Contents

[Danh mục hình ảnh 2](#_Toc57325646)

[Bài thực hành số 3: Đệ quy và khử đệ quy để giải quyết một số bài toán 3](#_Toc57325647)

[Phần 1. Thực hành về đệ quy 3](#_Toc57325648)

[1.1 Đệ quy - quay lui 3](#_Toc57325649)

[Bài tập 1: Tính dãy Lucas 3](#_Toc57325650)

[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 . 3](#_Toc57325651)

[Bài tập 2: Quân mã đi tuần 4](#_Toc57325652)

[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) 4](#_Toc57325653)

[1.2 Kỹ thuật nhánh cận 6](#_Toc57325654)

[Bài tập 3: Bài toán người du lịch 6](#_Toc57325655)

[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 6](#_Toc57325656)

[1.3 Đệ quy có nhớ 9](#_Toc57325657)

[Bài tập 4: LIS 9](#_Toc57325658)

[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](#_Toc57325659)

[Phần 2. Khử đệ quy 11](#_Toc57325660)

[Bài tập 5: Tính tổ hợp 11](#_Toc57325661)

[Tính Ckn 11](#_Toc57325662)

[Bài tập 6: Tìm ước chung lớn nhất 13](#_Toc57325663)

[Tính ước chung lớn nhất của hai số cho trước 13](#_Toc57325664)

[Bài tập 7: Liệt kê xâu nhị phân 14](#_Toc57325665)

[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 14](#_Toc57325666)

[Phần 3. Bài tập về nhà 16](#_Toc57325667)

[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] . 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 16](#_Toc57325668)

[Bài tập 10: Khoảng cách Hamming 19](#_Toc57325669)

[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 19](#_Toc57325670)

[Bài tập 11: Lịch trình chụp ảnh 21](#_Toc57325671)

[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. 21](#_Toc57325672)

[Bài tập 12: Đếm đường đi 25](#_Toc57325673)

[Cho đồ thị vô hướng G , hãy đếm số đường đi đi qua k cạnh và không đi qua đỉnh nào quá một lần. 25](#_Toc57325674)

# Danh mục hình ảnh

[Hình 1 Bài 3.1 Tính dãy Lucas. 4](#_Toc57325675)

[Hình 2 Bài 3.2 . Quân mã đi tuần 6](#_Toc57325676)

[Hình 3 Bài 3.3 Đệ quy và khử đệ quy 8](#_Toc57325677)

[Hình 4 Bài 3.4 LIS 11](#_Toc57325678)

[Hình 5 Bài 3.5 Tính tổ hợp 13](#_Toc57325679)

[Hình 6 Bài 3.6 Tìm ước chung lớn nhất. 14](#_Toc57325680)

[Hình 7 Bài 3.7 Liệt kê xâu nhị phân 16](#_Toc57325681)

[Hình 8 Bài 3.9 Lập lịch cho y tá 19](#_Toc57325682)

[Hình 9 Bài 3.10 Khoảng cách Hamming 21](#_Toc57325683)

[Hình 10 Bài 3.11 Lịch trình chụp ảnh 25](#_Toc57325684)

[Hình 11 Bài 3.12 Đếm đường đi 27](#_Toc57325685)

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

## Phần 1. Thực hành về đệ quy

### Đệ quy - quay lui

#### 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 .

**Tên file: 20183702-NguyenDucDat\_Bai 3\_1.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 va ten: Nguyen Duc Dat\n");

printf("MSSV: 20183702\n\n");

