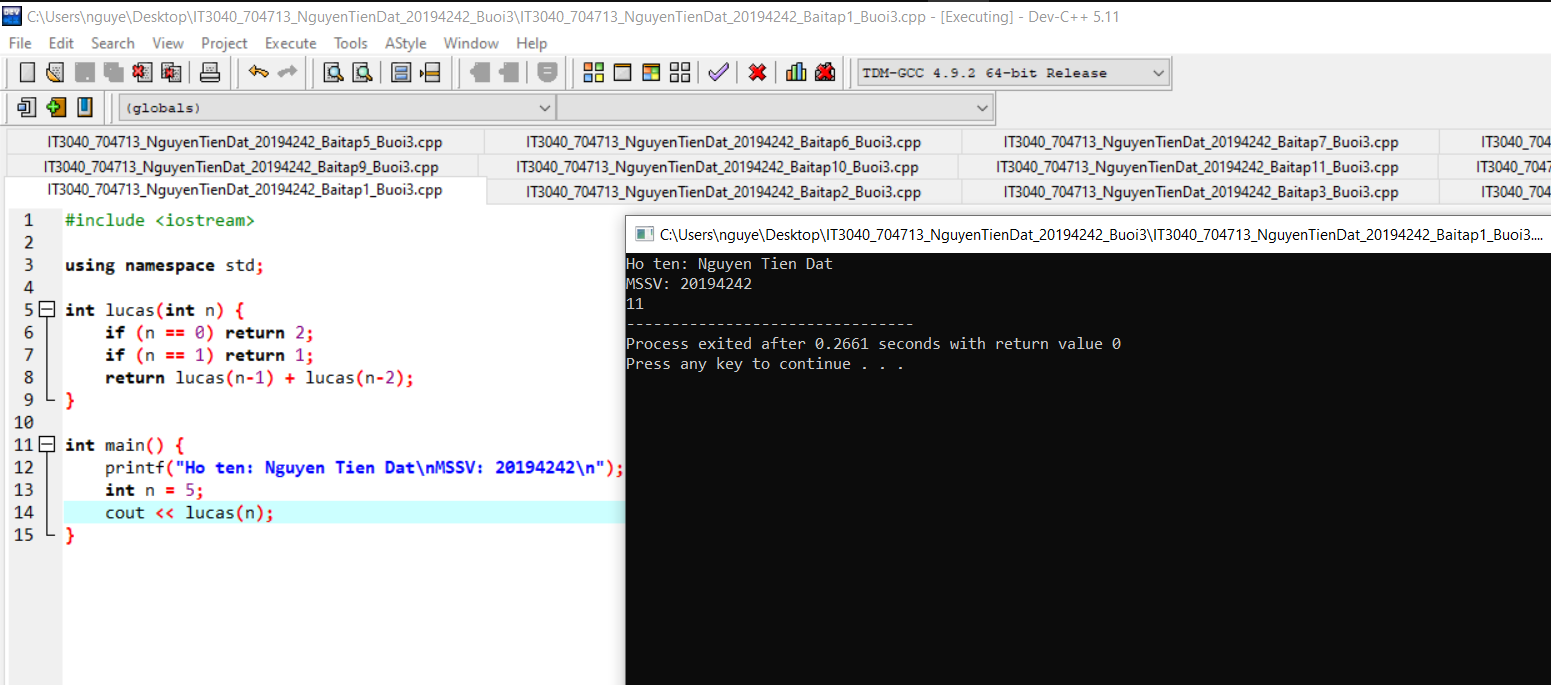
printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

int n = 5;

cout << lucas(n);

}

Kết quả:



### Bài tập 2: Quân mã đi tuần

### Trên bàn cờ vua kích thước n×nn×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)

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap2\_Buoi3.cpp

#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 isUCV(int xx, int yy) {

if (xx < 1 || xx > n || yy < 1 || yy > n || mark[xx][yy] == 1) return false;

return true;

}

//# Thu?t toán quay lui

void TRY(int k){

// chay tat ca loi giai bo phan co the xay ra

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

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

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

// code here

// check xem co phai UCV

if (isUCV(xx, yy)){

// cap nhat trang thai moi

X[k] = xx;

Y[k] = yy;

mark[xx][yy] = 1;

// neu hoan thanh 1 loi giai day du, buoc neo de quy

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

else TRY(k+1);

// sau khi hoan thanh 1 loi giai day du phai khoi phuc trang thai ban dau cua cac o

mark[xx][yy] = 0;

}

}

}

int main(){

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

cin >> n;

mark[1][1] = 1;

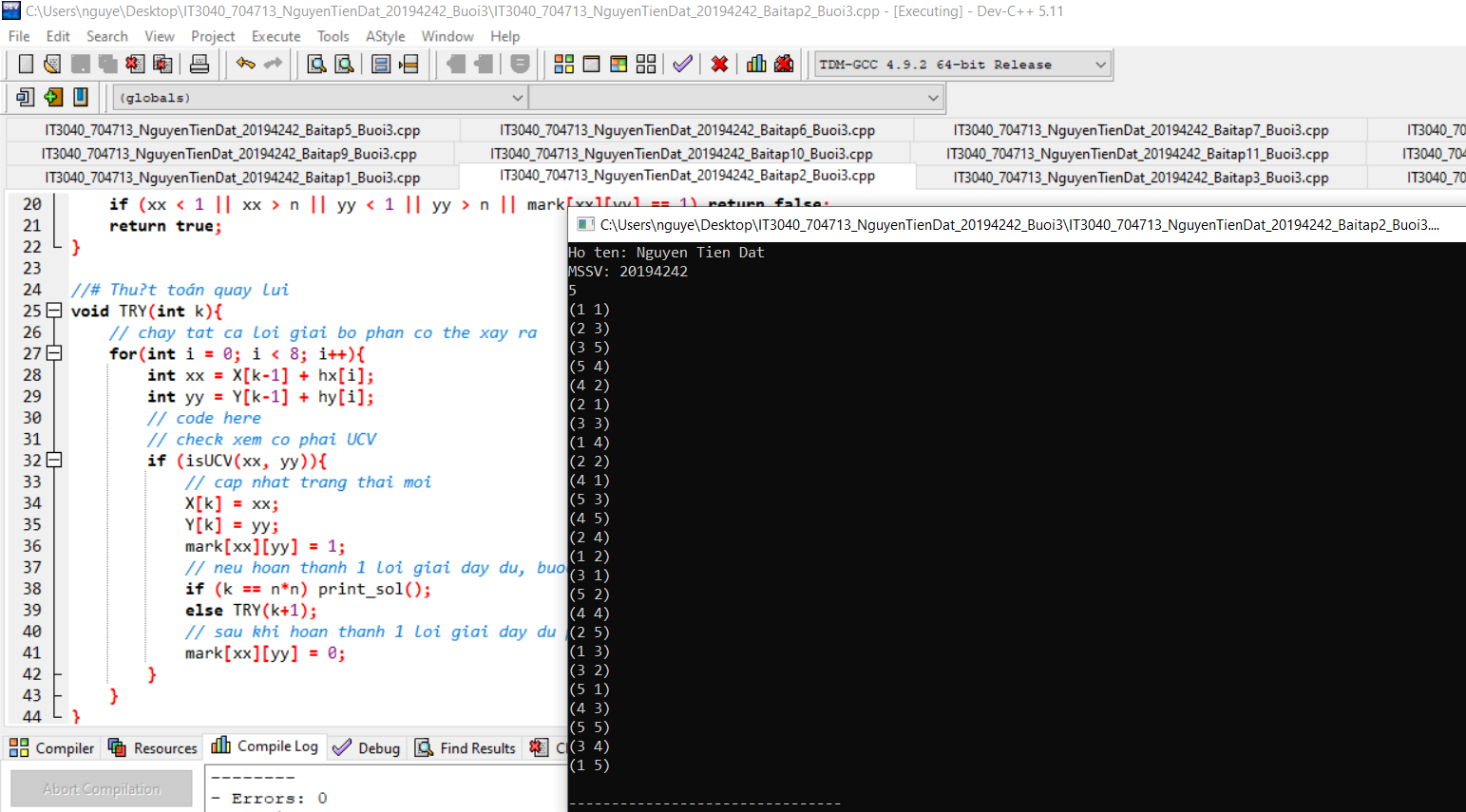
X[1] = Y[1] = 1;

