Mục lục nội dung

[Bài thực hành số 03 – Tuần 36 3](#_Toc166791386)

[BÀI TẬP TRÊN LAP 3](#_Toc166791387)

[**Bài tập 3.1.**Đệ quy – quay lui 3](#_Toc166791388)

[**Bài tập 3.2.** Đệ quy – quay lui 5](#_Toc166791389)

[**Bài tập 3.3.** Kỹ thuật nhánh cận 9](#_Toc166791390)

[**Bài tập 3.4.** Đệ quy có nhớ 12](#_Toc166791391)

[**Bài tập 3.5.** Tính tổ hợp 16](#_Toc166791392)

[**Bài tập 3.6.** Tìm ước chung lớn nhất 19](#_Toc166791393)

[**Bài tập 3.7.** Liệt kê xâu nhị phân 20](#_Toc166791395)

[**Bài tập 3.8.** Cân đĩa 23](#_Toc166791396)

[**BÀI TẬP VỀ NHÀ** 26](#_Toc166791397)

[**Bài tập 3.9.** Lập lịch cho y tá 26](#_Toc166791398)

[**Bài tập 3.10.** Khoảng cách Hamming 32](#_Toc166791399)

[**Bài tập 3.11.** Lịch trình chụp ảnh 35](#_Toc166791400)

[**Bài tập 3.12.** Đếm đường đi 39](#_Toc166791401)

Mục lục hình ảnh

[Hình 1 Bài 3.1 Đệ quy – quay lui](#_Toc161238452)

[Hình 2 Bài 3.2 Đệ quy – quay lui](#_Toc161238453)

[Hình 3 Bài 3.3 Kỹ thuật nhánh cận](#_Toc161238454)

[Hình 4 Bài 3.4 Đệ quy có nhớ](#_Toc161238455)

[Hình 5 Bài 3.5 Tính tổ hợp](#_Toc161238456)

[Hình 6 Bài 3.6 Tìm UCLN](#_Toc161238457)

[Hình 7 Bài 3.7 Liệt kê xâu nhị phân](#_Toc161238458)

[Hình 8 Bài 3.8 Cân đĩa](#_Toc161238458)

[Hình 9 Bài 3.9 Lập lịch cho y tá](#_Toc161238459)

[Hình 10 Bài 3.10 Khoảng cách Hamming](#_Toc161238459)

[Hình 11 Bài 3.11 Lịch trình chụp ảnh](#_Toc161238459)

[Hình 12 Bài 3.12 Đếm đường đi](#_Toc161238459)

# Bài thực hành số 03 – Tuần 36

# BÀI TẬP TRÊN LAP

## **Bài tập 3.1.**Đệ quy – quay lui

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

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Hình 1 Bài 3.1

|  |
| --- |
| int lucas(int n) {  if ( n==0 )  return 2;  else if ( n==1 )  return 1;  else  return lucas(n-1)+lucas(n-2);  } |

## **Bài tập 3.2.** Đệ quy – quay lui

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).

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Hình 2 Bài 3.2

|  |
| --- |
| #include <iostream>  using namespace std;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  int n;  int X[100], Y[100]; /// Luu toa do cac buoc di chuyen cua quan ma  int mark[100][100]; /// Danh dau cac o ma quan ma co the 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);  }  int checkbien(int x, int y){  if ( x<1 || x>n || y<1 || y>n )  return 0;  else  return 1;  }  /// Thuat toan 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 ( mark[xx][yy] == 1 || checkbien(xx,yy) == 0 ){  continue;  cout << "Here";  }  else {  ///cout << "Hello i = " << i << " k = " << k << endl ;  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()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r" ) ){  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  cin >> n;  mark[1][1] = 1;  X[1] =1;  Y[1] = 1;  TRY(2);  return 0;  } |

## **Bài tập 3.3.** Kỹ thuật nhánh cận

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.