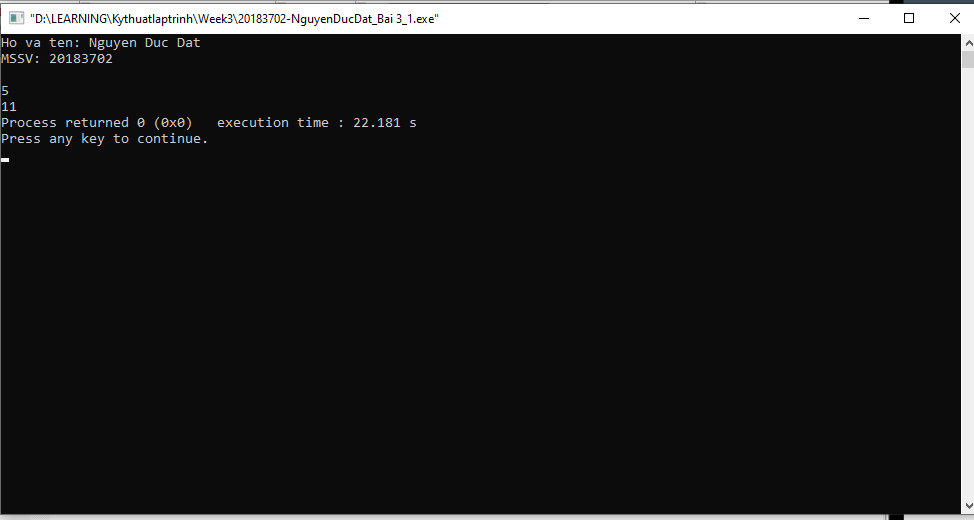
int n;

cin >> n;

cout << lucas(n);

}

Kết quả:



Hình 1 Bài 3.1 Tính dãy Lucas.

#### Bài tập 2: Quân mã đi tuần

#### 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)

**Tên file: 20183702-NguyenDucDat\_Bai 3\_2.cpp**

#include <iostream>

using namespace std;

int n;

int X[100], Y[100]; //# Lưu tọa độ các bước di chuyển của quân mã

int mark[100][100]; //# Đánh dấu vị trí các ô mà quân mã đã di chuyển qua

//# Mảng hx, hy mô tả 8 vị trí quân mã có thể di chuyển kể từ vị trí hiện tại

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 chuyển tìm được

void print\_sol(){

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

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

exit(0);

}

//# Thuật toán quay lui

void TRY(int k){

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

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

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

if(xx>0 && xx<=n && yy >0 && yy <=n && mark[xx][yy]==0){

X[k] = xx;

Y[k] = yy;

mark[xx][yy] = 1;

if(k==n\*n){

print\_sol();

}

else TRY(k+1);

mark[xx][yy] = 0;

}

}

}

int main(){

printf("Ho va ten: Nguyen Duc Dat\n");

printf("MSSV: 20183702\n\n");

cin >> n;

mark[1][1] = 1;

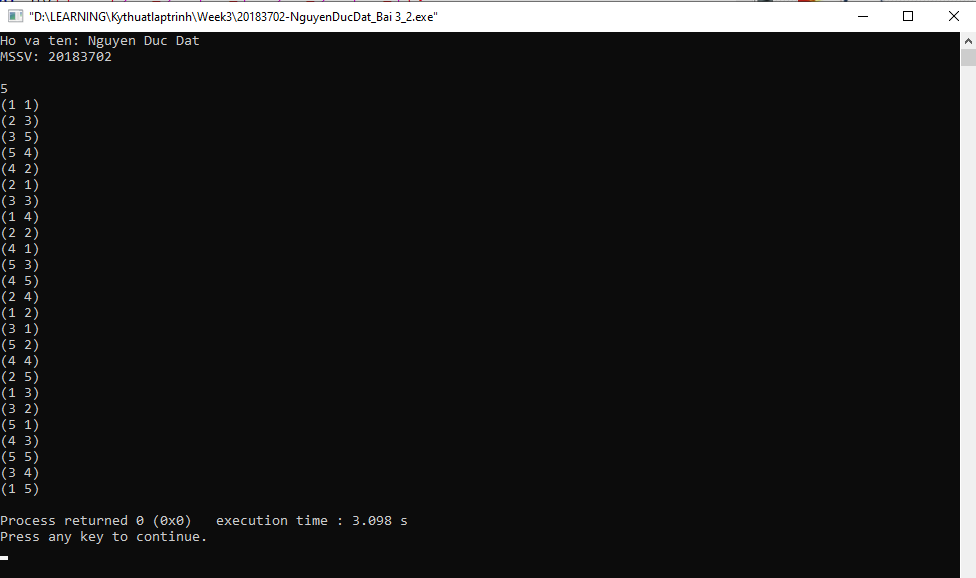
X[1] = Y[1] = 1;