TRY(2);

return 0;

}

Kết quả:



## 1.2 Kỹ thuật nhánh cận

### 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 ii sang thành phố jj là ci,jci,j. Hãy tìm tổng chi phí nhỏ nhất có thể

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap3\_Buoi3.cpp

#include <iostream>

using namespace std;

#define MAX 100

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

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

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 = 0; //# 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

//# Ð?c d? li?u vào

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]);

}

for(int i=0; i<MAX; i++) mark[i] = 0;

}

bool isUCV(int j) {

if (mark[j] == 1) return false;

return true;

}

//# Thu?t toán quay lui

void TRY(int k){

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

// code here

if (isUCV(i)) {

x[k] = i;

mark[i] = 1;

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

// loi giai bo phan cap cuoi,

// tinh them chi phi quay tro ve thanh pho 1

if (k == n) {

curr += c[x[k]][1]; // tong chi phi cuoi cung;

best = (best > curr ? curr : best);

curr -= c[x[k]][1]; // lay xog 1 nhanh, tro lai trang thai cu

}

else TRY(k+1);

// thuc hien xog 1 loi giai khoi phuc trang thai ban dau

mark[i] = 0;

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

}

}

}

int main() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

input();

x[1] = 1;

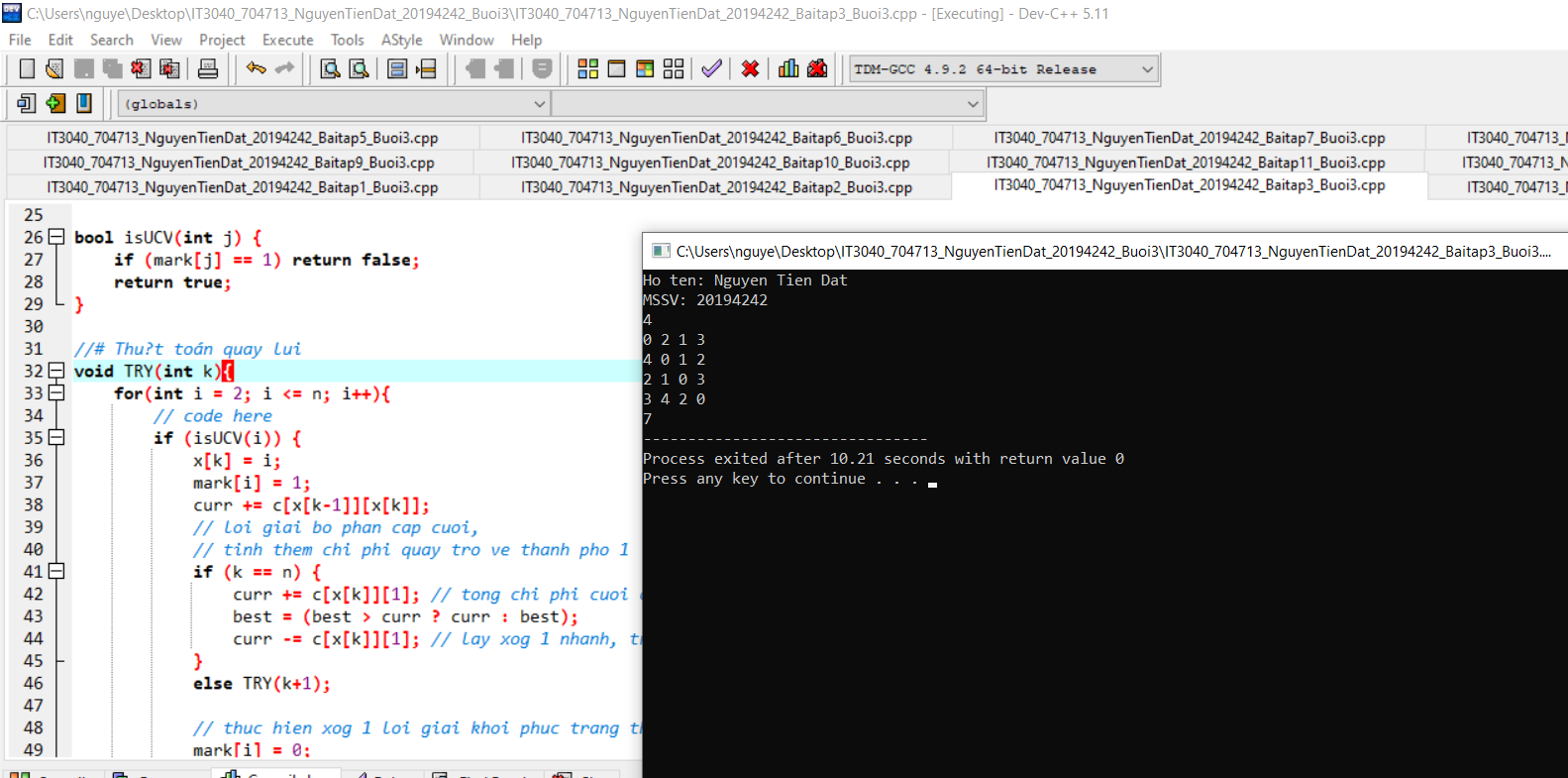
TRY(2);

cout << best;

return 0;

}

Kết quả:



## 1.3 Đệ quy có nhớ

### 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

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap4\_Buoi3.cpp

#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) {

// code here

int max\_length = 0;

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

if (a[j] < a[i] && mem[j] > max\_length) max\_length = mem[j];

}

mem[i] = max\_length + 1;

// mem[i] = [ max(0<=j<=i-1 && a[j]<=a[i]) (mem[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;

}

}

// in ra phan tu lon nhat cuoi cung cua day

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

}

int main(){

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

init();

cin >> n;

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

int res = 0, pos = 0;

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

// cout << lis(i) << ' ';

if (res < lis(i)){

res = lis(i);

pos = i; // pos la vi tri cuoi cung cua day con tang dai nhat

}

}

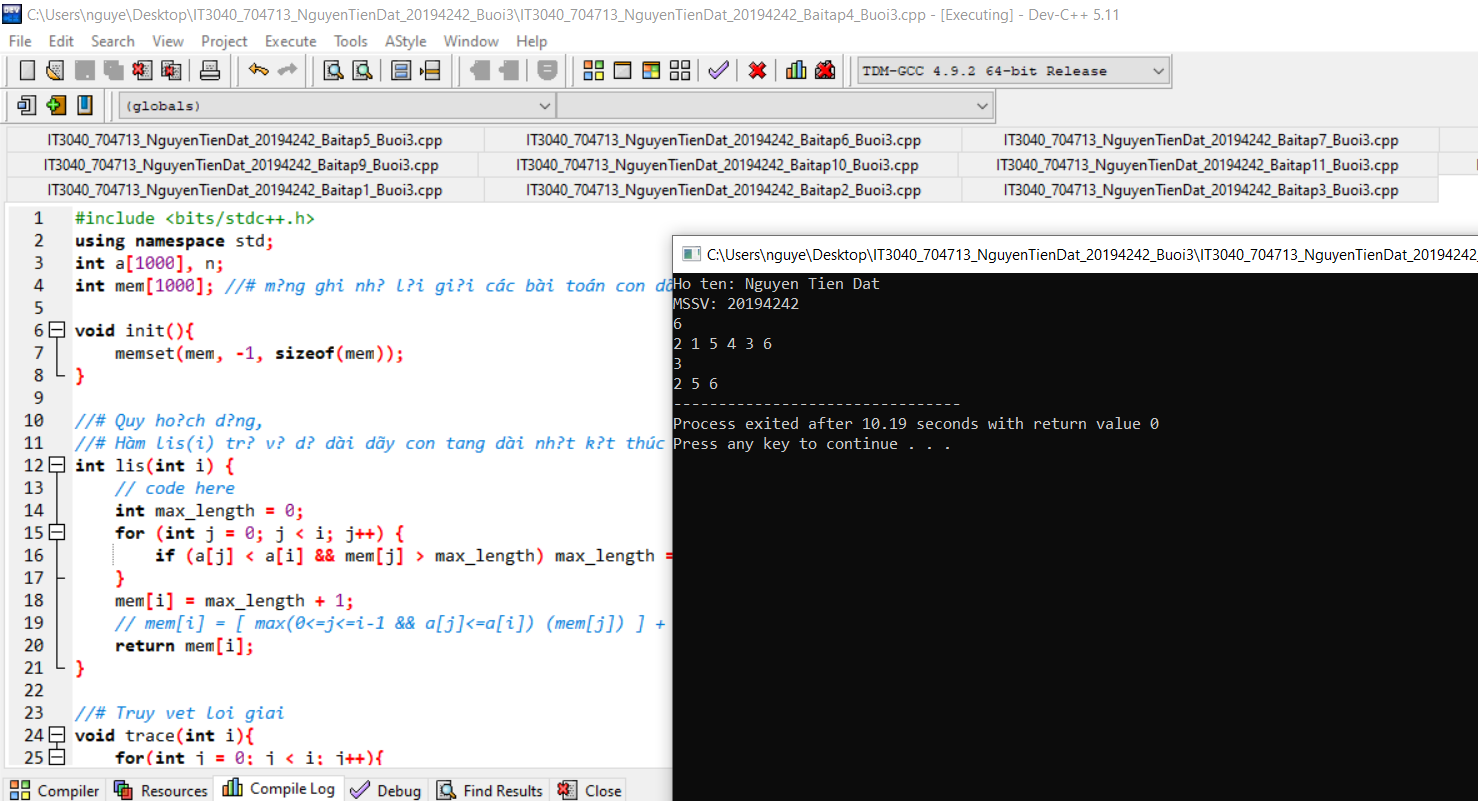
cout << res << endl;

trace(pos);

return 0;

}

Kết quả:



# Phần 2. Khử đệ quy

### Bài tập 5: Tính tổ hợp

### Tính Ckn

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap5\_Buoi3.cpp

#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){

//# Kh? d? quy

// code here

if (k > n) return 0;

int binom = 1;

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

binom = binom \* n / i;

n--;

}

return binom;

}

int main() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

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));

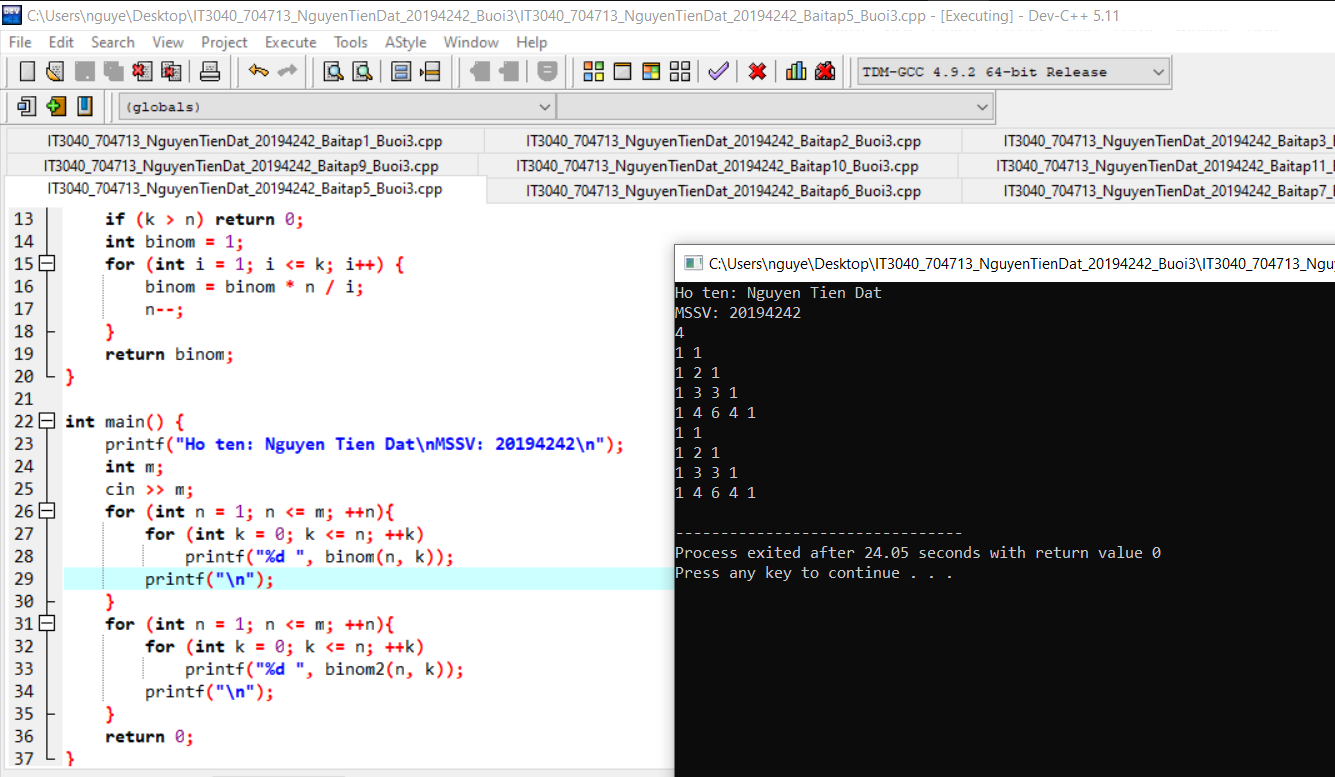
printf("\n");

