**CÁCH ĐÁNH GIÁ ĐIỂM THỰC HÀNH**

**HỌC PHẦN: IT3040 - KỸ THUẬT LẬP TRÌNH – 2022.2**

1. **Quy định, yêu cầu:** 
   * Tài liệu và nội dung thực hành chấm điểm trên hệ thống:

https://lab.soict.hust.edu.vn/

* + Bài tập trên lớp chấm điểm tự động (các bài không chấm trên hệ thống làm vào máy tính 🡺 làm báo cáo thực hành – Theo mẫu).
  + Hạn nộp báo cáo trên Teams (Bài tập trên lớp + Bài tập về nhà): 1 tuần.

1. **Đánh giá điểm thực hành**

1. Chuyên cần (đúng giờ, nghiêm túc trong giờ học) - Điểm danh trên Teams: 10%

2. Báo cáo thực hành (bài tập trên lớp + Về nhà) theo mẫu nộp trên Teams: 40%

3. Trắc nghiệm – Form trên Teams: 10%

4. Kiểm tra thực hành: 40%. (Tiết 2,3 buổi thực hành thứ 5).

**Điểm thưởng: 5% 🡪 10% (Cho Mục 1,2 điểm TB từ 9-10).**

Tham gia thực hành đúng giờ đầy đủ theo thời khóa biểu (nếu có lý do không đi thực hành đúng kíp được thì gửi mail xin phép thực hành bù trước 1 ngày qua mail [hoalt@soict.hust.edu.vn](mailto:hoalt@soict.hust.edu.vn), Tiêu đề: đăng ký học bù – IT3040 – MaLopTH.

Các kíp có thể bù:

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Thời gian, địa điểm, Tuần học** | **Mã nhóm** | **Mã lớp** |
| **1** |  |  |  |
| **2** |  |  |  |
| **3** |  |  |  |
| **4** |  |  |  |
| **5** |  |  |  |
| **6** |  |  |  |
| **7** |  |  |  |

**Nếu nghỉ không có lý do 3 buổi, không thực hành bù thì điểm chuyên cần, báo cáo và BTVN coi như 0 điểm thực hành.**

Mục lục

[Bài thực hành số 3 3](#_Toc151586806)

[Bài tập 1: Tính dãy Lucas 3](#_Toc151586807)

[Bài tập 2: Quân mã đi tuần 3](#_Toc151586808)

[Bài tập 3: Bài toán người du lịch 3](#_Toc151586809)

[Bài tập 4: LIS 4](#_Toc151586810)

[Bài tập 5: Tính tổ hợp 4](#_Toc151586811)

[Bài tập 6: Tìm ước chung lớn nhất 4](#_Toc151586812)

[Bài tập 7: Liệt kê xâu nhị phân 4](#_Toc151586813)

[Bài tập 8: Cân đĩa 5](#_Toc151586814)

[Bài tập 9: Lập lịch cho y tá 5](#_Toc151586815)

[Bài tập 10: Khoảng cách Hamming 6](#_Toc151586816)

[Bài tập 11: Lịch trình chụp ảnh 7](#_Toc151586817)

[Bài tập 12: Đếm đường đi 7](#_Toc151586818)

Danh mục hình ảnh minh họa

[Hình 1: Bài tập 1 6](#_Toc152087820)

[Hình 2: Kết quả bài tập 1 6](#_Toc152087821)

[Hình 3: Bài tập 2 phần 1 7](#_Toc152087822)

[Hình 4: Bài tập 2 phần 2 8](#_Toc152087823)

[Hình 5: Kết quả bài tập 2 phần 1 9](#_Toc152087824)

[Hình 6: Kết quả bài tập 2 phần 2 10](#_Toc152087825)

[Hình 7: Kết quả bài tập 2 phần 3 10](#_Toc152087826)

[Hình 8: Kết quả bài tập 2 phần 4 11](#_Toc152087827)

[Hình 9: Kết quả bài tập 2 phần 5 12](#_Toc152087828)

[Hình 10: Bài tập 3 phần 1 14](#_Toc152087829)

[Hình 11: Bài tập 3 phần 2 15](#_Toc152087830)

[Hình 12: Bài tập 3 phần 3 15](#_Toc152087831)

[Hình 13: Kết quả bài tập 3 16](#_Toc152087832)

[Hình 14: Bài tập 4 phần 1 19](#_Toc152087833)

[Hình 15: Bài tập 4 phần 2 19](#_Toc152087834)

[Hình 16: Bài tập 4 phần 3 20](#_Toc152087835)

[Hình 17: Kết quả bài tập 4 20](#_Toc152087836)

[Hình 18: Bài tập 5 phần 1 23](#_Toc152087837)

[Hình 19: Bài tập 5 phần 2 23](#_Toc152087838)

[Hình 20: Kết quả bài tập 5 phần 1 24](#_Toc152087839)

[Hình 21: Kết quả bài tập 5 phần 2 24](#_Toc152087840)

[Hình 22: Kết quả bài tập 5 phần 3 25](#_Toc152087841)

[Hình 23: Kết quả bài tập 5 phần 4 25](#_Toc152087842)

[Hình 24: Bài tập 6 phần 1 27](#_Toc152087843)

[Hình 25: Bài tập 6 phần 2 27](#_Toc152087844)

[Hình 26: Kết quả bài tập 6 28](#_Toc152087845)

[Hình 27: Bài tập 7 phần 1 30](#_Toc152087846)

[Hình 28: Bài tập 7 phần 2 30](#_Toc152087847)

[Hình 29: Bài tập 7 phần 3 31](#_Toc152087848)

[Hình 30: Kết quả bài tập 7 phần 1 31](#_Toc152087849)

[Hình 31: Kết quả bài tập 7 phần 2 32](#_Toc152087850)

[Hình 32: Kết quả bài tập 7 phần 3 32](#_Toc152087851)

[Hình 33: Kết quả bài tập 7 phần 4 33](#_Toc152087852)

[Hình 34: Kết quả bài tập 7 phần 5 33](#_Toc152087853)

[Hình 35: Kết quả bài tập 7 phần 6 34](#_Toc152087854)

[Hình 36: Bài tập 8 phần 1 37](#_Toc152087855)

[Hình 37: Bài tập 8 phần 2 37](#_Toc152087856)

[Hình 38: Bài tập 8 phần 3 38](#_Toc152087857)

[Hình 39: Kết quả bài tập 8 38](#_Toc152087858)

[Hình 40: Bài tập 9 phần 1 42](#_Toc152087859)

[Hình 41: Bài tập 9 phần 2 43](#_Toc152087860)

[Hình 42: Bài tập 9 phần 3 44](#_Toc152087861)

[Hình 43: Bài tập 9 phần 4 45](#_Toc152087862)

[Hình 44: Test case 1 bài tập 9 45](#_Toc152087863)

[Hình 45: Test case 2 bài tập 9 46](#_Toc152087864)

[Hình 46: Test case 3 bài tập 9 phần 1 47](#_Toc152087865)

[Hình 47: Test case 3 bài tập 9 phần 2 48](#_Toc152087866)

[Hình 48: Bài tập 10 phần 1 53](#_Toc152087867)

[Hình 49: Bài tập 10 phần 2 54](#_Toc152087868)

[Hình 50: Bài tập 10 phần 3 54](#_Toc152087869)

[Hình 51: Test case 1 bài tập 10 phần 1 55](#_Toc152087870)

[Hình 52: Test case 1 bài tập 10 phần 2 56](#_Toc152087871)

[Hình 53: Test case 1 bài tập 10 phần 3 57](#_Toc152087872)

[Hình 54: Bài tập 11 phần 1 61](#_Toc152087873)

[Hình 55: Bài tập 11 phần 2 62](#_Toc152087874)

[Hình 56: Bài tập 11 phần 3 63](#_Toc152087875)

[Hình 57: Bài tập 11 phần 4 64](#_Toc152087876)

[Hình 58: Bài tập 11 phần 5 64](#_Toc152087877)

[Hình 59: Test case 1 bài tập 11 65](#_Toc152087878)

[Hình 60: Test case 2 bài tập 11 phần 1 66](#_Toc152087879)

[Hình 61: Test case 2 bài tập 11 phần 2 66](#_Toc152087880)

[Hình 62: Bài tập 12 phần 1 72](#_Toc152087881)

[Hình 63: Bài tập 12 phần 2 73](#_Toc152087882)

[Hình 64: Bài tập 12 phần 3 74](#_Toc152087883)

[Hình 65: Bài tập 12 phần 4 74](#_Toc152087884)

[Hình 66: Test case 1 bài tập 12 phần 1 75](#_Toc152087885)

[Hình 67: Test case 1 bài tập 12 phần 2 76](#_Toc152087886)

[Hình 68: Test case 2 bài tập 12 phần 1 77](#_Toc152087887)

[Hình 69: Test case 2 bài tập 12 phần 2 78](#_Toc152087888)

[Hình 70: Test case 2 bài tập 12 phần 3 79](#_Toc152087889)

[Hình 71: Test case 2 bài tập 12 phần 4 80](#_Toc152087890)

[Hình 72: Test case 3 bài tập 12 phần 1 81](#_Toc152087891)

[Hình 73: Test case 3 bài tập 12 phần 2 82](#_Toc152087892)

[Hình 74: Test case 3 bài tập 12 phần 3 83](#_Toc152087893)

[Hình 75: Test case 3 bài tập 12 phần 4 84](#_Toc152087894)

[Hình 76: Test case 4 bài tập 12 phần 1 85](#_Toc152087895)

[Hình 77: Test case 4 bài tập 12 phần 2 86](#_Toc152087896)

[Hình 78: Test case 4 bài tập 12 phần 3 87](#_Toc152087897)

[Hình 79: Test case 4 bài tập 12 phần 4 88](#_Toc152087898)

[Hình 80: Test case 5 bài tập 12 phần 1 89](#_Toc152087899)

[Hình 81: Test case 5 bài tập 12 phần 2 90](#_Toc152087900)

[Hình 82: Test case 5 bài tập 12 phần 3 91](#_Toc152087901)

[Hình 83: Test case 5 bài tập 12 phần 4 92](#_Toc152087902)

[Hình 84: Test case 6 bài tập 12 phần 1 93](#_Toc152087903)

[Hình 85: Test case 6 bài tập 12 phần 2 94](#_Toc152087904)

[Hình 86: Test case 6 bài tập 12 phần 3 95](#_Toc152087905)

[Hình 87: Test case 6 bài tập 12 phần 4 96](#_Toc152087906)

[Hình 88: Test case 6 bài tập 12 phần 5 97](#_Toc152087907)

[Hình 89: Test case 7 bài tập 12 phần 1 97](#_Toc152087908)

[Hình 90: Test case 7 bài tập 12 phần 2 98](#_Toc152087909)

[Hình 91: Test case 7 bài tập 12 phần 3 99](#_Toc152087910)

[Hình 92: Test case 7 bài tập 12 phần 4 100](#_Toc152087911)

[Hình 93: Test case 7 bài tập 12 phần 5 100](#_Toc152087912)

[Hình 94: Test case 8 bài tập 12 phần 1 101](#_Toc152087913)

[Hình 95: Test case 8 bài tập 12 phần 2 102](#_Toc152087914)

[Hình 96: Test case 8 bài tập 12 phần 3 103](#_Toc152087915)

[Hình 97: Test case 8 bài tập 12 phần 4 104](#_Toc152087916)

[Hình 98: Test case 8 bài tập 12 phần 5 104](#_Toc152087917)

[Hình 99: Test case 9 bài tập 12 phần 1 105](#_Toc152087918)

[Hình 100: Test case 9 bài tập 12 phần 2 106](#_Toc152087919)

[Hình 101: Test case 9 bài tập 12 phần 3 107](#_Toc152087920)

[Hình 102: Test case 9 bài tập 12 phần 4 108](#_Toc152087921)

[Hình 103: Test case 9 bài tập 12 phần 5 108](#_Toc152087922)

# Bài thực hành số 3

## Bài tập 1: Tính dãy Lucas

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

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Hình : Bài tập 1

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Hình : Kết quả bài tập 1

/\* To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include <stdio.h>

int lucas(int n) {

//Truong hop co ban de ket thuc de quy

if(n == 0) return 2;

else if(n == 1) return 1;

//Truong hop de quy

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

}

int main() {

for(int n = 0; n <= 10; ++n) {

printf("L[%d] = %d\n", n, lucas(n));

}

return 0;

}

//To Thai Linh 20215414

## Bài tập 2: Quân mã đi tuần

Trên bàn cờ vua kích thước *n* x *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)

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Hình : Bài tập 2 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Hình : Bài tập 2 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Song song

Mô tả được tạo tự động

Hình : Kết quả bài tập 2 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, mẫu

Mô tả được tạo tự động

Hình : Kết quả bài tập 2 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Kết quả bài tập 2 phần 3

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Kết quả bài tập 2 phần 4

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Kết quả bài tập 2 phần 5

/\* To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include <iostream>

using namespace std;

int n;//Khai bao bien n kieu int

int X[100], Y[100]; //luu toa do cac buoc di chuyen cua quan ma

int mark[100][100]; //# Danh dau vi tri cac o ma quan ma da di chuyen qua

//# Mang hx, hy mo ta 8 vi tri quan ma co the di chuyen ke tu vi tri hien tai

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

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

//# In ra day cac di chuyen tim duoc

void print\_sol(){

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

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

exit(0);

}

//# Thuat toan quay lui

void TRY(int k){

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

//vong lap de tinh cac nuoc di co the xuat phat tu X[k - 1] va Y[k - 1]

int xx = X[k-1] + hx[i];//Tinh gia tri toa do X[k]

int yy = Y[k-1] + hy[i];//Tinh gia tri toa do Y[k]

if( (!mark[xx][yy]) && (1 <= xx) && (xx <= n) && (1 <= yy) && (yy <= n) ) {

mark[xx][yy] = 1;//Gan gia tri cho mark bang 1 de danh dau da kiem tra

X[k] = xx;//Gan gia tri cua xx cho toa do X[k]

Y[k] = yy;//Gan gia tri cua yy cho toa do Y[k]

if(k == n \* n) print\_sol();//Ket thuc de quy in ra ket qua

else TRY(k + 1);//De quy cho lan lap tiep theo

mark[xx][yy] = 0;//Dat lai cho mark gia tri 0 de quay lui

}

}

}

int main(){

cin >> n;//Nhap gia tri cho n

mark[1][1] = 1;//Danh dau o (1,1) da di qua

X[1] = Y[1] = 1;//Gan toa do cho X[1] va Y[1]

TRY(2);//Kiem tra den buoc di thu 2

return 0;

}//To Thai Linh

## 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.

**Dữ liệu vào:**

Dòng 1: Chứa số nguyên n (1≤n≤16)

n dòng tiếp theo: Chứa ma trận c (0≤ci,j≤1000000)

**Kết quả:**

Ghi tổng chi phí nhỏ nhất có thể

**Ví dụ:**

**Dữ liệu mẫu:** 4  
0 2 1 3  
4 0 1 2  
2 1 0 3  
3 4 2 0

**Kết quả mẫu:**  
7

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Hình : Bài tập 3 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Hình : Bài tập 3 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Hình : Bài tập 3 phần 3

Ảnh có chứa văn bản, ảnh chụp màn hình, số, phần mềm

Mô tả được tạo tự động

Hình : Kết quả bài tập 3

/\* To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include <bits/stdc++.h>

using namespace std;

#define MAX 100

int n, c[MAX][MAX]; //# so thanh pho va ma tran chi phi

int cmin = INT\_MAX; //# chi phi di lai nho nhat giua cac thanh pho khac nhau

int best = INT\_MAX;

//# tong chi phi nho nhat can tim, ban dau dat gia tri vo cung lon INT\_MAX = 2^31-1

int curr; //# tong chi phi toi thoi diem hien tai

int mark[MAX]; //# danh dau nhung thanh pho da di

int x[MAX]; //# luu tru cac thanh pho da di

//# Doc du lieu dau vao

void input(){

cin >> n;//Nhap gia tri cho n

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

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

cin >> c[i][j];//nhap chi phi cho tung thanh pho

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

//tim chi phi nho nhat giua cac thanh pho

}

}

//# Thuat toan quy lui

void TRY(int k){

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

if(!mark[i]) {

x[k] = i;//Den lan k tham thanh pho thu i

mark[i] = 1;//Danh dau thanh pho i la da tham

curr += c[x[k - 1]][i];//Tinh toan chi phi hien tai

if(k == n) best = min(best, curr + c[i][1]);

//Neu tham het n thanh pho thi tinh chi phi tot nhat

else if(curr + cmin \* (n - k + 1) < best) TRY(k + 1);

//Tinh khoang uoc luong khi het n - k thanh pho con lai

//Neu nho hon best thi tiep tuc di den thanh pho tiep theo

mark[i] = 0;

//Dat lai bien danh dau mark la 0

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

//Dat lai gia tri chi phi

}

}

}

int main() {

input();//Goi den ham nhap gia tri dau vao

x[1] = 1;//Thanh pho 1 da di

TRY(2);//De quy voi thanh pho thu 2

cout << best;//In ra chi phi nho nhat

return 0;

}

//To Thai Linh 20215414

## 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*

**Dữ liệu vào:**

Dòng 1: Chứa số nguyên *n*(1 ≤ *n* ≤ 1000)

Dòng 2: Chứa *n* số nguyên *a1 a2 …… an (|ai| ≤ 109)*

**Kết quả:**

Dòng đầu tiên chứa độ dài dãy con tăng dài nhất

Dòng thứ 2 chứa chỉ số các phần tử được chọn vào dãy con đó

Nếu có nhiều dãy con tăng dài nhất, in ra dãy bất kỳ trong số đó

**Ví dụ:**

**Dữ liệu mẫu:**  
6  
2 1 5 4 3 6

**Kết quả mẫu:**  
3  
2 5 6

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Hình : Bài tập 4 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Hình : Bài tập 4 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, số

Mô tả được tạo tự động

Hình : Bài tập 4 phần 3

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Kết quả bài tập 4

/\* To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include <bits/stdc++.h>

using namespace std;

int a[1000], n;//Khai bao mang a kieu int luu gia tri dau vao, bien n kieu int

int mem[1000]; //# mang ghi nho loi giai cua cac bai toan con da duoc giai

void init(){

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

//Khoi tao cac phan tu cua mem bang -1

}

//# Quy hoach dong,

//# Ham list(i) tra ve do dai cua day con tang dai nhat ket thuc boi a[i]

int lis(int i) {

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

//neu da co loi giai thi tra ve

return mem[i];

}

int res = 1;

//khoi tao res = 1 luu ket qua cua list(i)

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

//Duyet qua tung phan tu neu co phan tu nao be hon a[i]

//Cong vao ket qua res

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

res = max(res, 1 + lis(j));

}

}

//Luu vao trong mang ket qua mem[i]

mem[i] = res;

return res;

}

//# 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]){

//De quy truy vet loi giai

trace(j);

break;

}

}

//In ra a[i]

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

}

int main(){

init();//Khoi tao mang mem

cin >> n;//Nhap gia tri cua n

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

// Nhap gia tri tung phan tu cua mang a

int res = 1, pos = 0;

//res la do dai cua day con tang dai nhat

//pos vi tri cua phan tu cuoi cung trong day con tang dai nhat

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

if (res < lis(i)){

res = lis(i);

pos = i;

}

}

cout << res << endl;// In gia tri day con dai nhat

trace(pos);//In ra day con tang dai nhat

return 0;

}

//To Thai Linh 20215414

## Bài tập 5: Tính tổ hợp

Tính Ckn

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Hình : Bài tập 5 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

Hình : Bài tập 5 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

Hình : Kết quả bài tập 5 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, số

Mô tả được tạo tự động

Hình : Kết quả bài tập 5 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Kết quả bài tập 5 phần 3

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu

Mô tả được tạo tự động

Hình : Kết quả bài tập 5 phần 4

/\* To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include <iostream>

using namespace std;

//Ham de quy tinh c(k, n) = c(k -1, n -1) + C(k, n - 1)

int binom(int n, int k) {

if (k > n) return 0;//Truong hop co so neu k > n thi tra ve 0

if (k == 0) return 1;//Truong hop co so neu k == 0 thi tra ve 1

//De quy

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

}

int c[1000][1000];//Khai bao mang c de luu gia tri cac c[n][k]

int binom2(int n, int k){

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

//khoi tao c[0][i] = 1

c[0][i] = 1;

//Tinh gia tri cac C(0, i) den C(i, i)

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

//Cong thuc newton

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

}

}

//Tra ve gia tri C(k, n)

return c[k][n];

}

int main() {

int m;//khai bao bien m kieu int

cin >> m;//Nhap gia tri cho m

//In ra cac gia tri tu C(0, 1), C(1, 1),C(0, 2),..., C(n, n) bang binom(n, k)

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

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

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

printf("\n");

}

//In ra cac gia tri tu C(0, 1), C(1, 1),C(0, 2), ..., C(n, n) bang binom2(n, k)

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

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

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

printf("\n");

}

return 0;

}

//To Thai Linh 20215414

## Bài tập 6: Tìm ước chung lớn nhất

Tìm ước chung lớn nhất của hai số cho trước

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Hình : Bài tập 6 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Hình : Bài tập 6 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Phông chữ

Mô tả được tạo tự động

Hình : Kết quả bài tập 6

/\*To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include <iostream>

using namespace std;

int gcd(int a, int b){

//Truong hop co so dung de quy

if (b == 0) return a;

//De quy

return gcd(b, a % b);

}

int gcd2(int a, int b){

//Khu de quy khi nao b == 0 thi dung lai

while(b) {

a %= b;//a = du cua a / b

swap(a, b);//Doi cho a va b

}

return a;

}

int main() {

int a, b;//Khai bao 2 bien a va b kieu int

cin >> a >> b;//Nhap gia tri cho 2 bien a va b

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

//Tinh uoc chung cua a va b theo 2 cach

return 0;

}

//To Thai Linh 20215414

## 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

**Dữ liệu vào:**

Một dòng duy nhất chứa hai số nguyên *n k* (1 ≤ *k* ≤ *n* ≤20)

**Kết quả:**

Với mỗi xâu tìm được, in ra *n* ký tự trên một dòng, các ký tự cách nhau bởi dấu cách. Các xâu cần được liệt kê theo thứ tự từ điển

**Ví dụ:**

**Dữ liệu mẫu:**  
4 2

**Kết quả mẫu:**  
0 0 0 0  
0 0 0 1  
0 0 1 0  
0 1 0 0  
0 1 0 1  
1 0 0 0  
1 0 0 1  
1 0 1 0

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Hình : Bài tập 7 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Hình : Bài tập 7 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

Hình : Bài tập 7 phần 3

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

Hình : Kết quả bài tập 7 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Kết quả bài tập 7 phần 2

Ảnh có chứa ảnh chụp màn hình, văn bản, Hình chữ nhật

Mô tả được tạo tự động

Hình : Kết quả bài tập 7 phần 3

Ảnh có chứa ảnh chụp màn hình, Hình chữ nhật, văn bản, thiết kế

Mô tả được tạo tự động

Hình : Kết quả bài tập 7 phần 4

Ảnh có chứa ảnh chụp màn hình, văn bản

Mô tả được tạo tự động

Hình : Kết quả bài tập 7 phần 5

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Hình : Kết quả bài tập 7 phần 6

/\* To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include <bits/stdc++.h>

using namespace std;

struct state{

int i, j, old\_L;

//i: bieu dien loi giai bo phan cap i, truoc do la x1, x2,..., x[i - 1]

//j: gia tri ung cu vien dang xet cho vi tri x[i]

//old\_L so ky tu 1 lien tiep o cuoi day x1, x2,..., x[i - 1]

//# constructor

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

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

};

int main() {

int n, k;//Khai bao 2 bien n, k kieu int

cin >> n >> k;//Nhap gia tri cho 2 bien n va k

int x[n+1];//Khai bao mang chua n + 1 phan tu de luu ket qua

stack<state> s;//Khai bao stack s chua cau truc state

//# number of consecutive suffix 1

int L = 0;//Khoi tao bien L kieu int bang 0

s.push(state(1, 0));//Khoi tao phan tu dau tien cua stack la 1, 0, 0

while (!s.empty()){

state &top = s.top();

//# if a new binary sequence is found

if (top.i > n){

//In ket qua

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

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

//Loai bo khoi ngan xep

s.pop();

continue;

}

//# Khử đệ quy

//Back track

//Kiem tra ung cu vien neu lon hon 0 thi gan old\_L cho L

if(top.j > 0) {

L = top.old\_L;

}

//Neu moi day con da tham thi loai bo

if(top.j > 1) {

s.pop();

continue;

}

//Neu chua co k so 1 lien tiep hoac ung cu vien bang 0

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

x[top.i] = top.j;//Gan vi tri x[i] = j

top.old\_L = L; // Cap nhat gia tri cua old\_L neu j > 0

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

//Cap nhat gia tri cua L neu top.j > 0 thi tang 1

//Neu top.j = 0 thi khong tang

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

//Them trang thai moi vao ngan xep stack

}

//Tang gia tri cua ung cu vien

++top.j;

}

return 0;

}

//To Thai Linh 20215414

## 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.

**Dữ liệu mẫu:**

6 10  
7 1 2 3 4 5

**Kết quả mẫu:**

-1+2+3+4+5=10

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Hình : Bài tập 8 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 8 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 8 phần 3

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, số

Mô tả được tạo tự động

Hình : Kết quả bài tập 8

/\*To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include <bits/stdc++.h>

using namespace std;

struct state{

int i, j;

state(int \_i = 0, int \_j = 0): i(\_i), j(\_j) {}

};

int main() {

//Khai bao 2 bien n va M kieu int

//de luu so luong qua can va khoi luong M

int n, M;

cin >> n >> M;//Nhap gia tri cho 2 bien n va M

//Khai bao mang m gom n + 1 phan tu

//luu khoi luong cua tung qua can

int m[n+1];

//Nhap gia tri cho mang m

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

//Khai bao mang x luu trang thai chon can

// -1 neu chon dau tru, 1 neu chon dau cong

int x[n+1];

//Khai bao stack s de luu cac trang thai khi thuc hien de quy

stack<state> s;

//# sum of selected weights

int sum = 0;

//Khoi tao trang thai ban dau

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

while (!s.empty()){

//Lay phan tu dau tien cua stack

state &top = s.top();

if (top.i > n){

//Neu vuot qua so luong qua can va tong bang M thi in ket qua

if (sum == 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);

//Thoat khoi chuong trinh

}

//Loai bo phan tu dau stack

s.pop();

continue;

}

if(top.j > -1) {

sum -= m[top.i] \* x[top.i];

}

if(top.j > 1) {

s.pop();

continue;

}

x[top.i] = top.j;

//Gan dau top.j cho vi tri top.i trong mang a

sum += m[top.i] \* x[top.i];

//Ting tong khoi luong

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

//Them trang thai tiep theo vao ngan xep

++top.j;