TRY(2);

return 0;

}

Kết quả:



Hình 2 Bài 3.2 . Quân mã đi tuần

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

**Tên file: 20183702-NguyenDucDat\_Bai 3\_3.cpp**

#include <bits/stdc++.h>

using namespace std;

#define MAX 100

int n, c[MAX][MAX]; //# số thành phố và ma trận chi phí

int cmin = INT\_MAX; //# chi phí đi lại nhỏ nhất giữa hai thành phố khác nhau

int best = INT\_MAX; //# tổng chi phí nhỏ nhất cần tìm, ban đầu đặt bằng giá trị vô cùng lớn INT\_MAX = 2^31-1

int cur; //# tổng chi phí tới thời điểm hiện tại

int mark[MAX]; //# đánh dấu những thành phố đã đi

int x[MAX]; //# lưu giữ các thành phố đã đi

//# Đọc dữ liệu 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]);

}

}

//# Thuật toán quay lui

void TRY(int k)

{

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

{

if(!mark[i])

{

x[k] = i;

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

mark[i] = 1;

if(k==n)

{

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

{

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

}

}

else

{

int g = cur + cmin\*(n-k+1);

if(g<best)

TRY(k+1);

}

mark[i] = 0;

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

}

}

}

int main() {

printf("Ho va ten: Nguyen Duc Dat\n");

printf("MSSV: 20183702\n\n");

input();

x[1] = 1;

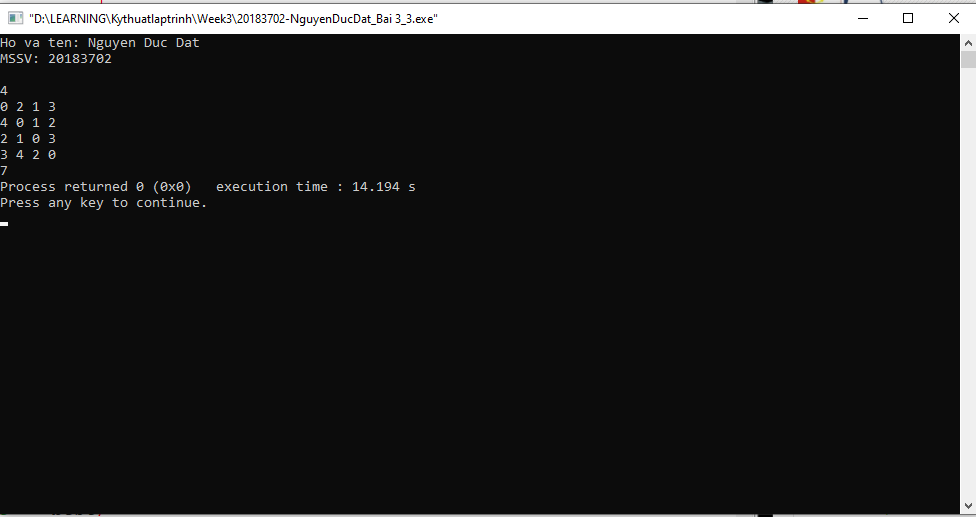
TRY(2);

cout << best;

return 0;

}

Kết quả:



Hình 3 Bài 3.3 Đệ quy và khử đệ quy

### 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: 20183702-NguyenDucDat\_Bai 3\_4.cpp**

#include <bits/stdc++.h>

using namespace std;

int a[1000], n;

int mem[1000]; //# mảng ghi nhớ lời giải các bài toán con đã được giải

void init(){

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

}

//# Quy hoạch động,

//# Hàm lis(i) trả về độ dài dãy con tăng dài nhất kết thúc bởi a[i]

int lis(int i) {

if(i==0) mem[i] = 1;

if(mem[i]==-1)

{

int res = 0;

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

{

int cur = lis(j);

if(a[i]>a[j] && cur> res)

{

res = cur;

}

}

mem[i] = res +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(){

printf("Ho va ten: Nguyen Duc Dat\n");

printf("MSSV: 20183702\n\n");

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;

}

}

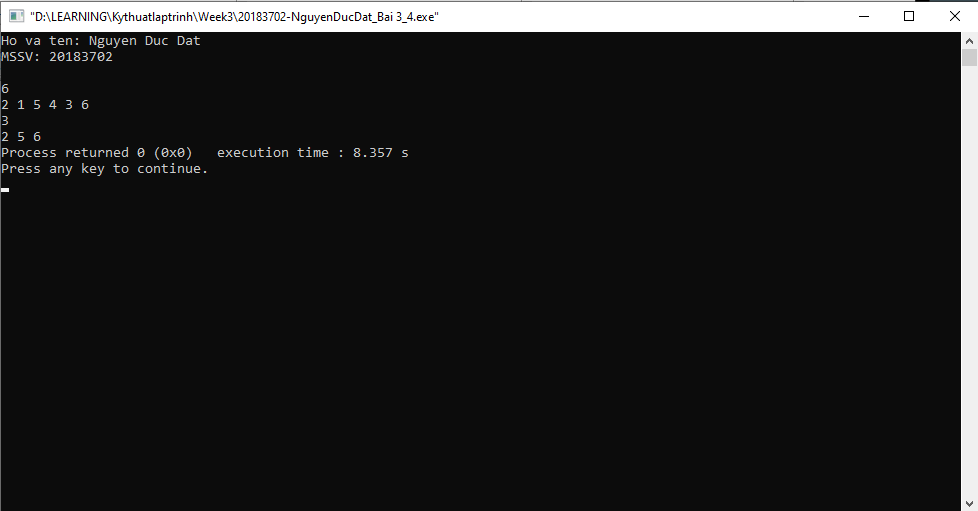
cout << res << endl;

trace(pos);

return 0;

}

Kết quả:



Hình 4 Bài 3.4 LIS

## Phần 2. Khử đệ quy

#### Bài tập 5: Tính tổ hợp

#### Tính Ckn

**Tên file: 20183702-NguyenDucDat\_Bai 3\_5.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){

//# Khử đệ quy

int res = 1;

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

res= res \*n/i;

}

return res;

}

int main() {

printf("Ho va ten: Nguyen Duc Dat\n");

printf("MSSV: 20183702\n\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));

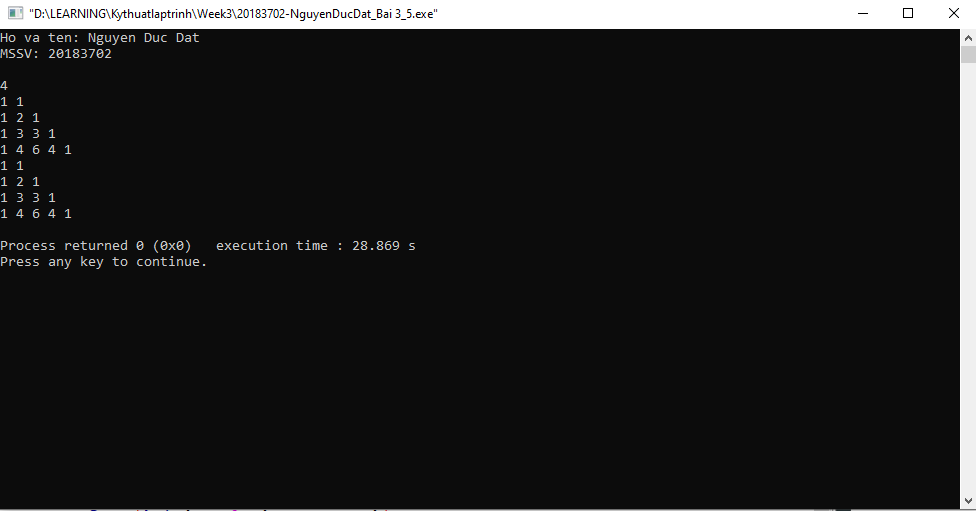
printf("\n");

}

return 0;

}

Kết quả:



Hình 5 Bài 3.5 Tính tổ hợp

#### Bài tập 6: Tìm ước chung lớn nhất

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

**Tên file: 20183702-NguyenDucDat\_Bai 2\_6.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){

//# Khử đệ quy

while(b!=0){

int sd = a%b;

a = b;

b = sd;

}

return a;

}

int main() {

printf("Ho va ten: Nguyen Duc Dat\n");

printf("MSSV: 20183702\n\n");

int a, b;

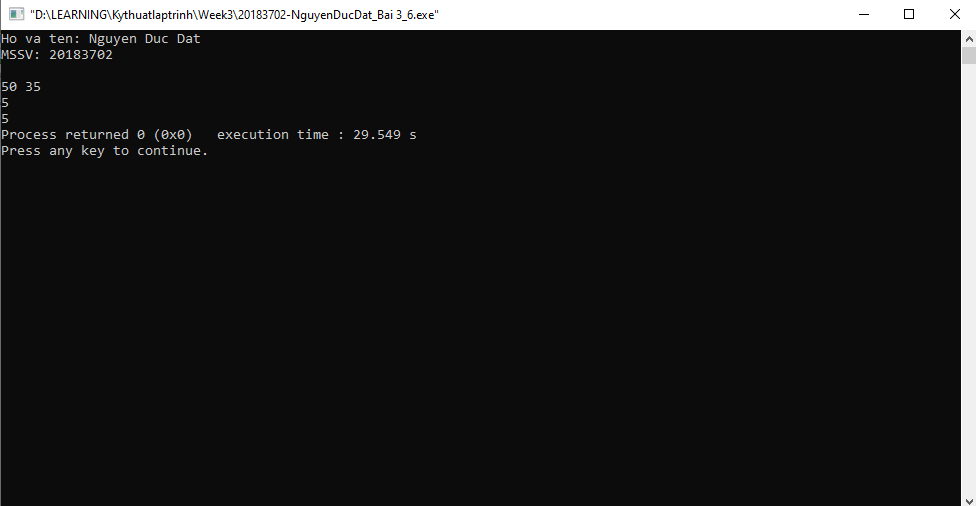
cin >> a >> b;

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

return 0;

}

Kết quả:



Hình 6 Bài 3.6 Tìm ước chung lớn nhất.

#### 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

**Tên file: 20183702-NguyenDucDat\_Bai 3\_7.cpp**

#include <bits/stdc++.h>

using namespace std;

int L = 0;

int n, k;

int x[25];

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 va ten: Nguyen Duc Dat\n");

printf("MSSV: 20183702\n\n");

cin >> n >> k;

stack<state> s;

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));

}

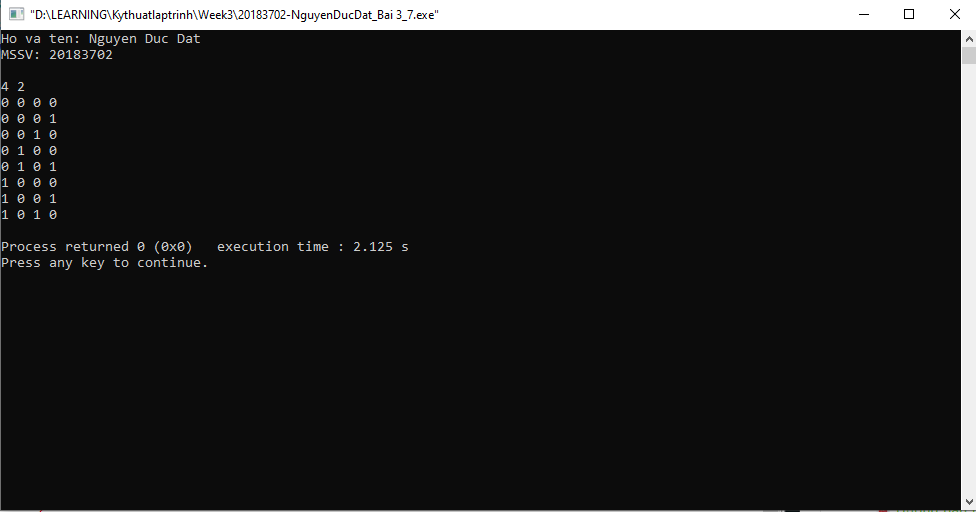
++top.j;

}

return 0;

}

Kết quả:



Hình 7 Bài 3.7 Liệt kê xâu nhị phân

## 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] . 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

**Tên file: 20183702-NguyenDucDat\_Bai 3\_9.cpp**

#include<iostream>

using namespace std;

int N, K1, K2;

int mark[200], x[200];

void solution(){

cout<<endl;

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

cout<<x[i];

}

}

int check(int v, int k)

{

if(k==1) return 1;

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

if(N-k+1<K1) return 0;

return v;

}

else{

int cnt = 0, i = k-1;

while(x[i]!=0 && i>=1){

cnt++;

i--;

}

if(cnt<K1) return v;

else if(cnt<K2) return 1;

else return !v;

}

}

void TRY(int k){

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

if(check(i,k)){

x[k] = i;

mark[k] = 1;

// cout<<"x["<<k<<"]="<<i<<endl;

if(k==N){

solution();

}

else{

TRY(k+1);

}

mark[k] = 0;

}

}

}

int main(){

printf("Ho va ten: Nguyen Duc Dat\n");

printf("MSSV: 20183702\n\n");

cin>>N>>K1>>K2;

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

mark[i] = 0;

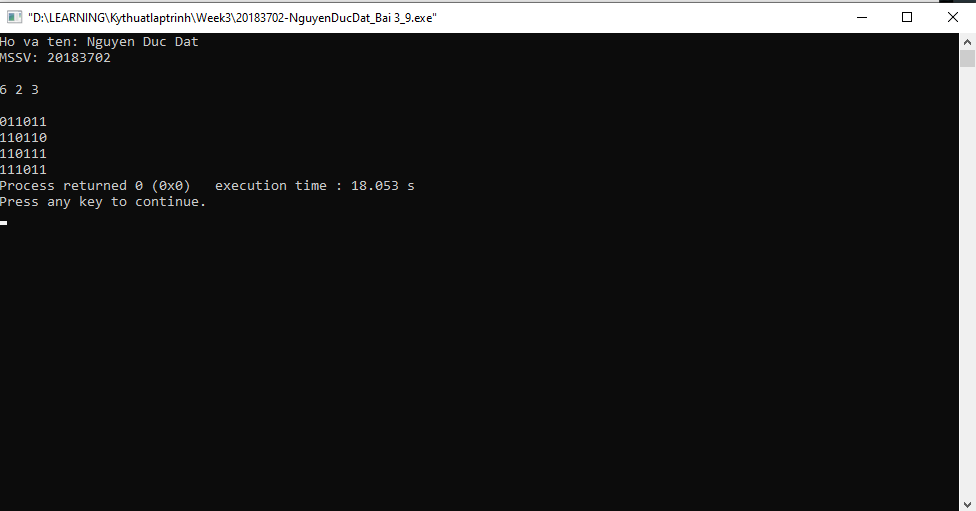
}

TRY(1);

return 0;

}

Kết quả:



Hình 8 Bài 3.9 Lập lịch cho y tá

### 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

**Tên file: 20183702-NguyenDucDat\_Bai 3\_10.cpp**

#include<iostream>

using namespace std;

int x[200], Y[200];

int N[20], H[20], T;

void solution(int n){

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

cout<<Y[i];

}

cout<<endl;

}

void ToHop (int i, int n, int h) {

if(h<=0 || n<h){

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

Y[i] = 0;

cout<<Y[i];

}

}

else

{

for (int j = x[i-1] + 1; j <= n-h+i; j++) {

x[i] = j;

Y[x[i]] = 0;

if (i == h)

solution(n);

else

ToHop(i+1,n,h);

Y[x[i]] = 1;

}

}

}

int main(){

printf("Ho va ten: Nguyen Duc Dat\n");

printf("MSSV: 20183702\n\n");

cin>>T;

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

cin>>N[i]>>H[i];

}

for(int i = 0; i<T; i++)

{

for(int j = 1; j<=N[i]; j++)

{

Y[j] = 1;

}

x[0] = 0;

ToHop(1,N[i], H[i]);

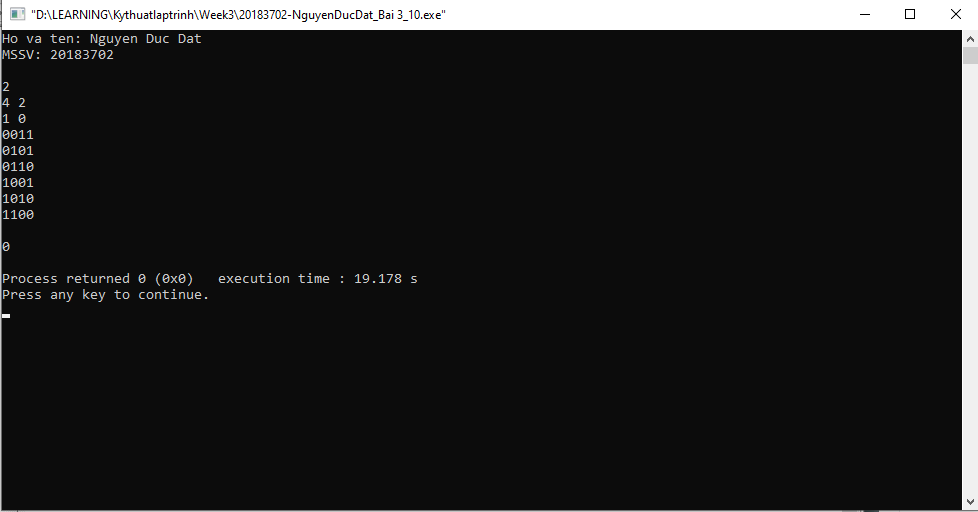
cout<<endl;

}

return 0;

}

Kết quả:



Hình 9 Bài 3.10 Khoảng cách Hamming

### 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.

**Tên file: 20183702-NguyenDucDat\_Bai 3\_11.cpp**

#include <bits/stdc++.h>

using namespace std;

#define MAX 100

int n,r, c[MAX][MAX]; //# số thành phố và ma trận chi phí

int cmin = INT\_MAX; //# chi phí đi lại nhỏ nhất giữa hai thành phố khác nhau

int best = INT\_MAX; //# tổng chi phí nhỏ nhất cần tìm, ban đầu đặt bằng giá trị vô cùng lớn INT\_MAX = 2^31-1

int cur = 0; //# tổng chi phí tới thời điểm hiện tại

int mark[MAX]; //# đánh dấu những thành phố đã đi

int x[MAX][MAX];

int t[MAX][MAX];

int s[MAX];

int inTour[MAX][MAX];

void input()

{

cin>>n>>r;

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 = 1; i<=r; i++)

{

string str;

if(i==1) cin.ignore();

getline(cin, str);

s[i] = 0;

for(int j = 0; j<str.length(); j++)

{

if(str[j]!=' ')

{

s[i]++;

t[i][s[i]] = str[j] - '0';

inTour[i][t[i][s[i]]] = 1;

}

}

}

}

void Clear(){

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

mark[i] = 0;

}

cur = 0;

best = INT\_MAX;

}

int Done(int k)

{

for(int i = 2; i<=s[k];i++){

if(!mark[t[k][i]]) return 0;

}

return 1;

}

void TRY(int k, int j)

{

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

{

if(!mark[i] && c[x[j][k-1]][i] && inTour[j][i])

{

x[j][k] = i;

mark[i] = 1;

cur = cur + c[x[j][k-1]][x[j][k]];

if(Done(j) && x[j][k]==t[j][s[j]])

{

if(best>cur)

{

best = cur;

}

}

else TRY(k+1,j);

mark[i] = 0;

cur = cur - c[x[j][k-1]][x[j][k]];

}

}

}

int main()

{

printf("Ho va ten: Nguyen Duc Dat\n");

printf("MSSV: 20183702\n\n");

input();

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

x[i][1] = t[i][1];

mark[t[i][1]] = 1;

TRY(2,i);

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

else cout << best << endl;

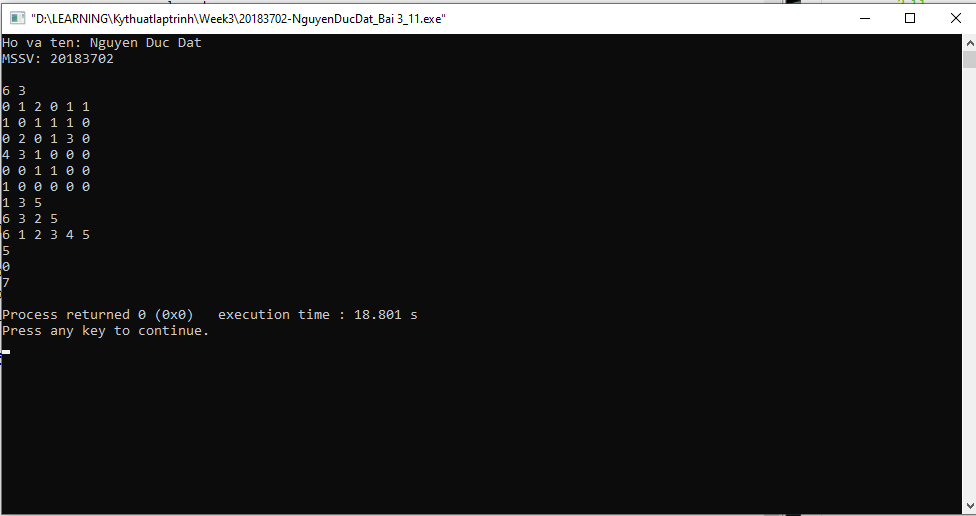
Clear();

}

return 0;

}

Kết quả:



Hình 10 Bài 3.11 Lịch trình chụp ảnh

### 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: 20183702-NguyenDucDat\_Bai 3\_12.cpp**

#include <bits/stdc++.h>

#define maxn 32

using namespace std;

int n, k, m;

map<int, vector<int> > e;

bool visited[maxn];

void init(){

cin >> n >> k >> m;

int u, v;

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

cin >> u >> v;

e[u].push\_back(v);

e[v].push\_back(u);

}

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

visited[i] = false;

}

int count\_path(int u, int k){

int res = 0;

if (k == 0) return 1;

visited[u] = true;

for (auto v : e[u]){

if (!visited[v])

res += count\_path(v, k - 1);

}

visited[u] = false;

return res;

}

int main(){

printf("Ho va ten: Nguyen Duc Dat\n");

printf("MSSV: 20183702\n\n");

init();

int res = 0;

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

if (!visited[i])

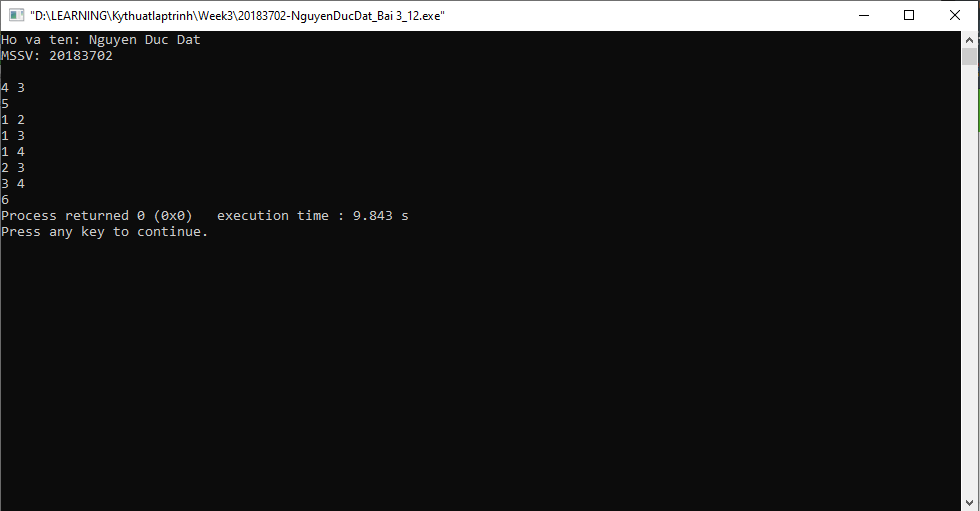
res += count\_path(i, k);

}

cout << res/2;

}}

Kết quả:



Hình 11 Bài 3.12 Đếm đường đi