}

return 0;

}

Kết quả:



### Bài tập 6: Tìm ước chung lớn nhất

### Tính ước chung lớn nhất của hai số cho trước

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap6\_Buoi3.cpp

#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){

//# Kh? d? quy

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

# YOUR CODE HERE #

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

int tmp;

while (b != 0) {

tmp = a % b;

a = b;

b = tmp;

// khi do a > b do b > tmp

}

return a;

}

int main() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

int a, b;

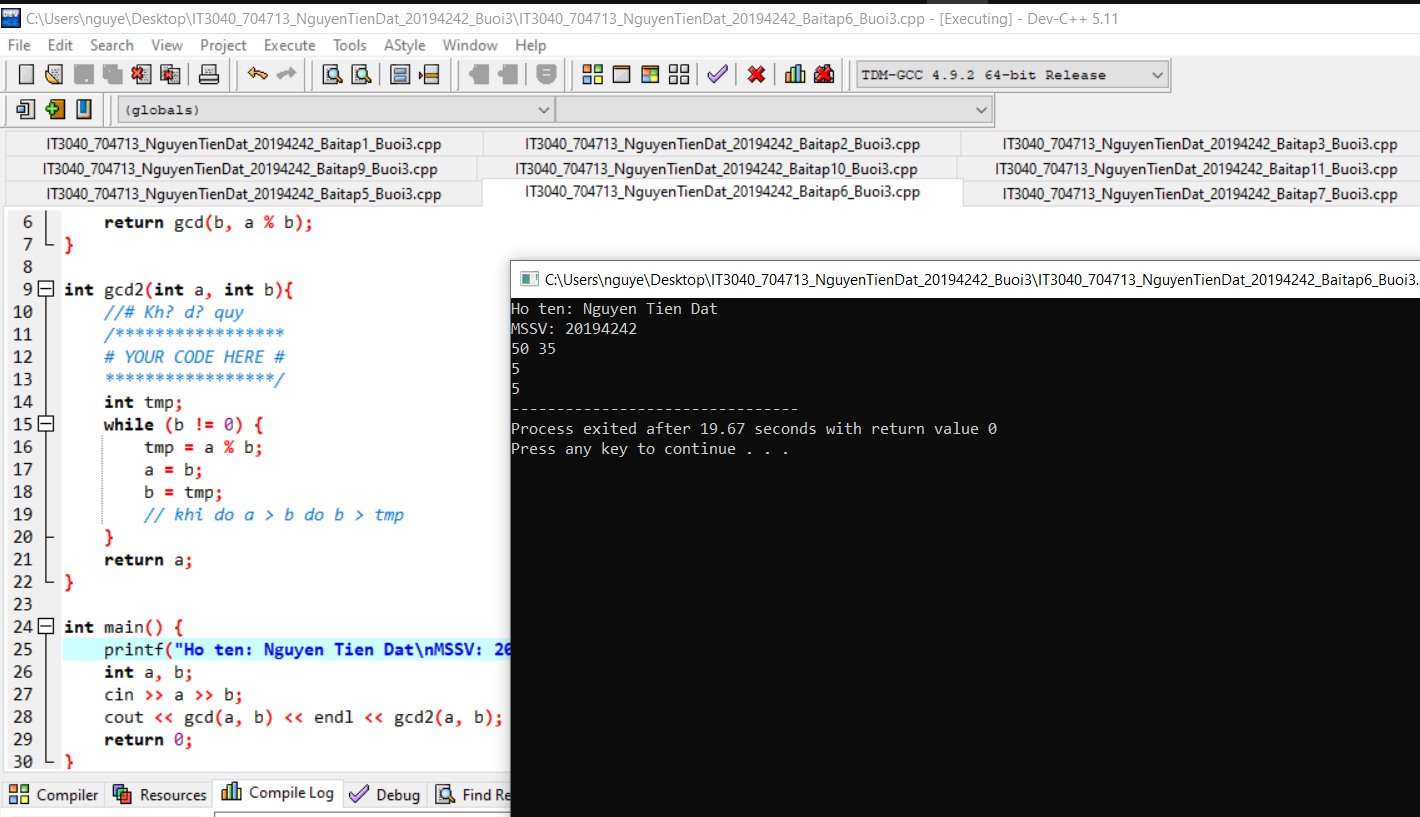
cin >> a >> b;

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

return 0;

}

Kết quả:



### 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 nn không có kk bit 1 nào liên tiếp

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap7\_Buoi3.cpp

#include <bits/stdc++.h>

using namespace std;

// Gi? s? l?i gi?i du?c luu b?i xâu x1, x2, ..., xn

// i : bieu dien loi giai bo phan cap i, truoc dó x1, x2, ..., x[i-1] dã duoc gán giá tr? => i bat dau tu 1

// j : giá tri ung cu viên dang xét cho vi trí x[i] // j la 0 hoac 1

// old\_L: so ký tu 1 liên tiep o cuoi dãy x1, x2, ..., x[i-1]

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() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

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, 0));

while (!s.empty()){

state &top = s.top();

// cout << top.i << top.j << top.old\_L << endl;

//# 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;

}

//# Kh? d? quy

// code here

if (top.j >= 1) L = top.old\_L;

if (top.j > 1) { // top.j chi co the = 0 hoac 1

s.pop();

continue;

}

// L la trang thai htai tu x[1]...x[top.i]