A screenshot of a computer

Description automatically generated

Hình 3 Bài 3.3

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  #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 hai thanh pho khac nhau  int best = INT\_MAX; //# tong chi phi nho nhat can tim, ban dau dat bang 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 giu cac thanh pho da di  //# Doc du lieu vao  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]);  }  }  //# Thuat toan quay lui  void TRY(int k)  {  for(int i = 2; i <= n; i++)  {  if ( mark[i] == 0 && i!=x[k-1] ){  x[k] = i;  mark[x[k]] = 1;  curr += c[x[k-1]][x[k]];  if ( k == n ){  best = min(best, curr+c[x[k]][1]);  }  else  TRY(k+1);  mark[x[k]] = 0;  curr -= c[x[k-1]][x[k]];  }  }  }  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r" ) ){  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  input();  x[1] = 1;  TRY(2);  cout << best;  return 0;  } |

## **Bài tập 3.4.** Đệ quy có nhớ

Cho dãy a có n phần tử. Một dãy con của aa 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 aa

A screenshot of a computer

Description automatically generated

Hình 4 Bài 3.4

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  int a[1000], n;  int mem[1000]; //# mang ghi nho loi giai cac bai toan con da duoc giai  void init()  {  memset(mem, -1, sizeof(mem));  }  //# quy hoach dong  //# Hàm lis(i) tra ve do dài dãy con tăng dài nhat ket thúc boi a[i]  int lis(int i)  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  # YOUR CODE HERE #  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  for ( int k=0 ; k<i ; k++ ){  if ( a[k] < a[i] )  mem[i] = max(mem[i],mem[k]+1);  }  if ( mem[i] == -1 ){  mem[i] = 1;  return 1;  }  else 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()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r" ) ){  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  init();  cin >> n;  for(int i = 0; i < n; i++) cin >> a[i];  int res = 1, pos = 0;  for(int i = 0; i < n; i++)  {  if ( res < lis(i) )  {  res = lis(i);  pos = i;  //cout << "i = " << i << " res = " << res << endl;  }  }  cout << res << endl;  trace(pos);  return 0;  } |

## **Bài tập 3.5.** Tính tổ hợp

Hãy giải các bài toán sau đây bằng phương pháp khử đệ quy

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

Hình 5 Bài 3.5

|  |
| --- |
| #include <iostream>  using namespace std;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  int c[100][100];  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)  {  //# Khu de quy  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  # YOUR CODE HERE #  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  if ( k==0 || n==k )  c[k][n] = 1;  else{  if ( c[k][n-1] == 0 )  c[k][n-1] = binom2(n-1,k);  if ( c[k-1][n-1] == 0 )  c[k-1][n-1] = binom2(n-1,k-1);  c[k][n] = c[k][n-1] + c[k-1][n-1];  }  return c[k][n];  }  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r" ) )  {  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  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;  } |

## **Bài tập 3.6.** Tìm ước chung lớn nhất

Tính ước chung lớn nhất của hai số cho trước A screenshot of a computer

Description automatically generated

Hình 6 Bài 3.6

|  |
| --- |
| #include <iostream>  #include <algorithm>  using namespace std;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  int gcd(int a, int b)  {  if (b ==0) return a;  return gcd(b, a% b);  }  int gcd2(int a, int b)  {  //# Khu de quy  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  # YOUR CODE HERE #  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  int tmp;  while ( b!= 0 ){  tmp=a;  a=b;  b=tmp%b;  }  return a;  }  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r" ) ){  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  int a, b;  cin >> a >> b;  cout << gcd(a, b) << endl << gcd2(a, b);  return 0;  } |

## **Bài tập 3.7.** 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

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Hình 7 Bài 3.7

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  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()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r" ) )  {  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  int n, k;  cin >> n >> k;  int x[n+1];  stack<state> s;  //# number of consecutive suffix 1  int L = 0;  s.push(state(1, 0));  while (!s.empty())  {  state &top = s.top();  //# if a new binary sequence is found  if (top.i > n)  {  for (int i = 1; i <= n; ++i)  cout << x[i] << " \n"[i == n];  s.pop();  continue;  }  //# Khu de quy  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;  if(top.j)  {  L = L + 1;  }  else  {  L = 0;  }  s.push(state(top.i+1,0));  }  top.j++;  }  return 0;  } |

## **Bài tập 3.8.** 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.

A screenshot of a computer program

Description automatically generated

Hình 8 Bài 3.8

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  struct state  {  /// Qua nang thu i mang trang thai j  /// j = 0 : khong duoc chon, j = -1 : qua nang mang dau -, j = 1 : qua nang mang dau +  int i, j;  state(int \_i = 0, int \_j = 0): i(\_i), j(\_j) {}  };  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r" ) ){  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  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;  //# tong cac qua nang duoc chon  int sum = 0;  s.push(state(1, -1));  while ( !s.empty() )  {  state &top = s.top();  if (top.i > n)  {  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);  }  s.pop();  continue;  }  //# Khu đe quy  /// Khi top.j > -1 tuc la bang 0 hoac 1 =>  /// trang thai do da duoc duyet qua =>  /// khoi phuc tong sum  if ( top.j > - 1 ){  sum = sum - m[top.i]\*x[top.i];  }  /// Khi top.j khong thuoc {-1,0,1} => loai  if ( top.j > 1 ){  s.pop();  continue;  }  /// Cap nhat  x[top.i] = top.j;  sum = sum + m[top.i]\*x[top.i]; /// cap nhat sum  s.push({top.i+1,-1}); /// push trang thai co so voi j=-1  top.j++;///sum da duoc cap nhat, cac phep toan duoc luu vao x[i]  /// => top.j++ ( khong anh huong )  }  cout << -1;  return 0;  } |

# **B****ÀI TẬP VỀ NHÀ**

Chụp ảnh kết quả của tất cả các test.

## **Bài tập 3.9.** 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.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Hình 9 Bài 3.9

|  |
| --- |
| #include<bits/stdc++.h>  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  using namespace std;  int n,k1,k2;  int x[250];  int cnt = 0;  void ghinhan(){  for(int i=1; i<=n; i++)  cout << x[i];  cout << endl;  cnt++;  }  int cnt1 = 0;  bool check1(int a, int i){  if ( a==1 )  return true;  if ( i==0 ){  /// Neu ghep vao day ngay nghi  if ( x[a-1] == 0 )  return false;  else{  /// neu nghir khi chua lam viec du k1 ngay thi sai  if ( cnt1 < k1 )  return false;  }  }  else {  /// Neu ghep vao day ngay di lam  if ( x[a-1] == 0 ){  /// Neu so ki tu con lai khong du de lap k1 thi false  if ( n-a+1 < k1 )  return false;  }  /// neu da di lam k2 hoac hoac hon roi thi phai nghir  else if ( cnt1 >=k2 ) return false;  }  /// con lai thi return true  return true;  }  /// Kiem tra xem ngay a co the 0/1 : nghi/hay lam  bool check(int a, int i){  if ( a==1 )  return true;  /// Neu ngay truoc do da nghi  if ( x[a-1] == 0 ){  /// thi khong duoc nghi nua  if ( i==0 )  return false;  /// nguoc lai neu di lam ma so ngay con lai khong du k1 ngay di lam  ///= > loi  /// nhanh can !!!  else if ( n-a+1 < k1 ) return false;  }  /// neu ngay truoc do da di lam  else if ( x[a-1] == 1 ) {  /// ma hom nay muon nghi  if ( i==0 ){  /// nhung chua di lam du k1 ngay => loi  if ( cnt1 <k1 )  return false;  }  /// neu di lam hom nay  else if ( i==1 ){  /// ma so ngay di lam da dat dinh => phai nghi  if ( cnt1 >= k2 )  return false;  }  }  /// con lai thi return true  return true;  }  void TRY(int a){  for(int i=0; i<=1; i++){  if( check(a,i) ) {  x[a] = i;  int tmp = cnt1;  if ( i==1 ){  if ( x[a-1] == 1 )  cnt1++;  else  cnt1=1;  }  else cnt1 = 0;  if ( a==n )  ghinhan();  else  TRY(a+1);  cnt1 = tmp;  }  }  }  int main(){  #define taskname "hello123"  if ( fopen(taskname".inp","r" ) ){  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  ios\_base::sync\_with\_stdio(false);  cin.tie(nullptr);  cout.tie(nullptr);  cin >> n >> k1 >> k2;  TRY(1);  return 0;  } |

## **Bài tập 3.10.** 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 screenshot of a computer program

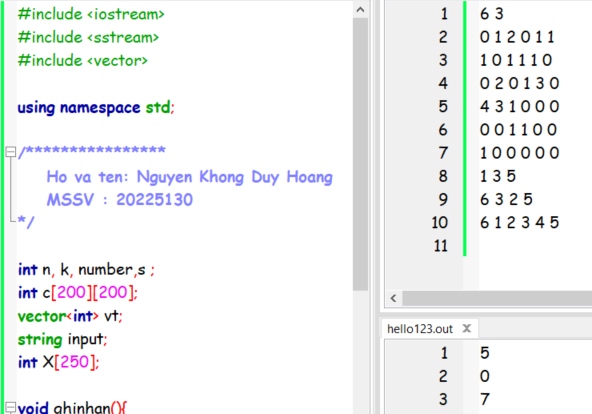
Description automatically generated

Hình 10 Bài 3.10

|  |
| --- |
| #include <iostream>  using namespace std;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  int t;  int n,h;  int hamming = 0;  int x[20];  void ghinhan(){  for ( int i=1 ; i<=n ; i++ )  cout << x[i];  cout << "\n";  }  bool check( int a, int i ){  if ( i==1 && hamming >= h )  return false;  return true;  }  void TRY(int a){  for ( int i=0 ; i<=1 ; i++ ){  if ( check(a,i) ){  x[a] = i;  if ( i==1 )  hamming++;  if ( a==n ){  if ( hamming == h )  ghinhan();  }  else  TRY(a+1);  if ( i==1 )  hamming--;  }  /// nhanh can : neu hamming da > h thi bo qua tat ca truong hop ve sau  if ( hamming > h )  continue;  }  }  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r" ) ){  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  ios\_base::sync\_with\_stdio(false);  cin.tie(nullptr);  cout.tie(nullptr);  cin >> t;  for ( int t1 = 1 ; t1<=t ; t1++ ){  cin >> n >> h;  for ( int i=0 ; i<20 ; i++ )  x[i] = 0;  hamming = 0;  TRY(1);  cout << "\n";  }  return 0;  } |

## **Bài tập 3.11.** 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 screenshot of a computer

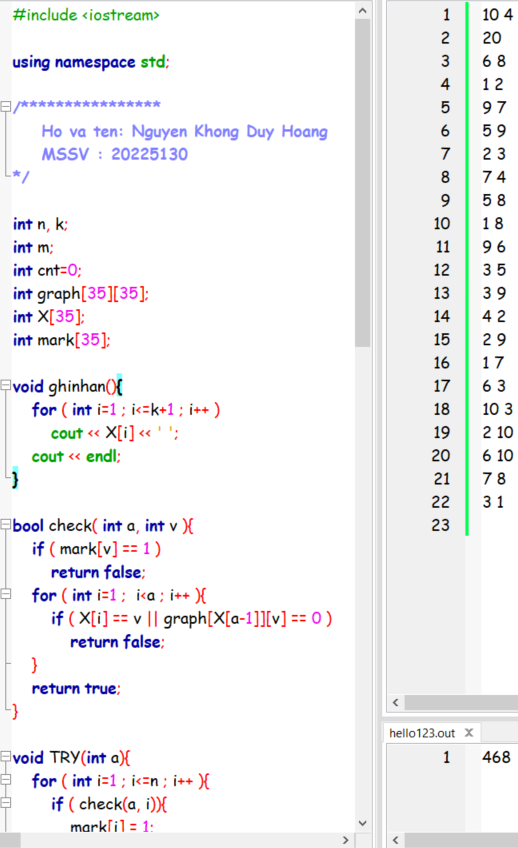
Description automatically generated

Hình 11 Bài 3.11

|  |
| --- |
| #include <iostream>  #include <sstream>  #include <vector>  using namespace std;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  int n, k, number,s ;  int c[200][200];  vector<int> vt;  string input;  int X[250];  void ghinhan(){  for ( int i=1 ; i<=s ; i++ )  cout << X[i] << ' ';  cout << endl;  }  bool check( int a, int v ){  for ( int i=1 ; i<a ; i++ ){  if ( X[i] == v || v == X[s] || c[X[a-1]][v] == 0 )  return false;  }  return true;  }  int curr = 0, best = INT\_MAX;  void TRY(int a){  for ( int i=0 ; i<s ; i++ ){  if ( check(a, vt[i])){  X[a] = vt[i];  curr = curr + c[X[a-1]][X[a]];  if ( a==s-1 ){  ///ghinhan();  if ( c[X[a]][X[s]] != 0 )  best = min(best, curr + c[X[a]][X[s]]);  }  else  TRY(a+1);  curr = curr - c[X[a-1]][X[a]];  }  }  }  int main()  {  #define taskname "hello123"  if ( fopen(taskname".inp","r" ) ){  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  ios\_base::sync\_with\_stdio(false);  cin.tie(nullptr);  cout.tie(nullptr);  cin >> n >> k;  for ( int i=1 ; i<=n ; i++ ){  for ( int j=1 ; j<=n ; j++ )  cin >> c[i][j];  }  cin.ignore(); // Ignore the newline character after reading k  for ( int i=1 ; i<=k ; i++ ){  fill(begin(X), end(X), 0);  getline(cin, input);  stringstream ss(input);  while ( ss >> number )  vt.push\_back(number);  s = vt.size();  s = vt.size();  if (s == 0) {  cout << 0 << endl;  continue;  }  X[1] = vt[0];  X[s] = vt[s-1];  curr = 0;  best = INT\_MAX;  ///cout << X[1] << ' ' << X[s] << endl;  TRY(2);  if ( best == INT\_MAX )  cout << 0;  else  cout << best;  cout << endl;  vt.clear();  }  return 0;  } |

## **Bài tập 3.12.** 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 generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Hình 12 Bài 3.12

|  |
| --- |
| #include <iostream>  using namespace std;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Ho va ten: Nguyen Khong Duy Hoang  MSSV : 20225130  \*/  int n, k;  int m;  int cnt=0;  int graph[35][35];  int X[35];  int mark[35];  void ghinhan(){  for ( int i=1 ; i<=k+1 ; i++ )  cout << X[i] << ' ';  cout << endl;  }  bool check( int a, int v ){  if ( mark[v] == 1 )  return false;  for ( int i=1 ; i<a ; i++ ){  if ( X[i] == v || graph[X[a-1]][v] == 0 )  return false;  }  return true;  }  void TRY(int a){  for ( int i=1 ; i<=n ; i++ ){  if ( check(a, i)){  mark[i] = 1;  X[a] = i;  if ( a==k+1 ){  //ghinhan();  cnt++;  }  else  TRY(a+1);  mark[i] = 0;  }  }  }  int main(){  #define taskname "hello123"  if ( fopen(taskname".inp","r" ) ){  freopen(taskname".inp","r",stdin);  freopen(taskname".out","w",stdout);  }  ios\_base::sync\_with\_stdio(false);  cin.tie(nullptr);  cout.tie(nullptr);  for ( int i=0 ; i<33 ; i++ )  for ( int j=0 ; j<33 ; j++ )  graph[i][j] = 0;  fill(begin(mark),end(mark),0);  cin >> n >> k >> m;  int a, b;  for ( int i=0 ; i<m ; i++ ){  cin >> a >> b;  graph[a][b] = 1;  graph[b][a] = 1;  }  /\*\*  for ( int i=1 ; i<=n ; i++ )  for ( int j=1 ; j<=n ; j++ )  cout << graph[i][j] << " \n"[j==n];  \*/  TRY(1);  cout << cnt/2;  return 0;  } |