//Tang bien dau them 1 don vi

}

cout << -1;

//Neu khong tim thay in ra -1

return 0;

}

//To Thai Linh 20215414

## 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

**Dữ liệu vào:**

Ghi 3 số nguyên *N, K1, K2* (*N ≤ 200, K1 <K2 ≤70*)

**Kết quả:**

Ghi danh sách các lịch tìm được theo thứ tự từ điển

**Ví dụ:**

**Dữ liệu mẫu:**

6 2 3

**Kết quả mẫu:**

011011  
110110  
110111  
111011

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 9 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 9 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 9 phần 3

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 9 phần 4

Case 1:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

Hình : Test case 1 bài tập 9

Case 2:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

Hình : Test case 2 bài tập 9

Case 3:

Ảnh có chứa ảnh chụp màn hình, văn bản, mẫu, đen và trắng

Mô tả được tạo tự động

Hình : Test case 3 bài tập 9 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, mẫu, đen và trắng

Mô tả được tạo tự động

Hình : Test case 3 bài tập 9 phần 2

/\*To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include <bits/stdc++.h>

using namespace std;

const int MAX = 210;

int N, K1, K2, counter = 0;

int a[MAX], sum = 0, dau = -1;

void TRY(int k) {

//Ham thu tat ca cac kha nang cua ngay thu k

if(k > N) {

//Duyet qua tat ca N ngay thi in ket qua

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

if(sum < K1 && sum > 0) return;

cout<<a[i];

//In ra lich lam viec

}

counter++;

cout<<endl;

return;

}

if(dau == -1) {

//Kiem tra xu ly ngay dau tien

//Truong hop ngay dau tien la nghi lam

a[k] = 0;

dau = 0;

//Goi de quy ngay tiep theo

TRY(k + 1);

//Truong hop ngay dau tien di lam

a[k] = 1;

dau = 1;

sum = 1;

//Goi de quy ngay tiep theo

TRY(k + 1);

exit(0);

}

//Luu gia tri cua sum vao old\_S de quay lui

int old\_S = sum;

//Luu gia tri cua dau vao old\_D de quay lui

int old\_D = dau;

if(old\_D == 0) {

//Neu phan tu dung truoc bang 0

a[k] = 1;//Gan phan tu hien tai bang 1

sum = 1;

dau = 1;

//De quy voi phan tu tiep theo

TRY(k + 1);

//Quay lui

sum = old\_S;

dau = old\_D;

}

//Neu ngay truoc do la ngay di lam

//hay phan tu truoc do la 1

else {

//Neu tong so ngay lam viec nho hon K2

if(old\_S < K2) {

//Tong so ngay lam viec lon hon hoac bang K1

if(old\_S >= K1) {

//Gan ngay tiep theo la ngay nghi

a[k] = 0;

dau = 0;

sum = 0;

TRY(k + 1);

sum = old\_S;

dau = old\_D;

}

//Truong hop ngay tiep theo la ngay di lam

a[k] = 1;

dau = 1;

sum += 1;

TRY(k + 1);//De quy

sum = old\_S;//Quay lui

dau = old\_D;

}

else {

//Xu ly tong ngay lam viec khong lon hon K2

a[k] = 0;

dau = 0;

sum = 0;

//De quy

TRY(k + 1);

//Quay lui

sum = old\_S;

dau = old\_D;

}

}

}

int main(int argc, char\*\* argv) {

//Nhap vao gia tri N, K1, K2

cin>>N>>K1>>K2;

//Bat dau thu voi ngay dau tien

TRY(1);

cout<<counter<<endl;

return 0;

}

//To Thai Linh 20215414

## 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.

**Dữ liệu vào:**

Dòng đầu chứa *T* là số testcase

*T* dòng tiếp theo, mỗi dòng mô tả một testcase, ghi *N* và *H* (1 ≤ *H* ≤ *N* ≤16)

**Kết quả:**

Với mỗi testcase, in ra danh sách các xâu thỏa mãn. In ra một dòng trống giữa hai testcase

**Ví dụ:**

**Dữ liệu mẫu:**

2  
4 2

1 0

**Kết quả mẫu:**

0011  
0101  
0110  
1001  
1010  
1100

0

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 10 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 10 phần 2

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, số

Mô tả được tạo tự động

Hình : Bài tập 10 phần 3

Case 1:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế đồ họa

Mô tả được tạo tự động

Hình : Test case 1 bài tập 10 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế đồ họa

Mô tả được tạo tự động

Hình : Test case 1 bài tập 10 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế đồ họa

Mô tả được tạo tự động

Hình : Test case 1 bài tập 10 phần 3

/\*To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include <bits/stdc++.h>

using namespace std;

//Khai bao cac bien toan cuc

//T la so test case, t[20][2] la gia tri cac test case

//H la so vi tri khac nhau

//N la do dai xau

//mang a luu gia tri tai moi vi tri

//sum luu tong cua cac phan tu

int T, t[20][2], H, N;

int a[20], sum = 0;

void TRY(int i){

//Ham de quy thu tat ca cac gia tri cho bit tai vi tri i trong xau

if (i > N) {

//Neu da xet den tat ca cac vi tri trong xau

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

if (sum != H) return;

//Neu tong cac phan tu khong bang H thi khong in

cout << a[i];

//In ra tung phan tu

if (i == N) cout << endl;

}

return;

}

int old\_S = sum;

//Luu gia tri cua sum vao old\_S de luu trang thai truoc thay doi

a[i] = 0;

//Khoi tao phan tu dau tien bang 0

TRY(i+1); //De quy

sum = old\_S;//Quay lui

if (sum < H) {

//Truong hop chua du khoang cach hamming

a[i] = 1;//Tai vi tri i gan bang 1

++sum;//Tang tong

TRY(i+1);//De quy

sum = old\_S;//Quay lui

}

}

int main(int argc, const char\*\* argv) {

cin >> T;//Nhap vao so test case

//Nhap gia tri cua tung test case

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

cin >> t[i][0];

cin >> t[i][1];

}

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

N = t[i][0];//Lay gia tri do dai xau

H = t[i][1];//Lay gia tri hamming

cout << endl;

TRY(1);//De quy voi phan tu dau tien

}

return 0;

}

//To Thai Linh 20215414

## 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.

**Dữ liệu vào:**

Dòng đầu chứa *n* và *r*

Tiếp theo là ma trận *n×n* mô tả chi phí đi lại giữa các địa điểm. Chi phí bằng 0 có nghĩa là không thể đi lại giữa hai địa điểm đó.

*r* dòng tiếp theo chứa danh sách các địa điểm mà người *r* đưa ra. Lưu ý là hành mỗi hành trình cần phải bắt đầu và kết thúc bởi hai đỉnh đầu và cuối của danh sách, còn các địa điểm còn lại có thể thăm theo bất kỳ thứ tự nào

**Kết quả:**

Gồm *r* dòng ghi chi phí đi lại ít nhất của *r* người theo thứ tự đầu vào

**Ví dụ:**

**Dữ liệu mẫu:**

6 3  
0 1 2 0 1 1  
1 0 1 1 1 0  
0 2 0 1 3 0  
4 3 1 0 0 0  
0 0 1 1 0 0  
1 0 0 0 0 0  
1 3 5  
6 3 2 5  
6 1 2 3 4 5

**Kết quả mẫu:**

5  
0

7

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 11 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 11 phần 2

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 11 phần 3

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 11 phần 4

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Bài tập 11 phần 5

Case 1:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

Hình : Test case 1 bài tập 11

Case 2:

Ảnh có chứa văn bản, ảnh chụp màn hình, mẫu, đen và trắng

Mô tả được tạo tự động

Hình : Test case 2 bài tập 11 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màu đen

Mô tả được tạo tự động

Hình : Test case 2 bài tập 11 phần 2

/\*To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include<bits/stdc++.h>

using namespace std;

const int MAX = 10000;

int n, r;//Khai bao bien toan cuc n va r kieu int

int price[MAX][MAX];//Ma tran chi phi

int x[MAX];//Luu hanh trinh tot nhat

bool visited[MAX];//Bien danh dau da tham

vector<int> vt;//Luu hanh trinh

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

vector<int> result; //Vector luu ket qua

//Ham nhap du lieu dau vao

void inputData(){

cin >> n >> r;//Nhap gia tri n va r

//Nhap ma tran chi phi

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

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

cin >> price[i][j];

}

}

}

//Ham kiem tra nguoi a co the di den i hay khong

bool check(int a, int i){

//Neu da tham tra ve false

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

//Neu chi phi = 0 tra ve false

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

return true;

}

//Ham tinh gia tri tot nhat

void solution(){

//Neu khong co chi phi di den diem ket thuc thi tra ve

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

//Chi phi tot nhat la gia tri nho nhat cua cac chi phi

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

}

void TRY(int k){

//Ham kiem tra lich trinh di chuyen

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

if (check(k, i)){

//Danh dau diem da tham

visited[vt[i]] = true;

//Tinh tong chi phi

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

//Cap nhat duong di tot nhat

x[k] = vt[i];

//Neu di den diem gan cuoi thi tinh gia tri tot nhat

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

//De quy

else TRY(k+1);

//Quay lui

visited[vt[i]] = false;

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

}

}

}

int main(){

string str;

inputData(); //Goi ham nhap du lieu

getline(cin,str);//Doc mot dong tu dau vao gan cho str

while(r > 0){

//Vong lap tuong ung voi so nguoi trong doan

best = INT\_MAX;

//Khoi chi phi nho nhat la INT\_MAX

sum\_price = 0;//Khoi tao tong chi phi

getline(cin, str);

while (!str.empty()){

//Chay vong lap cho den khi het ky tu

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

int tmp = 0;

convert >> tmp;

vt.push\_back(tmp - 1);

//Them gia tri vao vector de luu danh sach diem yeu cau

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

//Kiem tra neu khong con khoang trang thi thoat

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

//Xoa phan tu chuyen doi tu str

}

//Khoi tao diem bat dau la phan tu dau tien trong danh sach

start = vt[0];

//Diem den la phan tu cuoi cung

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

//So diem can di la kich thuoc cua vector

numOfPoint = vt.size();

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

//Ham khoi tao bien tham cac dinh bang false

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

visited[i] = false;

}

//De quy bat dau tu phan tu dau tien

TRY(1);

//Neu gia tri tot nhat khong thay doi thi in ra 0

if (best == INT\_MAX) result.push\_back(0);

else result.push\_back(best);

//Them vao day ket qua

//Xoa vector

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

r--;//Duyet den nguoi tiep theo

}

//In day ket qua

for(int i : result) {

cout<<i<<endl;

}

}

//To Thai Linh 20215414

## 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.

**Dữ liệu vào:**

Dòng 1: Chứa hai số nguyên *n* và *k* (1 ≤ *n* ≤ 30, 1 ≤ *k* ≤10) với *n* là số đỉnh của *G*. Các đỉnh sẽ được đánh số từ 1 đến *n*

Dòng 2: Chứa số nguyên *m* (1≤ *m* ≤60) là số cạnh của *G*

*m* dòng tiếp theo: Mỗi dòng chưa hai số nguyên là một cạnh của *G*

**Kết quả:**

Số lượng đường đi đơn độ dài *k*

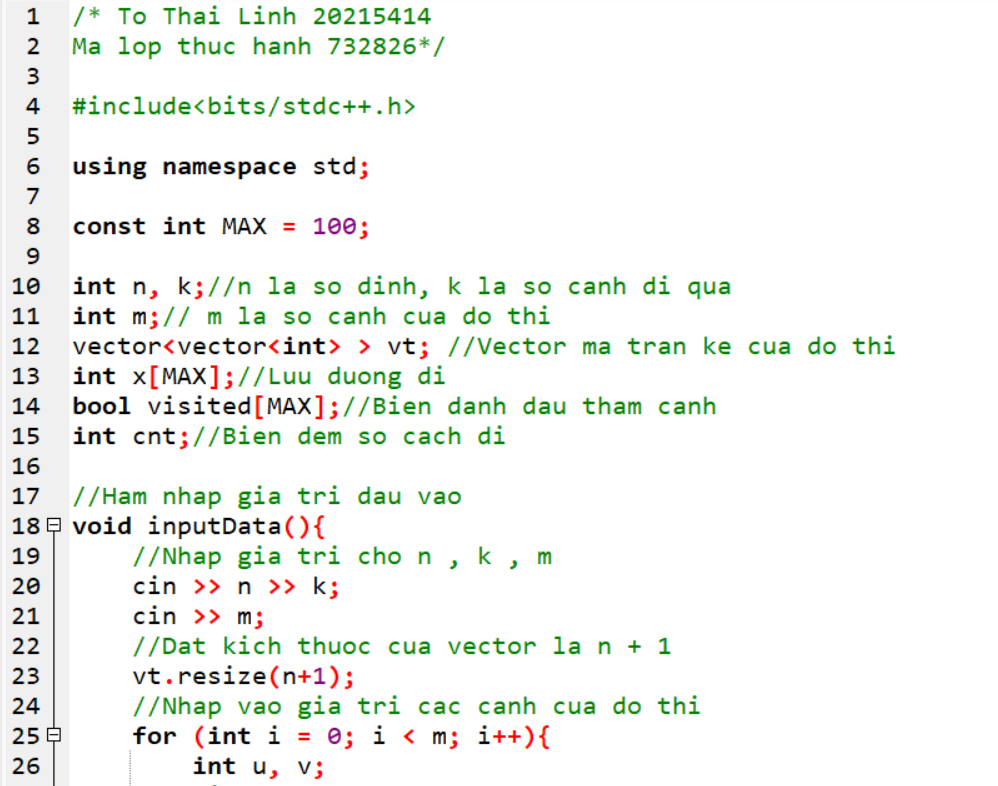
**Ví dụ:**

**Dữ liệu mẫu:**

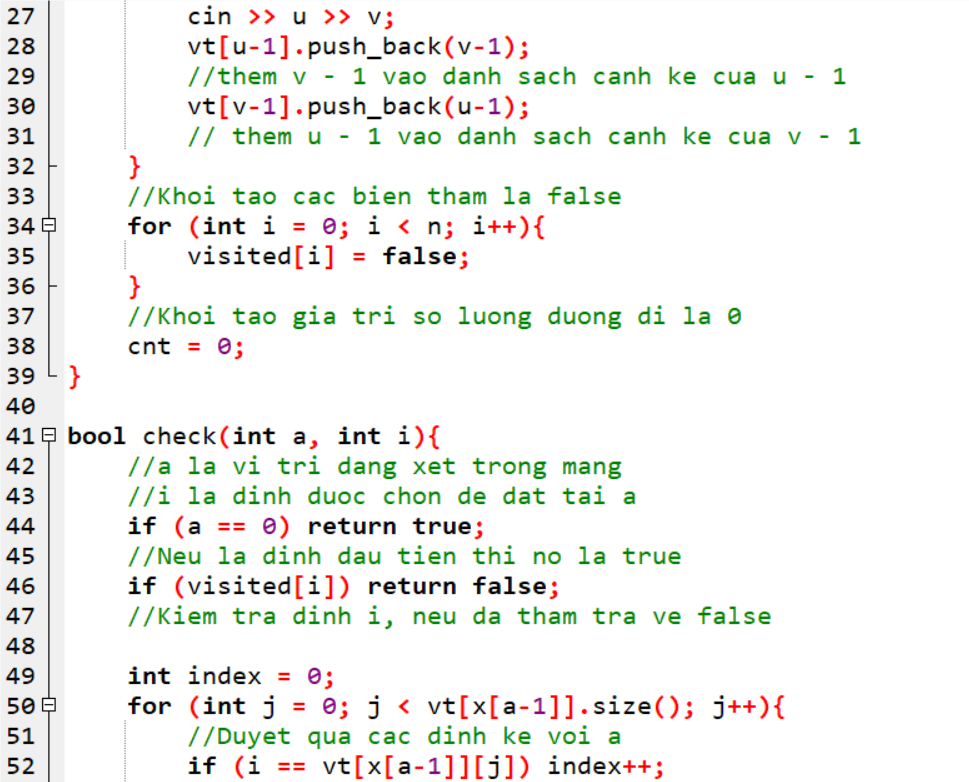
4 3  
5  
1 2  
1 3  
1 4  
2 3  
3 4

**Kết quả mẫu:**

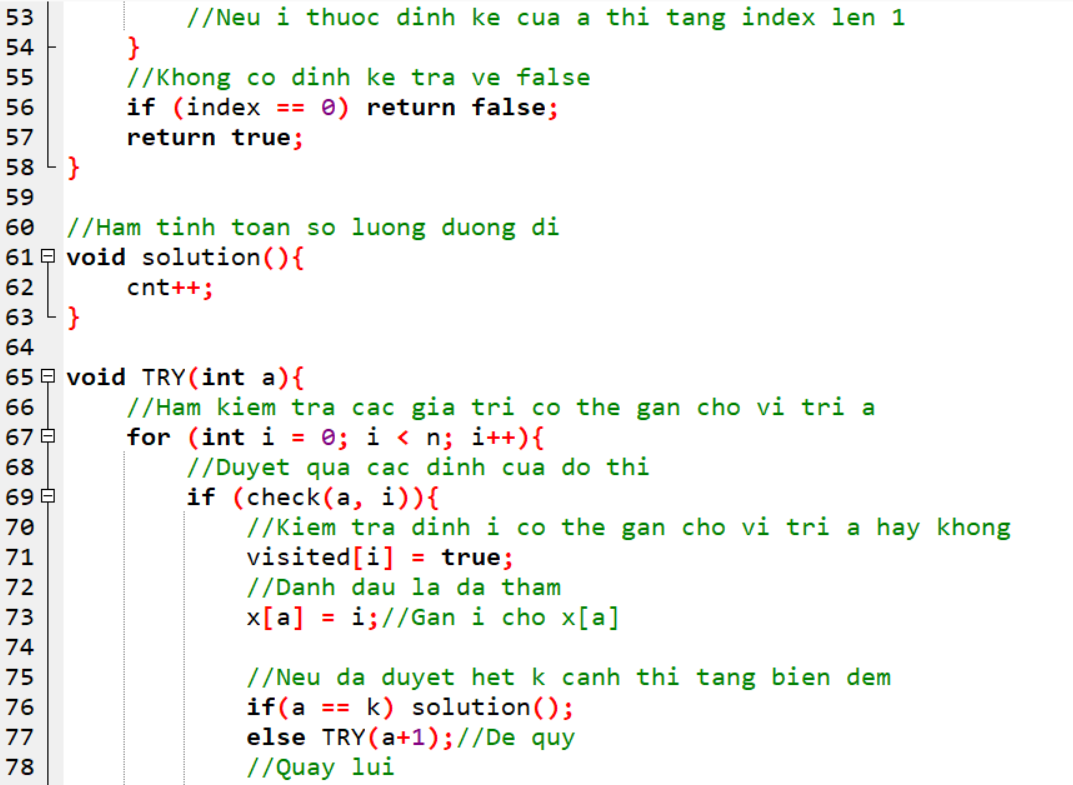
6

****

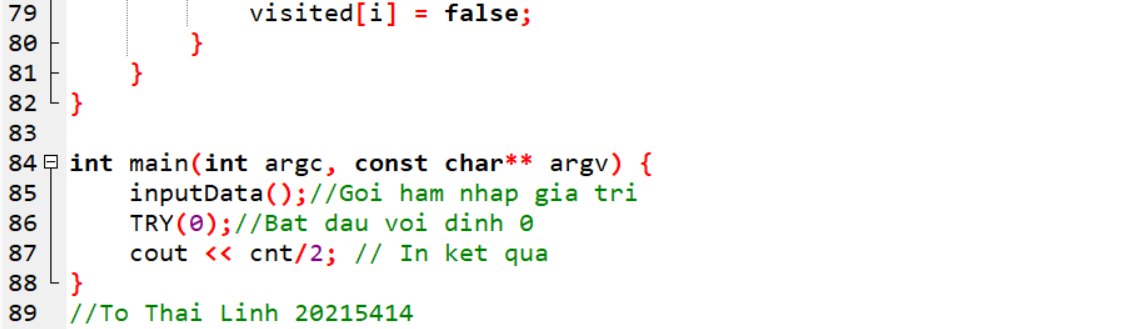
Hình : Bài tập 12 phần 1

****

Hình : Bài tập 12 phần 2

****

Hình : Bài tập 12 phần 3

****

Hình : Bài tập 12 phần 4

Case 1:  
Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Test case 1 bài tập 12 phần 1

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

Hình : Test case 1 bài tập 12 phần 2

Case 2:

Ảnh có chứa ảnh chụp màn hình, màu đen, thiết kế

Mô tả được tạo tự động

Hình : Test case 2 bài tập 12 phần 1

Ảnh có chứa ảnh chụp màn hình, màu đen, thiết kế

Mô tả được tạo tự động

Hình : Test case 2 bài tập 12 phần 2

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Test case 2 bài tập 12 phần 3

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Test case 2 bài tập 12 phần 4

Case 3:

Ảnh có chứa ảnh chụp màn hình, màu đen

Mô tả được tạo tự động

Hình : Test case 3 bài tập 12 phần 1

Ảnh có chứa ảnh chụp màn hình, màu đen

Mô tả được tạo tự động

Hình : Test case 3 bài tập 12 phần 2

Ảnh có chứa ảnh chụp màn hình, màu đen

Mô tả được tạo tự động

Hình : Test case 3 bài tập 12 phần 3

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Test case 3 bài tập 12 phần 4

Case 4:

Ảnh có chứa ảnh chụp màn hình, văn bản

Mô tả được tạo tự động

Hình : Test case 4 bài tập 12 phần 1

Ảnh có chứa ảnh chụp màn hình, màu đen, thiết kế

Mô tả được tạo tự động

Hình : Test case 4 bài tập 12 phần 2

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Test case 4 bài tập 12 phần 3

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Hình : Test case 4 bài tập 12 phần 4

Case 5:

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Test case 5 bài tập 12 phần 1

Ảnh có chứa ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

Hình : Test case 5 bài tập 12 phần 2

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Test case 5 bài tập 12 phần 3

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thuật in máy

Mô tả được tạo tự động

Hình : Test case 5 bài tập 12 phần 4

Case 6:  
Ảnh có chứa ảnh chụp màn hình, văn bản, màu đen

Mô tả được tạo tự động

Hình : Test case 6 bài tập 12 phần 1

Ảnh có chứa ảnh chụp màn hình, màu đen

Mô tả được tạo tự động

Hình : Test case 6 bài tập 12 phần 2

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Test case 6 bài tập 12 phần 3

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Test case 6 bài tập 12 phần 4

Ảnh có chứa ảnh chụp màn hình, văn bản, Phông chữ, số

Mô tả được tạo tự động

Hình : Test case 6 bài tập 12 phần 5

Case 7:

Ảnh có chứa ảnh chụp màn hình, văn bản, màu đen, thiết kế

Mô tả được tạo tự động

Hình : Test case 7 bài tập 12 phần 1

Ảnh có chứa ảnh chụp màn hình, màu đen

Mô tả được tạo tự động

Hình : Test case 7 bài tập 12 phần 2

Ảnh có chứa ảnh chụp màn hình, màu đen

Mô tả được tạo tự động

Hình : Test case 7 bài tập 12 phần 3

Ảnh có chứa ảnh chụp màn hình, màu đen

Mô tả được tạo tự động

Hình : Test case 7 bài tập 12 phần 4

Ảnh có chứa Phông chữ, ảnh chụp màn hình, văn bản, số

Mô tả được tạo tự động

Hình : Test case 7 bài tập 12 phần 5

Case 8:

Ảnh có chứa ảnh chụp màn hình, màu đen, thiết kế

Mô tả được tạo tự động

Hình : Test case 8 bài tập 12 phần 1

Ảnh có chứa ảnh chụp màn hình, văn bản

Mô tả được tạo tự động

Hình : Test case 8 bài tập 12 phần 2

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Test case 8 bài tập 12 phần 3

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Test case 8 bài tập 12 phần 4

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, số

Mô tả được tạo tự động

Hình : Test case 8 bài tập 12 phần 5

Case 9:

Ảnh có chứa ảnh chụp màn hình, màu đen, thiết kế

Mô tả được tạo tự động

Hình : Test case 9 bài tập 12 phần 1

Ảnh có chứa ảnh chụp màn hình, màu đen

Mô tả được tạo tự động

Hình : Test case 9 bài tập 12 phần 2

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

Hình : Test case 9 bài tập 12 phần 3

Ảnh có chứa ảnh chụp màn hình, màu đen

Mô tả được tạo tự động

Hình : Test case 9 bài tập 12 phần 4

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, thuật in máy

Mô tả được tạo tự động

Hình : Test case 9 bài tập 12 phần 5

/\* To Thai Linh 20215414

Ma lop thuc hanh 732826\*/

#include<bits/stdc++.h>

using namespace std;

const int MAX = 100;

int n, k;//n la so dinh, k la so canh di qua

int m;// m la so canh cua do thi

vector<vector<int> > vt; //Vector ma tran ke cua do thi

int x[MAX];//Luu duong di

bool visited[MAX];//Bien danh dau tham canh

int cnt;//Bien dem so cach di

//Ham nhap gia tri dau vao

void inputData(){

//Nhap gia tri cho n , k , m

cin >> n >> k;

cin >> m;

//Dat kich thuoc cua vector la n + 1

vt.resize(n+1);

//Nhap vao gia tri cac canh cua do thi

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

int u, v;

cin >> u >> v;

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

//them v - 1 vao danh sach canh ke cua u - 1

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

// them u - 1 vao danh sach canh ke cua v - 1

}

//Khoi tao cac bien tham la false

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

visited[i] = false;

}

//Khoi tao gia tri so luong duong di la 0

cnt = 0;

}

bool check(int a, int i){

//a la vi tri dang xet trong mang

//i la dinh duoc chon de dat tai a

if (a == 0) return true;

//Neu la dinh dau tien thi no la true

if (visited[i]) return false;

//Kiem tra dinh i, neu da tham tra ve false

int index = 0;

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

//Duyet qua cac dinh ke voi a

if (i == vt[x[a-1]][j]) index++;

//Neu i thuoc dinh ke cua a thi tang index len 1

}

//Khong co dinh ke tra ve false

if (index == 0) return false;

return true;

}

//Ham tinh toan so luong duong di

void solution(){

cnt++;

}

void TRY(int a){

//Ham kiem tra cac gia tri co the gan cho vi tri a

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

//Duyet qua cac dinh cua do thi

if (check(a, i)){

//Kiem tra dinh i co the gan cho vi tri a hay khong

visited[i] = true;

//Danh dau la da tham

x[a] = i;//Gan i cho x[a]

//Neu da duyet het k canh thi tang bien dem

if(a == k) solution();

else TRY(a+1);//De quy

//Quay lui

visited[i] = false;

}

}

}

int main(int argc, const char\*\* argv) {

inputData();//Goi ham nhap gia tri

TRY(0);//Bat dau voi dinh 0

cout << cnt/2; // In ket qua

}

//To Thai Linh 20215414