// top.old\_L la trang thai truoc cua L tu x[1]...x[top.i - 1]

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

x[top.i] = top.j;

top.old\_L = L;

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

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

}

top.j++;

// cout << "sau vong lap" << endl;

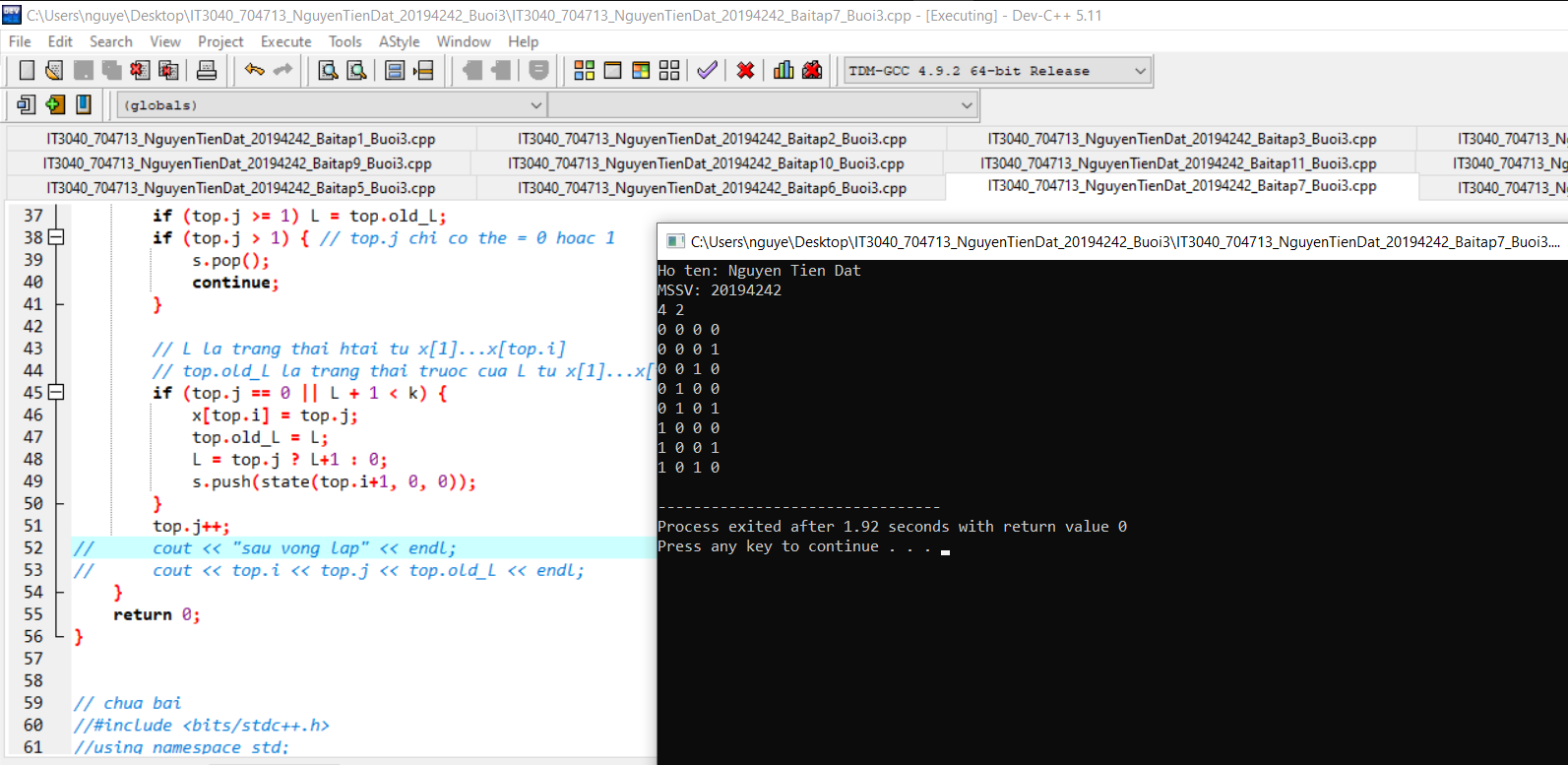
// cout << top.i << top.j << top.old\_L << endl;

}

return 0;

}

Kết quả:



### Bài tập 8: Cân đĩa

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

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap8\_Buoi3.cpp

#include <iostream>

using namespace std;

int M; // dat qua can ben phai

int n;

int m[1000];

int mark[1000]; // dau + hoac -

int curr = 0; // khoi luong dia can hien tai ben trai

int loi\_giai\_thu\_may = 0; // loi giai thu n, neu bang 0 => ko co cach can nao

void print\_sol() {

if (loi\_giai\_thu\_may == 1) cout << "Cac cach can:" << endl;

cout << "Loi giai thu " << loi\_giai\_thu\_may << ": ";

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

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

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

}

cout << "=" << M << endl;

}

bool isUCV(int k) {

if (k > n) return false;

return true;

}

void TRY(int k) {

if (isUCV(k)) {

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

mark[k] = j;

curr += m[k] \* mark[k];

if (curr == M) {

loi\_giai\_thu\_may++;

print\_sol();

}

else TRY(k+1);

curr -= m[k] \* mark[k];

mark[k] = 0;

}

}

}

int main() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

cin >> n >> M;

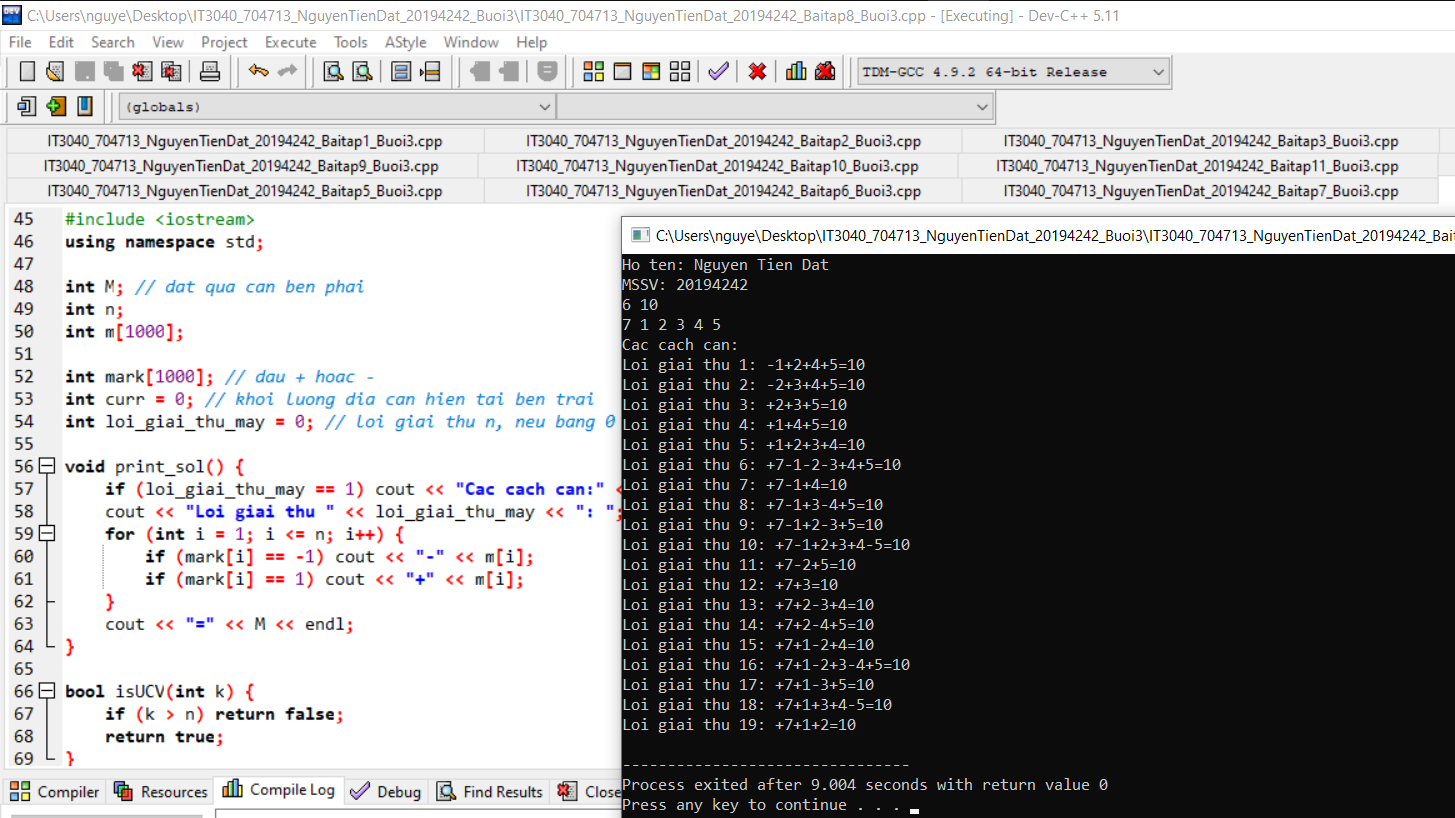
for (int i = 1; i <= n; ++i) cin >> m[i];

TRY(1);

if (loi\_giai\_thu\_may == 0) cout << "Khong co loi giai nao";

}

Kết quả:



# Phần 3. Bài tập về nhà

### 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][K1,K2]. Hãy liệt kê tất cả các cách lập lịch tốt, với mỗi lịch in ra trên một dòng một xâu nhị phân độ dài nn với bit 0/1 tương ứng là nghỉ/làm việc. Các xâu phải được in ra theo thứ tự từ điển

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap9\_Buoi3.cpp

#include <iostream>

using namespace std;

int n, k1, k2;

// so ngay lam viec lien tiep khi chua cap nhat loi giai thu k

int res[1000]; // loi giai

void print\_sol() {

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

cout << res[i];

}

cout << endl;

}

bool isUCV(int j, int k, int old\_state) {

if (j == 0 && res[k-1] == 0) return false;

if (j == 0 && old\_state < k1 && old\_state != 0) return false;

if (j == 1 && old\_state + 1 > k2) return false;

if (j == 1 && old\_state + 1 < k1 && k==n) return false;

return true;

}

// old\_state = so ngay lam viec lien tiep khi chua cap nhat loi giai thu k

void TRY(int k, int old\_state) {

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

if (isUCV(j, k, old\_state)) {

res[k] = j;

int curr\_state = j ? ++old\_state : 0;

if (k==n) {

print\_sol();

}

else TRY(k+1, curr\_state);

}

}

}

int main() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

res[0] = -1;

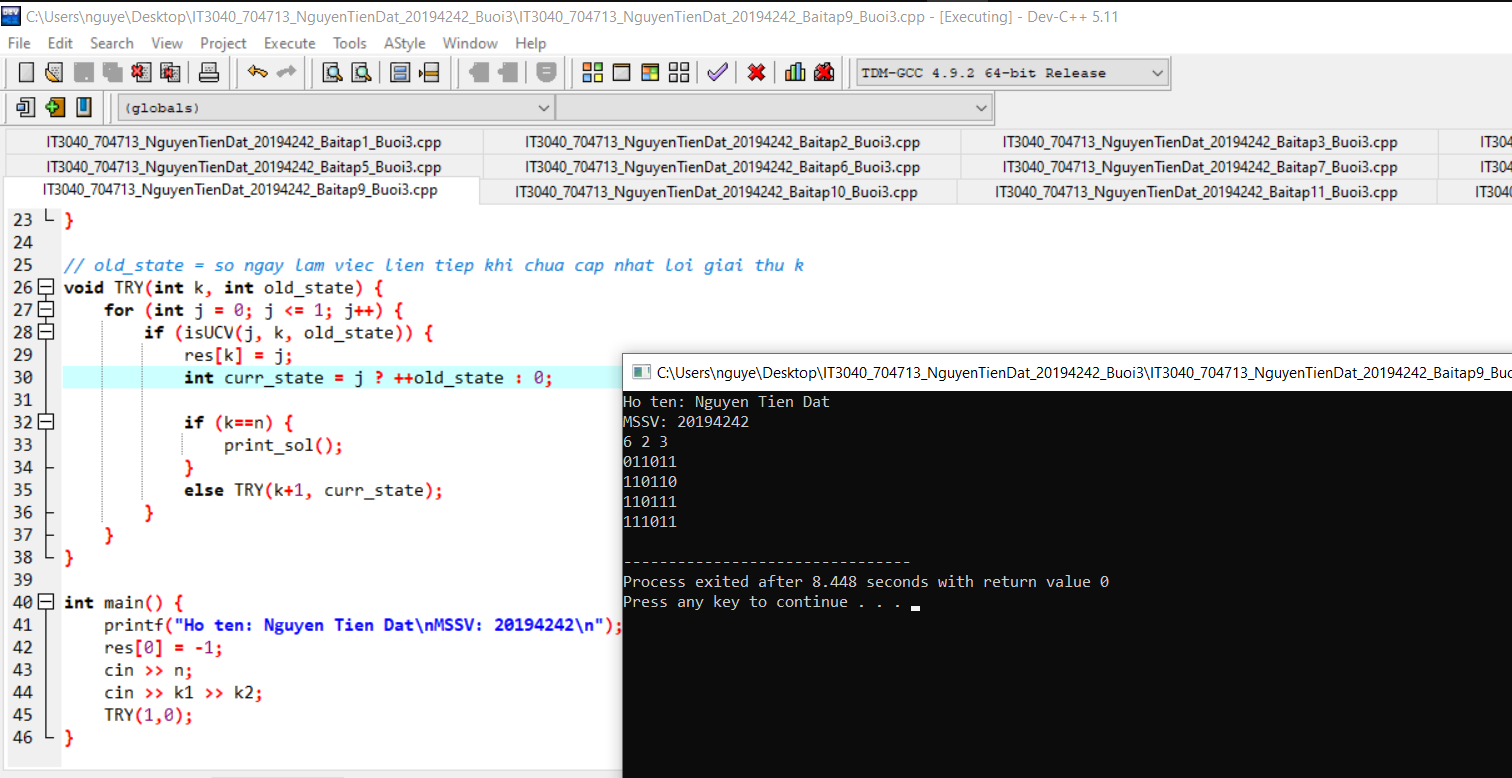
cin >> n;

cin >> k1 >> k2;

TRY(1,0);

}

Kết quả:



### 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 SS là xâu gồm nn ký tự 0. Hãy liệt kê tất cả các xâu nhị phân độ dài nn, có khoảng cách Hamming với SS bằng HH. Các xâu phải được liệt kê theo thứ tự từ điển

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap10\_Buoi3.cpp

#include <iostream>

using namespace std;

int n, H;

int res[1000];

void print\_sol() {

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

cout << res[i];

}

cout << endl;

}

bool isUCV(int j, int k, int old\_state) {

if (j == 1 && old\_state + 1 > H) return false;

return true;

}

void TRY(int k, int old\_state) {

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

if (isUCV(j, k, old\_state)) {

res[k] = j;

if (j == 1) ++old\_state;

if (k==n) {

if (old\_state == H) print\_sol();

}

else TRY(k+1, old\_state);

}

}

}

int main() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

int t;

cin >> t;

while (t>0) {

cin >> n >> H;

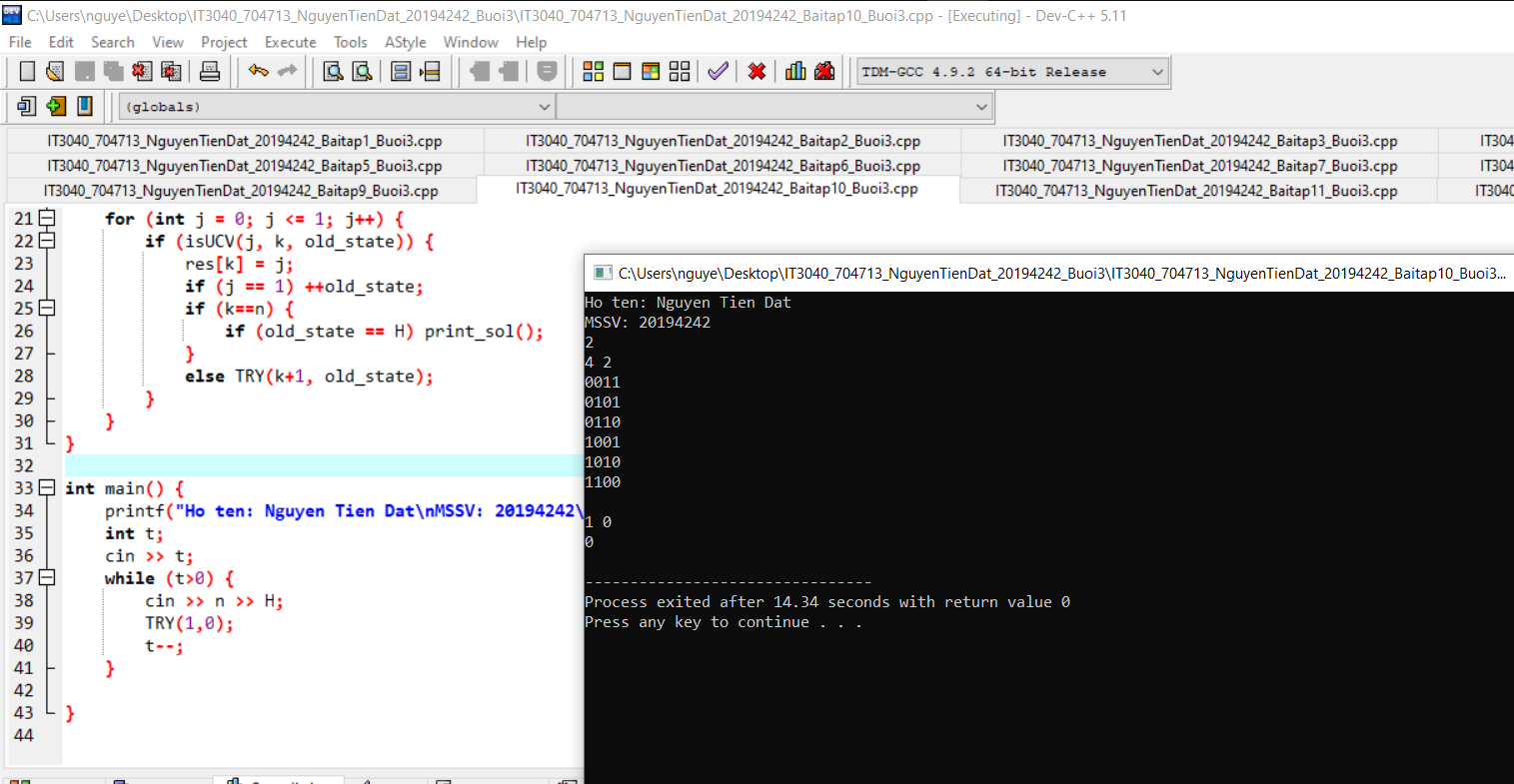
TRY(1,0);

t--;

}

}

Kết quả:



### Bài tập 11: Lịch trình chụp ảnh

### Superior là một hòn đảo tuyệt đẹp với nn địa điểm chụp ảnh và các đường một chiều nối các điểm chụp ảnh với nhau. Đoàn khách tham quan có rr 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.

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap11\_Buoi3.cpp

#include <iostream>

#include <vector>

#include <sstream>

using namespace std;

#define MAX 100

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

int n, r, c[MAX][MAX]; //# so thành pho, so khach và ma tran chi phí

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 = 0; //# 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

vector<int> place;

int start, des, num\_place; // luu thanh pho dau, cuoi va so thanh pho muon di

//# Ð?c d? li?u vào

void input(){

cin >> n >> r;

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

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

cin >> c[i][j];

}

}

bool isUCV(int k, int j) {

if (mark[j] == 1) return false;

if (c[x[k-1]][j] == 0) return false;

return true;

}

//# Thu?t toán quay lui

void TRY(int k){

// i la chi so cua vector place

for(int i = 2; i < num\_place; i++){

// code here

if (isUCV(k,place[i])) {

x[k] = place[i];

mark[place[i]] = 1;

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

// loi giai bo phan cap cuoi,

// tinh chi phi den tp cuoi cung de so sanh lay best

if (k == num\_place-1) {

if (c[x[k]][des] != 0) {

curr += c[x[k]][des];

best = (best > curr ? curr : best);

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

}

}

else TRY(k+1);

// thuc hien xog 1 loi giai khoi phuc trang thai ban dau

mark[place[i]] = 0;

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

}

}

}

int main() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

input();

string str; getline(cin,str);

while (r>0) {

best = INT\_MAX;

curr = 0;

getline(cin, str);

place.push\_back(-1);

while (!str.empty()){

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

int tmp = 0;

convert >> tmp;

place.push\_back(tmp);

if (str.find(" ") > str.size()){

break;

} else {

str.erase(0, str.find(" ") + 1); // Update string

}

}

start = place[1];

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

num\_place = place.size()-1;

x[1] = start;

x[num\_place] = des;

for(int i=0; i<MAX; i++) mark[i] = 0;

// cout << start << " " << des << " " << num\_place << endl;

// for (int i = 1; i <= num\_place; i++) cout << place[i] << " ";

TRY(2);

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

else cout << best << endl;

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

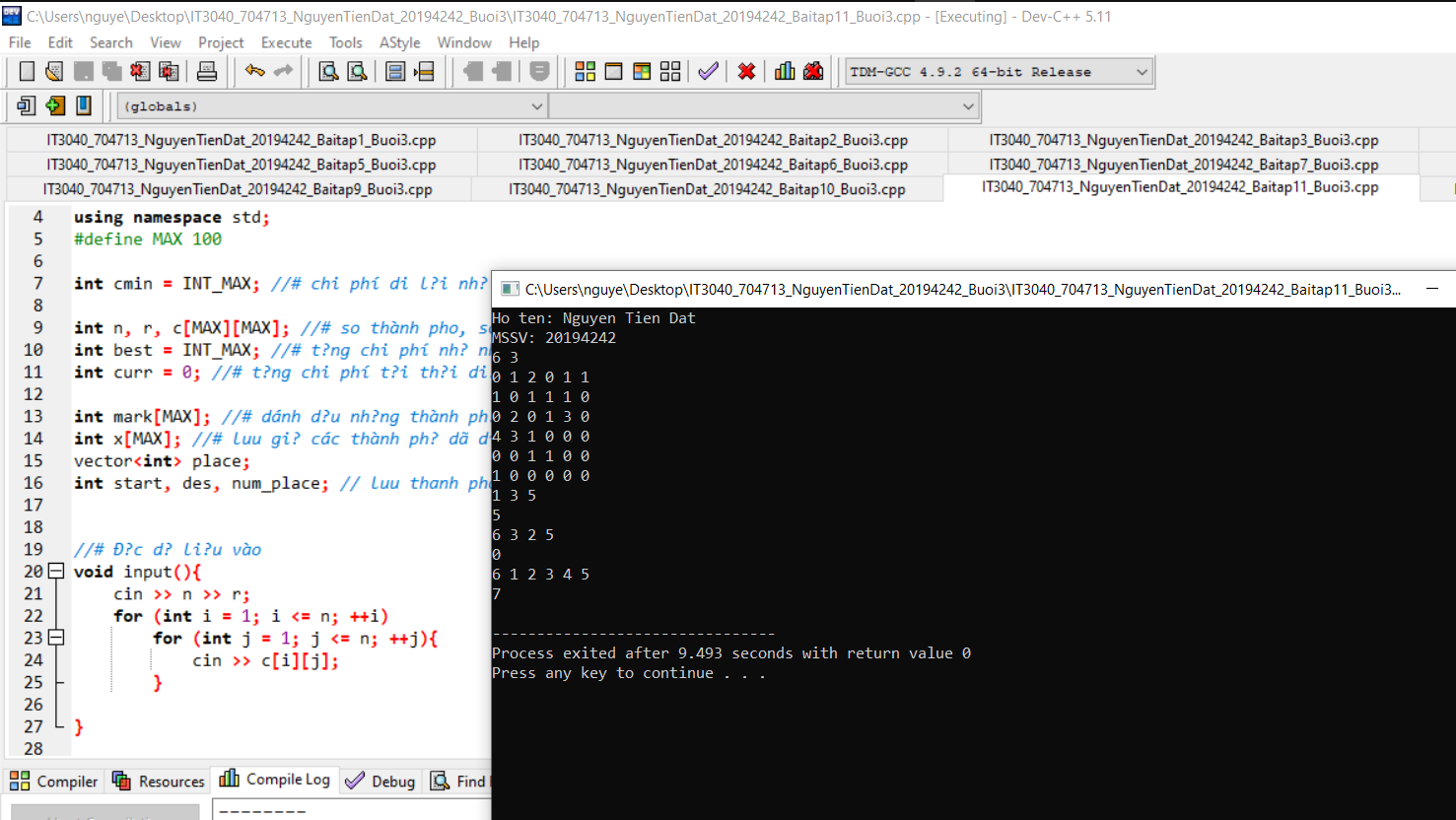
r--;

}

return 0;

}

Kết quả:



### 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.

Tên file: IT3040\_704713\_NguyenTienDat\_20194242\_Baitap12\_Buoi3.cpp

#include <iostream>

#include <vector>

using namespace std;

int n, m;

vector<int> a[1000];

int k;

int mark[1000];

int x[1000]; // luu dinh da di qua

int result = 0;

void input() {

cin >> n >> k;

cin >> m;

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

mark[i] = 0;

a[i].push\_back(-1);

}

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

int tmp1, tmp2;

cin >> tmp1 >> tmp2;

a[tmp1].push\_back(tmp2);

a[tmp2].push\_back(tmp1);

}

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

// for (int j = 1; j < a[i].size(); j++) cout << a[i][j] << " ";

// cout << endl;

// }

}

bool isUCV(int i, int j) {

if (i==1) return true;

if (mark[j] == 1) return false;

// chech co duong di tu x[i-1] den x[i] ko?

int check\_path = 0;

for (int l = 1; l < a[x[i-1]].size(); l++) {

if (j == a[x[i-1]][l]) check\_path = 1;

}

if (check\_path == 0) return false;

return true;

}

void print\_sol() {

// for (int i = 1; i <= k; i++) cout << x[i] << " ";

// cout << endl;

result++;

}

void TRY(int i) {

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

if (isUCV(i,j)) {

x[i] = j;

mark[j] = 1;

if (i - 1 == k) { // i la so dinh , k la so canh

print\_sol();

}

else TRY(i+1);

mark[j] = 0;

}

}

}

int main() {

printf("Ho ten: Nguyen Tien Dat\nMSSV: 20194242\n");

input();

TRY(1);

cout << result/2;

}

Kết quả:

