

BÁO CÁO BUỔI THỰC HÀNH SỐ 3

Bộ Môn Kỹ Thuật Lập Trình



Sinh viên: 20235050 – Hồ Minh Dũng

Giáo viên hướng dẫn: ThS. Lê Thị Hoa

HTGD: Nguyễn Kim Cường

Contents

Bài thực hành số 3 – Tuần 36	4
Bài tập 1: Tính dãy Lucas.....	4
Bài tập 2: 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).	5
Bài tập 3: Bài toán người du lịch.....	7
Bài tập 4: LIS.....	9
Bài tập 5: Tính tổ hợp	11
Bài tập 6: Tìm ước chung lớn nhất	13
Bài tập 7: Liệt kê xâu nhị phân.....	14
Bài tập 8: Cân đĩa	17
Bài tập 9: Lập lịch cho y tá	18
Bài tập 10: Khoảng cách Hamming.....	19
Bài tập 11: Lịch trình chụp ảnh	20
Bài tập 12: Đếm đường đi	22

Figure

Hình 1. 1. Code bài 1.....	4
Hình 1. 2. Test case bài 1.....	5
Hình 2. 1: Code bài 2.....	6
Hình 2. 2: Code bài 2.....	6
Hình 2. 3. Test case bài 2.....	7
Hình 3. 1. Code bài 3.....	8
Hình 3. 2. Code bài 3.....	8
Hình 3. 3. Test case bài 3.....	9
Hình 4. 1. Code bài 4.....	9
Hình 4. 2. Code bài 4.....	10
Hình 4. 3. Test case bài 4.....	10
Hình 5. 1. Code bài 5.....	11
Hình 5. 2. Code bài 5.....	11
Hình 5. 3. Test case bài 5.....	12
Hình 5. 4. Test case bài 5.....	13
Hình 6. 1. Code bài 6.....	13
Hình 6. 2. Test case bài 6.....	14
Hình 7. 1. Code bài 7.....	15
Hình 7. 2. Code bài 7.....	15
Hình 7. 3. Test case bài 7.....	16
Hình 7. 4. Test case bài 7.....	17
Hình 8. 1. Test case bài 8.....	18
Hình 8. 2. Test case bài 8.....	18
Hình 9. 1. Test case 1 và 2 bài 9.....	19
Hình 9. 2. Test case 3 bài 9.....	19
Hình 10. 1. Test case bài 10.....	20
Hình 11. 1. Test case 1 bài 11.....	21
Hình 11. 2. Test case 1 bài 11.....	21
Hình 11. 3. Test case 1 bài 11.....	22
Hình 11. 4. Test case 2 bài 11.....	22
Hình 12. 1 Test case 1 bài 12.....	23
Hình 12. 2. Test case 2 bài 12.....	23
Hình 12. 3. Test case 3 bài 12.....	24
Hình 12. 4. Test case 4 bài 12.....	24
Hình 12. 5 Test case 5 bài 12.....	25
Hình 12. 6 Test case 6 bài 12.....	25
Hình 12. 7. Test case 7 bài 12.....	26
Hình 12. 8. Test case 8 bài 12.....	26
Hình 12. 9. Test case 9 bài 12.....	27

Bài thực hành số 3 – Tuần 36

Bài tập 1: Tính dãy Lucas

Dãy Lucas được định nghĩa bởi $L_n = L_{n-1} + L_{n-2}$ và bắt đầu bởi $L_0=2$, $L_1=1$. Viết hàm tính số Lucas thứ n .

The screenshot shows a terminal window with two panes. The left pane contains the C++ code for calculating the Lucas sequence. The right pane shows the command line and its output.

```
Bai3_1_Tuan36_Lab3.cpp > lucas(int)
1  /* Bai 3.1 Dãy Lucas được định nghĩa bởi Ln=Ln-1+Ln-2
2  và bắt đầu bởi L0=2, L1=1. Viết hàm tính số Lucas thứ n.
3  Hồ Minh Dũng - 20235050 - 750829 - Tuan 36*/
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 int lucas(int n_50){
8     int a_50[100] = {0};
9     a_50[0] = 2;
10    a_50[1] = 1;
11    if (n_50 < 2) return a_50[n_50];
12    for (int i_50 = 2; i_50 <= n_50; i_50++){
13        a_50[i_50] = a_50[i_50 - 1] + a_50[i_50 - 2];
14    }
15    return a_50[n_50];
16 }
17
18 int main(){
19     cout << "Hồ Minh Dũng - 5050\n";
20     cout << "input = ";
21     int n; cin >> n;
22     cout << "output = " << lucas(n);
23 }
24
```

```
Ky_thuat_lap_trinh\Dung\Lab02_Tuan36> cd "g:\HUST\Ky_thuat_lap_trinh\K
Dung\Lab02_Tuan36"
1 if ($?) { g++ Ba13_1_Tuan36_Lab3.cpp -o Ba13_1_
Tuan36_Lab3 } ; if ($?) { ./Ba13_1_Tuan36_Lab3 }
Hồ Minh Dũng - 5050
input = 5
output = 11
PS G:\HUST\Ky_thuat_lap_trinh\K
Dung\Lab02_Tuan36>
```

Hình 1. 1. Code bài 1.

Answer: (penalty regime: 10, 20, ... %)

```
1 /* Bai 2.1 Dãy Lucas được định nghĩa bởi Ln=Ln-1+Ln-2
2 và bắt đầu bởi L0=2, L1=1. Viết hàm tính số Lucas thứ n.
3 Hồ Minh Dũng - 20235050 - 750829 - Tuan 34 */
4
5 int lucas(int n_50){
6     int a_50[100] = {0};
7     a_50[0] = 2;
8     a_50[1] = 1;
9     if (n_50 < 2) return a_50[n_50];
10    for (int i_50 = 2; i_50 <=n_50; i_50++){
11        a_50[i_50] = a_50[i_50 - 1] + a_50[i_50 - 2];
12    }
13    return a_50[n_50];
14 }
15
16 }
```

Precheck

Check

	Test	Expected	Got	
✓	cout << lucas(5);	11	11	✓
✓	cout << lucas(10);	123	123	✓
✓	cout << lucas(30);	1860498	1860498	✓

Passed all tests! ✓

Hình 1. 2. Test case bài 1.

Bài tập 2: Trên bàn cờ vua kích thước $n \times n$ 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 ($\text{ô } (1, 1)$ được xem là đã đi qua).

Hồ Minh Dũng - 20235050

```

1 // Bai3_2_Tuan36_Lab3.cpp > ...
2 /* Bai 3.2 Trên bàn cờ vua kích thước n*n có một quân mã đang ở ô (1, 1).
3 Hãy đưa ra một dãy các di chuyển của mã sao cho mỗi ô trên bàn cờ đều được
4 đi qua đúng 1 lần (ô (1, 1) được xem là đã đi qua).
5 Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
6 #include <iostream>
7 using namespace std;
8
9 int n_50;
10 int X_50[100], Y_50[100]; // Lưu tọa độ các bước di chuyển của quân mã
11 int mark_50[100][100]; // Đánh dấu vị trí các ô mà quân mã đã di chuyển qua
12
13 const int hx_50[] = {1, 1, 2, 2, -1, -1, -2, -2};
14 const int hy_50[] = {2, -2, 1, -1, 2, -2, 1, -1};
15
16 // In ra dãy các di chuyển tìm được
17 void print_sol(){
18     for (int j_50 = 1; j_50 <= n_50 * n_50; ++j_50)
19         printf("%d %d\n", X_50[j_50], Y_50[j_50]);
20     exit(0);
21 }
22
23 // Thuật toán quay lui
24 void TRY(int k_50){
25     for(int i_50 = 0; i_50 < 8; i_50++){
26         int xx_50 = X_50[k_50-1] + hx_50[i_50];
27         int yy_50 = Y_50[k_50-1] + hy_50[i_50];
28         ****
29         # YOUR CODE HERE #
30         ****
31         if (xx_50 >= 1 && xx_50 <= n_50 && yy_50 >= 1 && yy_50 <= n_50 && !mark_50[xx_50][yy_50]){
32             X_50[k_50] = xx_50;
33             Y_50[k_50] = yy_50;
34             mark_50[xx_50][yy_50] = 1;
35             if (k_50 == n_50 * n_50) print_sol();
36             else TRY(k_50 + 1);
37             mark_50[xx_50][yy_50] = 0;
38         }
39     }
40 }
41
42 int main(){
43     cout << "Ho Minh Dũng - 5050\n";
44     cin >> n_50;
45     mark_50[1][1] = 1;
46     X_50[1] = Y_50[1] = 1;
47     TRY(2);
48     return 0;
49 }

```

Hình 2. 1: Code bài 2.

```

1 // Bai3_2_Tuan36_Lab3.cpp > ...
2
3 void print_sol(){
4     for (int j_50 = 1; j_50 <= n_50 * n_50; ++j_50)
5         printf("%d %d\n", X_50[j_50], Y_50[j_50]);
6     exit(0);
7 }
8
9 // Thuật toán quay lui
10 void TRY(int k_50){
11     for(int i_50 = 0; i_50 < 8; i_50++){
12         int xx_50 = X_50[k_50-1] + hx_50[i_50];
13         int yy_50 = Y_50[k_50-1] + hy_50[i_50];
14         ****
15         # YOUR CODE HERE #
16         ****
17         if (xx_50 >= 1 && xx_50 <= n_50 && yy_50 >= 1 && yy_50 <= n_50 && !mark_50[xx_50][yy_50]){
18             X_50[k_50] = xx_50;
19             Y_50[k_50] = yy_50;
20             mark_50[xx_50][yy_50] = 1;
21             if (k_50 == n_50 * n_50) print_sol();
22             else TRY(k_50 + 1);
23             mark_50[xx_50][yy_50] = 0;
24         }
25     }
26 }
27
28 int main(){
29     cout << "Ho Minh Dũng - 5050\n";
30     cin >> n_50;
31     mark_50[1][1] = 1;
32     X_50[1] = Y_50[1] = 1;
33     TRY(2);
34     return 0;
35 }

```

Hình 2. 2: Code bài 2.

Answer: (penalty regime: 10, 20, ... %)

Time left 23:50:15

```

1 /* Bai 2.2 Trên bàn cờ vua kích thước n*n có một quân mã đang ở ô (1, 1).
2 Hãy đưa ra một dãy các di chuyển của mã sao cho mỗi ô trên bàn cờ đều được
3 đi qua đúng 1 lần (ô (1, 1) được xem là đã đi qua).
4 Hồ Minh Dũng - 20235050 - 750829 - Tuan 36*/
5 #include <iostream>
6 using namespace std;
7
8 int n_50;
9 int X_50[100], Y_50[100]; // Lưu tọa độ các bước di chuyển của quân mã
10 int mark_50[100][100]; // Đánh dấu vị trí các ô mà quân mã đã di chuyển qua
11
12 // Mảng hx, hy mô tả 8 vị trí quân mã có thể di chuyển kể từ vị trí hiện tại
13 const int hx_50[] = {1, 1, 2, 2, -1, -1, -2, -2};
14 const int hy_50[] = {2, -2, 1, -1, 2, -2, 1, -1};
15
16 // In ra dãy các di chuyển tìm được
17 void print_sol(){
18     for (int j_50 = 1; j_50 <= n_50 * n_50; ++j_50)
19         printf("(%d %d)\n", X_50[j_50], Y_50[j_50]);
20     exit(0);
21 }
22

```

Precheck Check

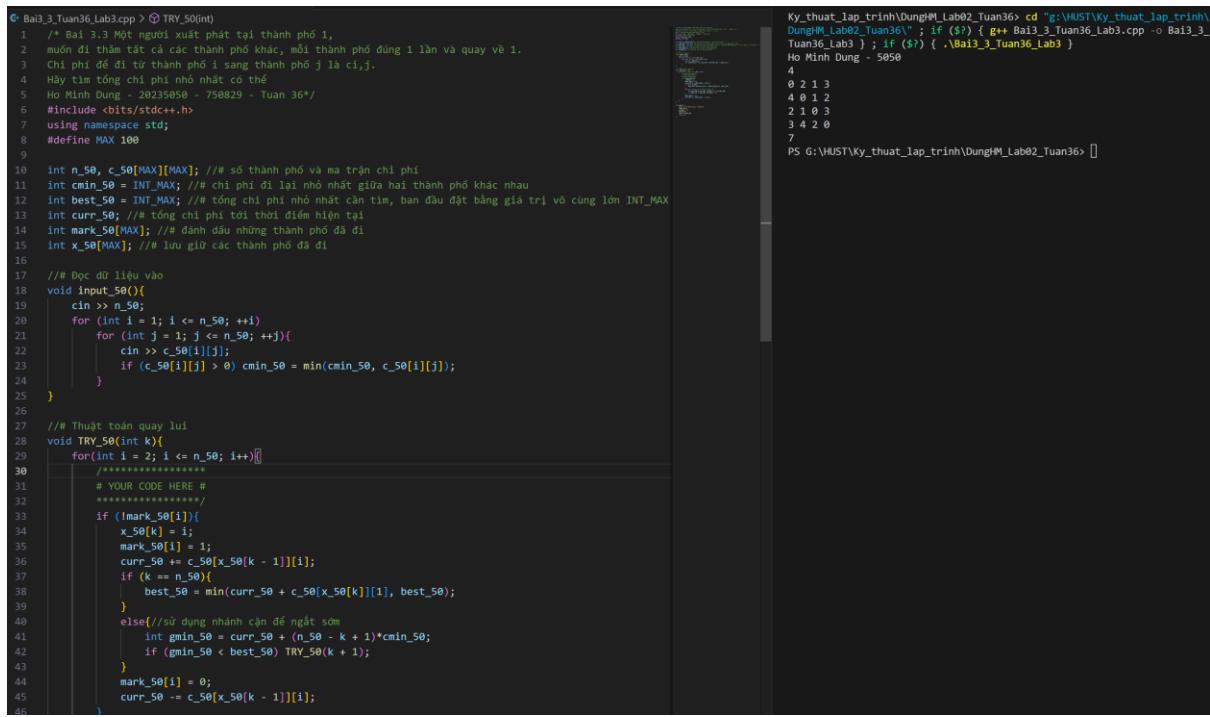
	Input	Expected	Got	
✓	5	(1 1) (2 3) (3 5) (5 4) (4 2) (2 1) (3 3) (1 4)	(1 1) (2 3) (3 5) (5 4) (4 2) (2 1) (3 3) (1 4)	✓

Hình 2. 3. Test case bài 2

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 $c_{i,j}$. Hãy tìm tổng chi phí nhỏ nhất có thể

Hồ Minh Dũng - 20235050

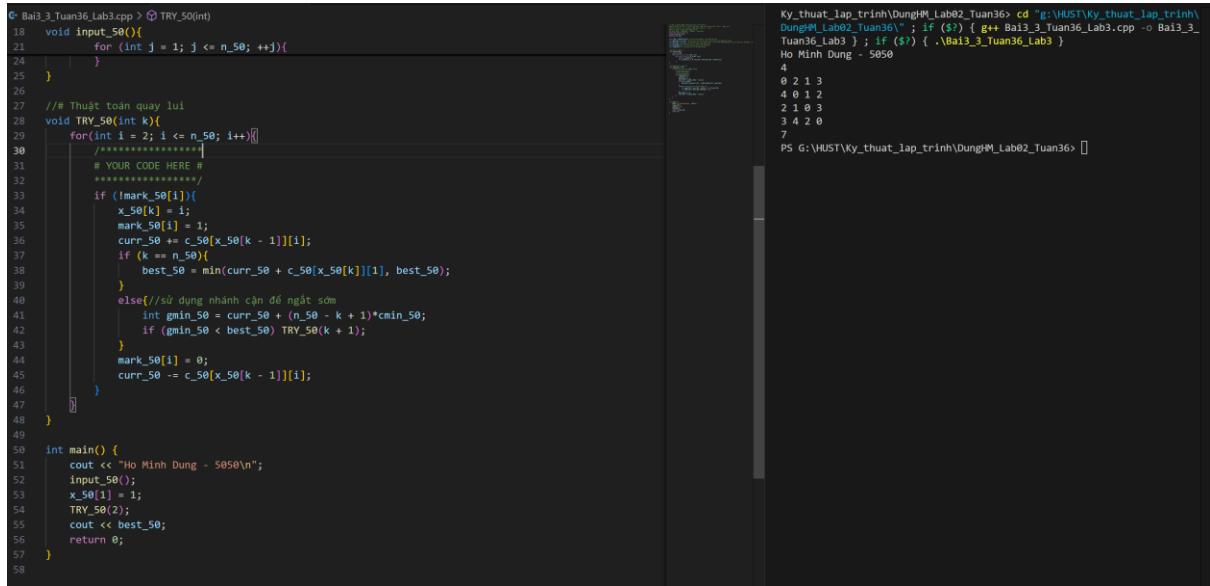


```

1 // Bai3_3_Tuan36_Lab3.cpp > TRY_50(int)
2 /* Bai 3.3 Một người xuất phát tại thành phố 1,
3 muốn đi thăm tất cả các thành phố khác, mỗi thành phố dùng 1 lần và quay về 1.
4 Chỉ phí để di từ thành phố i sang thành phố j là c[i,j].
5 Hãy tìm tổng chi phí nhỏ nhất có thể
6 Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
7 #include <iostream>
8 using namespace std;
9 #define MAX 100
10 int n_50, c_50[MAX][MAX]; //## số thành phố và ma trận chi phí
11 int cmin_50 = INT_MAX; //## chi phí lại nhỏ nhất giữa hai thành phố khác nhau
12 int best_50 = INT_MAX; //## tổng chi phí nhỏ nhất cần tìm, ban đầu đặt bằng giá trị vô cùng lớn INT_MAX
13 int curr_50; //## tổng chi phí tới thời điểm hiện tại
14 int mark_50[MAX]; //## đánh dấu những thành phố đã đi
15 int x_50[MAX]; //## lưu giữ các thành phố đã đi
16
17 //## Đọc dữ liệu vào
18 void input_50(){
19     cin >> n_50;
20     for (int i = 1; i <= n_50; ++i){
21         for (int j = 1; j <= n_50; ++j){
22             cin >> c_50[i][j];
23             if (c_50[i][j] > 0) cmin_50 = min(cmin_50, c_50[i][j]);
24         }
25     }
26
27 //## Thuật toán quay lui
28 void TRY_50(int k){
29     for(int i = 2; i <= n_50; i++) {
30         //*****#
31         # YOUR CODE HERE #
32         //*****#
33         if (!mark_50[i]){
34             x_50[k] = i;
35             mark_50[i] = 1;
36             curr_50 += c_50[x_50[k - 1]][i];
37             if (k == n_50){
38                 best_50 = min(curr_50 + c_50[x_50[k]][1], best_50);
39             }
40             else{//sử dụng nhánh cận để ngắt sớm
41                 int gmin_50 = curr_50 + (n_50 - k + 1)*cmin_50;
42                 if (gmin_50 < best_50) TRY_50(k + 1);
43             }
44             mark_50[i] = 0;
45             curr_50 -= c_50[x_50[k - 1]][i];
46         }
47     }
48 }
49
50 int main() {
51     cout << "Ho Minh Dũng - 5050\n";
52     input_50();
53     x_50[1] = 1;
54     TRY_50(2);
55     cout << best_50;
56     return 0;
57 }

```

Hình 3. 1. Code bài 3.



```

1 // Bai3_3_Tuan36_Lab3.cpp > TRY_50(int)
2 void input_50(){
3     for (int j = 1; j <= n_50; ++j){
4     }
5
6 //## Thuật toán quay lui
7 void TRY_50(int k){
8     for(int i = 2; i <= n_50; i++) {
9         //*****#
10        # YOUR CODE HERE #
11        //*****#
12        if (!mark_50[i]){
13            x_50[k] = i;
14            mark_50[i] = 1;
15            curr_50 += c_50[x_50[k - 1]][i];
16            if (k == n_50){
17                best_50 = min(curr_50 + c_50[x_50[k]][1], best_50);
18            }
19            else{//sử dụng nhánh cận để ngắt sớm
20                int gmin_50 = curr_50 + (n_50 - k + 1)*cmin_50;
21                if (gmin_50 < best_50) TRY_50(k + 1);
22            }
23            mark_50[i] = 0;
24            curr_50 -= c_50[x_50[k - 1]][i];
25        }
26    }
27
28 int main() {
29     cout << "Ho Minh Dũng - 5050\n";
30     input_50();
31     x_50[1] = 1;
32     TRY_50(2);
33     cout << best_50;
34     return 0;
35 }

```

Hình 3. 2. Code bài 3.

Answer: (penalty regime: 10, 20, ... %)

```

1. /* Bai 3.3 Một người xuất phát tại thành phố 1,
2. muốn đi thăm tất cả các thành phố khác, mỗi thành phố đúng 1 lần và quay về 1.
3. Chi phí di từ thành phố i sang thành phố j là c[i,j].
4. Hãy tìm tổng chi phí nhỏ nhất có thể
5. Hồ Minh Dũng - 20235050 - 750829 - Tuan 36*/
6. #include <bits/stdc++.h>
7. using namespace std;
8. #define MAX 100
9.
10. int n_50, c_50[MAX][MAX]; // số thành phố và ma trận chi phí
11. int cmin_50 = INT_MAX; // chi phí di lại nhõ nhất giữa hai thành phố khác nhau
12. int best_50 = INT_MAX; // tổng chi phí nhõ nhất cần tìm, ban đầu đặt bằng giá trị vô cùng lớn INT_MAX = 2^31-1
13. int curr_50; // tổng chi phí tối thời điểm hiện tại
14. int mark_50[MAX]; // đánh dấu những thành phố đã di
15. int x_50[MAX]; // lưu giữ các thành phố đã di
16.
17. //Đọc dữ liệu vào
18. void input_50(){
19.     cin >> n_50;
20.     for (int i = 1; i <= n_50; ++i)
21.         for (int j = 1; j <= n_50; ++j){
22.             cin >> c_50[i][j];

```

Precheck

Check

	Input	Expected	Got	
✓	4 0 2 1 3 4 0 1 2 2 1 0 3 3 4 2 0	7	7	✓
✓	6 0 2 1 3 7 3 4 0 1 2 8 5 2 1 0 3 6 9 3 4 2 0 2 3 1 7 3 9 0 4 2 1 4 5 6 0	11	11	✓
✓	8 0 2 1 3 7 3 2 7	14	14	✓

Hình 3. 3. Test case bài 3.

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

```

Bai3_4_Tuan36_Lab3.cpp > main()
1. /* Bai 3.4 Cho dãy a có n phần tử. Một dãy con của a là dãy thu được bằng cách xóa đi
2. mõi phần tử của a và giữ nguyên thứ tự các phần tử còn lại (có thể không xóa phần tử nào).
3. Hãy tìm dãy con tăng dài nhất của a
4. Hồ Minh Dũng - 20235050 - 750829 - Tuan 36*/
5. #include <bits/stdc++.h>
6. using namespace std;
7. int a_50[1000], n_50;
8. int mem_50[1000]; // mang ghi nhớ lời giải các bài toán con đã được giải
9.
10. void init(){
11.     memset(mem_50, -1, sizeof(mem_50));
12. }
13.
14. /// Quy hoạch động,
15. /// Ham lis(i) trả về độ dài dãy con tăng dài nhất kết thúc bởi a[i]
16. int lis(int i) {
17.     //*****#
18.     // YOUR CODE HERE #
19.     //*****#
20.     if (mem_50[i] != -1) return mem_50[i];
21.     int res = 1;
22.     for (int j = i-1; j >= 0; j--){
23.         if (a_50[j] < a_50[i]){
24.             if (a_50[j] < a_50[i]){
25.                 res = max(res, lis(j) + 1);
26.             }
27.             mem_50[i] = res;
28.             return res;
29.         }
30.     }/// Truy vết lời giải
31. void trace(int i){
32.     for(int j = 0; j < i; j++){
33.         if (a_50[j] < a_50[i] && mem_50[i] == 1 + mem_50[j]){
34.             trace(j);
35.             break;
36.         }
37.     }
38.     cout << a_50[i] << " ";
39. }
40.

```

Hình 4. 1. Code bài 4.

Hồ Minh Dũng - 20235050

```

C:\Bai3_4_Tuan36_Lab3.cpp > main()
16 int lis(int i){
17     if (mem_50[i] != -1) return mem_50[i];
18     int res = 1;
19     for (int j = i-1; j >= 0; j--){
20         if (a_50[j] < a_50[i]){
21             res = max(res, lis(j) + 1);
22         }
23     }
24     mem_50[i] = res;
25     return res;
26 }
27 // Truy vết lời giải
28 void trace(int i){
29     for(int j = i; j < i; j++){
30         if (a_50[j] < a_50[i] && mem_50[i] == 1 + mem_50[j]){
31             trace(j);
32             break;
33         }
34     }
35 }
36 cout << a_50[i] << " ";
37 }
38 }
39 }
40
41 int main(){
42     cout << "Ho Minh Dũng - 5050\n";
43     init();
44     cin >> n_50;
45     for(int i = 0; i < n_50; i++) cin >> a_50[i];
46     int res = 1, pos = 0;
47     for(int i = 1; i < n_50; i++){
48         if (res < lis(i)){
49             res = lis(i);
50             pos = i;
51         }
52     }
53     cout << res << endl;
54     trace(pos);
55     return 0;
56 }
57 /*Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
58

```

Hình 4. 2. Code bài 4.

Answer: (penalty regime: 10, 20, ... %)

Time left 23:36:16

```

1 /* Bai 3.4 Cho dãy a có n phần tử. Một dãy con của a là dãy thu được bằng cách xóa đi
2 một số phần tử của a và giữ nguyên thứ tự các phần tử còn lại (có thể không xóa phần tử nào).
3 Hãy tìm dãy con tăng dài nhất của a
4 Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
5 #include <bits/stdc++.h>
6 using namespace std;
7 int a_50[1000], n_50;
8 int mem_50[1000]; //# mảng ghi nhớ lời giải các bài toán con đã được giải
9
10 void init(){
11     memset(mem_50, -1, sizeof(mem_50));
12 }
13
14 //## Quy hoạch động,
15 //## Hàm lis(i) trả về độ dài dãy con tăng dài nhất kết thúc bởi a[i]
16 int lis(int i) {
17     //*****
18     # YOUR CODE HERE #
19     //*****
20     if (mem_50[i] != -1) return mem_50[i];
21     int res = 1;
22     for (int j = i-1; j >= 0; j--) {

```

Precheck Check

	Input	Expected	Got	
✓	6 2 1 5 4 3 6	3 2 5 6	3 2 5 6	✓
✓	10 2 1 5 4 3 6 3 -6 9 10	5 2 5 6 9 10	5 2 5 6 9 10	✓
✓	20 2 -10 -8 5 4 3 6 3 -6 9 10 12 2 5 25 9 15 30 -100 45	10 -10 -8 5 6 9 10 12 25 30 45	10 -10 -8 5 6 9 10 12 25 30 45	✓

Passed all tests! ✓

Hình 4. 3. Test case bài 4.

Bài tập 5: Tính tổ hợp

```

C:\Bai3_5_Tuan36_Lab3.cpp > main()
1 /* Bai 3.5 Tính tổ hợp
2 Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
3 #include <iostream>
4 using namespace std;
5
6 int binom(int n_50, int k_50) {
7     if (k_50 > n_50) return 0;
8     if (k_50 == 0) return 1;
9     return binom(n_50 - 1, k_50) + binom(n_50 - 1, k_50 - 1);
10 }
11
12 //dùng hash table để lưu lại kết quả
13 int c_50[100][100];
14 void table() {
15     for (int i_50 = 0; i_50 < 100; i_50++) {
16         c_50[i_50][0] = 1;
17         c_50[i_50][i_50] = 1;
18     }
19     for (int i_50 = 1; i_50 < 100; i_50++) {
20         for (int j_50 = 1; j_50 < 100; j_50++) {
21             c_50[i_50][j_50] = c_50[i_50 - 1][j_50] + c_50[i_50 - 1][j_50 - 1];
22         }
23     }
24 }
25 int binom2(int n_50, int k_50) {
26     // khử đệ quy
27     //*****
28     # YOUR CODE HERE #
29     //*****
30     return c_50[n_50][k_50];
31 }
32
33 int main() {
34     cout << "Ho Minh Dũng - 5050\n";
35     table();
36     int m_50;
37     cin >> m_50;
38     for (int n_50 = 1; n_50 <= m_50; ++n_50) {
39         for (int k_50 = 0; k_50 <= n_50; ++k_50)
40             printf("%d ", binom(n_50, k_50));
41             printf("\n");
42     }
43     for (int n_50 = 1; n_50 <= m_50; ++n_50) {
44         for (int k_50 = 0; k_50 <= n_50; ++k_50)
45             printf("%d ", binom2(n_50, k_50));
46             printf("\n");
47     }
48 }
49

```

Hình 5. 1. Code bài 5.

```

C:\Bai3_5_Tuan36_Lab3.cpp > main()
1 void table() {
2     for (int i_50 = 0; i_50 < 100; i_50++) {
3         c_50[i_50][0] = 1;
4         c_50[i_50][i_50] = 1;
5     }
6     for (int i_50 = 1; i_50 < 100; i_50++) {
7         for (int j_50 = 1; j_50 < 100; j_50++) {
8             c_50[i_50][j_50] = c_50[i_50 - 1][j_50] + c_50[i_50 - 1][j_50 - 1];
9         }
10    }
11 }
12 int binom2(int n_50, int k_50) {
13     // khử đệ quy
14     //*****
15     # YOUR CODE HERE #
16     //*****
17     return c_50[n_50][k_50];
18 }
19
20 int main() {
21     cout << "Ho Minh Dũng - 5050\n";
22     table();
23     int m_50;
24     cin >> m_50;
25     for (int n_50 = 1; n_50 <= m_50; ++n_50) {
26         for (int k_50 = 0; k_50 <= n_50; ++k_50)
27             printf("%d ", binom(n_50, k_50));
28             printf("\n");
29     }
30     for (int n_50 = 1; n_50 <= m_50; ++n_50) {
31         for (int k_50 = 0; k_50 <= n_50; ++k_50)
32             printf("%d ", binom2(n_50, k_50));
33             printf("\n");
34     }
35 }
36

```

Hình 5. 2. Code bài 5.

The screenshot shows a programming contest interface. At the top right, a red box indicates "Time left 23:29:21". The main area contains C++ code for calculating binomial coefficients. The code includes a function `binom` and a `table` function that initializes a 2D array `c_50` with dimensions [100][100]. The `table` function uses nested loops to calculate values based on the formula $c_{50}[i_{50}][j_{50}] = c_{50}[i_{50}-1][j_{50}] + c_{50}[i_{50}-1][j_{50}-1]$. Below the code are two buttons: "Precheck" and "Check".

```

1 /* Bai 3.5 Tính tổ hợp
2 Ho Minh Dũng - 20235050 - 750829 - Tuần 36*/
3 #include <iostream>
4 using namespace std;
5
6 int binom(int n_50, int k_50) {
7     if (k_50 > n_50) return 0;
8     if (k_50 == 0) return 1;
9     return binom(n_50 - 1, k_50) + binom(n_50 - 1, k_50 - 1);
10}
11
12 //dùng hash_table để lưu lại kết quả
13 int c_50[100][100];
14 void table() {
15     for (int i_50 = 0; i_50 < 100; i_50++) {
16         c_50[i_50][0] = 1;
17         c_50[0][i_50] = 1;
18     }
19     for (int i_50 = 1; i_50 < 100; i_50++) {
20         for (int j_50 = 1; j_50 < 100; j_50++) {
21             c_50[i_50][j_50] = c_50[i_50 - 1][j_50] + c_50[i_50 - 1][j_50 - 1];
22         }
23     }
24 }

```

	Input	Expected	Got
✓	4	1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1	1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1
✓	10	1 1	1 1

Hình 5. 3. Test case bài 5

		Time left 23:29:10
✓	10	<pre> 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1 1 7 21 35 35 21 7 1 1 8 28 56 70 56 28 8 1 1 9 36 84 126 126 84 36 9 1 1 10 45 120 210 252 210 120 45 10 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1 1 7 21 35 35 21 7 1 1 8 28 56 70 56 28 8 1 1 9 36 84 126 126 84 36 9 1 1 10 45 120 210 252 210 120 45 10 1 </pre>
✓	20	<pre> 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1 1 7 21 35 35 21 7 1 1 8 28 56 70 56 28 8 1 1 9 36 84 126 126 84 36 9 1 1 10 45 120 210 252 210 120 45 10 1 1 11 55 165 330 462 462 330 165 55 11 1 1 12 66 220 495 792 924 792 495 220 66 12 1 1 13 78 286 715 1287 1716 1716 1287 715 286 78 13 1 1 14 91 364 1800 2002 3003 3432 3003 2002 1001 364 91 14 1 1 15 105 455 1365 3003 5005 6435 6435 5005 3003 1365 455 105 15 1 1 16 120 560 1820 4368 8008 11440 12870 11440 8008 4368 1820 560 120 16 1 </pre>

Hình 5. 4. Test case bài 5

Bài tập 6: Tìm ước chung lớn nhất

```

Bai3_6_Tuan36_Lab3.cpp > main()
1 /* Bai 3.6 Tính ước chung lớn nhất của hai số cho trước
2 Hồ Minh Dũng - 20235050 - 750829 - Yuan 36*/
3 #include <iostream>
4 using namespace std;
5
6 int gcd(int a_50, int b_50){
7     if (b_50 == 0) return a_50;
8     return gcd(b_50, a_50 % b_50);
9 }
10
11 int gcd2(int a_50, int b_50){
12
13     //# khử đệ quy
14     /*****
15     # YOUR CODE HERE #
16     *****/
17     while (a_50 != 0){
18         if (a_50 > b_50){
19             a_50 %= b_50;
20         }
21         else{
22             int tmp_50 = b_50 % a_50;
23             b_50 = a_50;
24             a_50 = tmp_50;
25         }
26     }
27     return b_50;
28 }
29
30 int main() {
31     cout << "Ho Minh Dũng - 5050\n";
32     int a_50, b_50;
33     cin >> a_50 >> b_50;
34     cout << gcd(a_50, b_50) << endl << gcd2(a_50, b_50);
35     return 0;
36 }
37

```

Hình 6. 1. Code bài 6.

Answer: (penalty regime: 10, 20, ... %)

```
1 /* Bai 3.6 Tính ước chung lớn nhất của hai số cho trước
2 Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
3 #include <iostream>
4 using namespace std;
5
6 int gcd(int a_50, int b_50){
7     if (b_50 == 0) return a_50;
8     return gcd(b_50, a_50 % b_50);
9 }
10
11 int gcd2(int a_50, int b_50){
12
13     //# khử đệ quy
14     //*****#
15     # YOUR CODE HERE #
16     //*****#
17     while (a_50 != 0){
18         if (a_50 > b_50){
19             a_50 %= b_50;
20         }
21         else{
22             int tmp_50 = b_50 % a_50;
```

Precheck

Check

	Input	Expected	Got	
✓	50 35	5 5	5 5	✓
✓	217 413	7 7	7 7	✓

Passed all tests! ✓

Hình 6. 2. Test case bài 6.

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

Hồ Minh Dũng - 20235050

```

Bai3_7_Tuan36_Lab3.cpp > main()
1  /* Bai 3.6 Liet ke xau nhanh
2  Suy dung phuong phap khur de quy bang stack,
3  hay liet ke cac xau khi phan do dai n khong co k bit 1 nao lien tiep
4  Ho Minh Dung - 20235050 - 750829 - Tuan 36*/
5 #include <iostream>
6 using namespace std;
7
8 struct state{
9     int i, j, old_l;
10    // constructor
11    state(int _i = 0, int _j = 0, int _l = 0):
12        i(_i), j(_j), old_l(_l){}
13 };
14
15 int main() {
16     cout << "Ho Minh Dung - 5050\n";
17     int n_50, k_50;
18     cin >> n_50 >> k_50;
19     int x_50[n_50+1];
20     stack<state> s_50;
21     // number of consecutive suffix 1
22     int l_50 = 0;
23     s_50.push(state(1, 0));
24     while (!s_50.empty()){
25         state &top_50 = s_50.top();
26         // if a new binary sequence is found
27         if (top_50.i > n_50){
28             for (int i_50 = 1; i_50 <= n_50; ++i_50)
29                 cout << x_50[i_50] << " \n"[i_50 == n_50];
30             s_50.pop();
31             continue;
32         }
33
34         // khur de quy
35         //*****
36         # YOUR CODE HERE #
37         *****/
38
39         // Tra lai trang thai l neu vua backtrack tu nhanh 1
40         if (top_50.j == 1) {
41             l_50 = top_50.old_l;
42         }
43
44         // Neu da thu cua 0 va 1 => quay lui
45         if (top_50.j > 1) {
46             s_50.pop();
47             continue;
48         }
49
50         // Kiem tra xem co the dat 0 hoac 1 tai vi tri i khong
51         if (l_50 + 1 < k_50 || top_50.j == 0) {
52             x_50[top_50.i] = top_50.j;
53             top_50.old_l = l_50;
54
55             if (top_50.j == 1) l_50++;
56             else l_50 = 0;
57
58             s_50.push(state(top_50.i + 1, 0));
59         }
60
61         ++top_50.j; // chuyen sang thu gi tri tiep theo (từ 0 -> 1)
62     }
63     return 0;
64 }
/*Ho Minh Dung - 20235050 - 750829 - Tuan 36*/

```

Hình 7. 1. Code bài 7.

```

Bai3_7_Tuan36_Lab3.cpp > main()
1  /* Bai 3.6 Liet ke xau nhanh
2  Suy dung phuong phap khur de quy bang stack,
3  hay liet ke cac xau khi phan do dai n khong co k bit 1 nao lien tiep
4  Ho Minh Dung - 20235050 - 750829 - Tuan 36*/
5 #include <iostream>
6 using namespace std;
7
8 struct state{
9     int i, j, old_l;
10    // constructor
11    state(int _i = 0, int _j = 0, int _l = 0):
12        i(_i), j(_j), old_l(_l){}
13 };
14
15 int main() {
16     cout << "Ho Minh Dung - 5050\n";
17     int n_50, k_50;
18     cin >> n_50 >> k_50;
19     int x_50[n_50+1];
20     stack<state> s_50;
21     // number of consecutive suffix 1
22     int l_50 = 0;
23     s_50.push(state(1, 0));
24     while (!s_50.empty()){
25         state &top_50 = s_50.top();
26         // if a new binary sequence is found
27         if (top_50.i > n_50){
28             for (int i_50 = 1; i_50 <= n_50; ++i_50)
29                 cout << x_50[i_50] << " \n"[i_50 == n_50];
30             s_50.pop();
31             continue;
32         }
33
34         // khur de quy
35         //*****
36         # YOUR CODE HERE #
37         *****/
38
39         // Tra lai trang thai l neu vua backtrack tu nhanh 1
40         if (top_50.j == 1) {
41             l_50 = top_50.old_l;
42         }
43
44         // Neu da thu cua 0 va 1 => quay lui
45         if (top_50.j > 1) {
46             s_50.pop();
47             continue;
48         }
49
50         // Kiem tra xem co the dat 0 hoac 1 tai vi tri i khong
51         if (l_50 + 1 < k_50 || top_50.j == 0) {
52             x_50[top_50.i] = top_50.j;
53             top_50.old_l = l_50;
54
55             if (top_50.j == 1) l_50++;
56             else l_50 = 0;
57
58             s_50.push(state(top_50.i + 1, 0));
59         }
60
61         ++top_50.j; // chuyen sang thu gi tri tiep theo (từ 0 -> 1)
62     }
63     return 0;
64 }
/*Ho Minh Dung - 20235050 - 750829 - Tuan 36*/

```

Hình 7. 2. Code bài 7.

Answer: (penalty regime: 10, 20, ... %)

Time left 23:01:08

```
1 /* Bai 3.6 Liệt kê xâu nhị phân
2 Sử dụng phương pháp khử đệ quy bằng stack,
3 hãy liệt kê các xâu nhị phân độ dài n không có k bit 1 nào liên tiếp
4 Hồ Minh Dũng - 20235050 - 750829 - Tuan 36*/
5 #include <bits/stdc++.h>
6 using namespace std;
7
8 struct state{
9     int i, j, old_L;
10    //## constructor
11    state(int _i = 0, int _j = 0, int _L = 0):
12        i(_i), j(_j), old_L(_L){}
13};
14
15 int main() {
16     int n_50, k_50;
17     cin >> n_50 >> k_50;
18     int x_50[n_50+1];
19     stack<state> s_50;
20     //## number of consecutive suffix 1
21     int L_50 = 0;
22     s_50.push(state(1, 0));
```

Precheck Check

	Input	Expected	Got		
✓	4 2	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	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	✓	

Hình 7. 3. Test case bài 7.

		1 0 0 0 1 0 0 1 1 0 1 0	1 0 0 0 1 0 0 1 1 0 1 0		
✓	5 3	0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 0 1 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 0 0 1 1 0 0 1 1 1 0 1 0 1 1 0 1 1	0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 0 1 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 1 0 1 0 0 1 0 1 0 1 1 0 1 1 0 1 1 0 0 0 1 1 0 0 1 1 1 0 1 0 1 1 0 1 1	✓	Time left 23:00:42
✓	7 3	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 1 1	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 1 1	✓	

Hình 7. 4. Test case bài 7.

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.

Hồ Minh Dũng - 20235050

```

Bai3_8_Tuan36_Lab3.cpp > main()
1  /* Bai 3.8 Cán dài
2  Bạn đang muốn kiểm tra xem một vật cho trước có dùng nặng M
3  như người ta nói hay không. Có một cân thăng bằng và n quả cần.
4  Quả thứ i nặng mi. Hãy chỉ ra một cách cân thỏa mãn.
5  Quy cách in ra đã được tích hợp trong mã nguồn dưới.
6  Hồ Minh Dũng - 20235050 - 750829 - Tuan 36*/
7 #include <iostream>
8 using namespace std;
9
10 struct state{
11     int i, j, old_S;
12     state(int _i = 0, int _j = 0): i(_i), j(_j) {}
13 };
14
15 int main() {
16     cout << "Ho Minh Dũng - 5050\n";
17     int n_50, M_50;
18     cin >> n_50 >> M_50;
19     int m_50[n_50+1];
20     for (int i_50 = 1; i_50 <= n_50; ++i_50) cin >> m_50[i_50];
21     int x_50[n_50+1];
22     stack<state> s_50;
23     //# sum of selected weights
24     int sum_50 = 0;
25     s_50.push(state(1, -1));
26     while (!s_50.empty()){
27         state &top_50 = s_50.top();
28         if (top_50.i > n_50){
29             if (sum_50 == M_50){
30                 cout << "output = ";
31                 for (int i_50 = 1; i_50 <= n_50; ++i_50){
32                     if (x_50[i_50] == -1) cout << '-' << m_50[i_50];
33                     if (x_50[i_50] == 1) cout << '+' << m_50[i_50];
34                 }
35                 cout << "\n" << M_50;
36                 exit(0);
37             }
38             s_50.pop();
39             continue;
40         }
41     }
42 }

```

Hình 8. 1. Test case bài 8.

```

Bai3_8_Tuan36_Lab3.cpp > main()
15  int main() {
16      while (!s_50.empty()){
17          if (top_50.i > n_50){
18              if (sum_50 == M_50){
19                  for (int i_50 = 1; i_50 <= n_50; ++i_50){
20                      cout << "=" << M_50;
21                      exit(0);
22                  }
23                  s_50.pop();
24                  continue;
25              }
26          }
27          /*Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
28          // Khứ để quy
29          //*****
30          # YOUR CODE HERE #
31          //*****
32          //backtrack
33          if (top_50.j > -1){
34              sum_50 = top_50.old_S;
35          }
36          // -1: cân trái, 1: cân phải, 0: ko dùng
37          if (top_50.j > 1) {
38              x_50[top_50.i] = 0;
39              s_50.pop();
40              continue;
41          }
42          if (sum_50 != M_50 && top_50.i <= n_50) {
43              x_50[top_50.i] = top_50.j;
44              top_50.old_S = sum_50;
45              if (top_50.j == 1) sum_50 += m_50[top_50.i];
46              if (top_50.j == -1) sum_50 -= m_50[top_50.i];
47              s_50.push(state(top_50.i + 1, -1));
48          }
49          ++top_50.j;
50      }
51      cout << -1;
52
53      return 0;
54  }

```

Hình 8. 2. Test case bài 8.

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 đa làm việc liên tiếp đều có số ngày thuộc đoạn [K1,K2][1,2]. Hãy liệt kê tất cả các cách lập lịch tốt, với mỗi lịch in ra trên một dòng một xâu nhị phân độ dài n với bit 0/1 tương ứng là nghỉ/làm việc. Các xâu phải được in ra theo thứ tự từ điển

Hồ Minh Dũng - 20235050

```

Ba13_9_Tuan36.Lab3.cpp > main()
1  /* Bai 3.8 Lập lịch cho y tá
2  Một y tá cần lập lịch làm việc trong N ngày, mỗi ngày chỉ có thể là làm việc hay nghỉ ngơi.
3  Một lịch làm việc là tốt nếu không có hai ngày nghỉ nào liên tiếp và mọi chuỗi ngày tối đa
4  làm việc liên tiếp đều có số ngày thuộc đoạn [k1,k2]. Hãy liệt kê tất cả các cách lập lịch tốt,
5  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.
6  Các xâu phải được in ra theo thứ tự từ điển
7  Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
8 #include <bits/stdc++.h>
9 using namespace std;
10
11 int n_50, k1_50, k2_50;
12 int len1_50;
13 vector<string> ans_50;
14 string current_50 = "";
15
16 void Try(int k_50){
17     for (int i_50 = 0; i_50 <= 1; i_50++){
18         //tmp và s lưu lại trạng thái (backtrack đỡ phức tạp)
19         int tmp_50 = len1_50;
20         string s_50 = current_50;
21
22         if (i_50 == 0){
23             //chưa làm việc dù thi ko dc nghỉ
24             if (len1_50 < k1_50 && k_50 != 0) continue;
25             current_50 += to_string(i_50);
26             len1_50 = 0;
27         }
28         else{
29             //còn có thể làm việc
30             if (len1_50 < k2_50){
31                 current_50 += to_string(i_50);
32                 len1_50++;
33             }
34         }
35         if (k_50 == n_50 - 1){
36             //check trường hợp những ngày cuối làm việc không đủ trong khoảng [k1, k2]
37             if (current_50[k_50] == '1' && len1_50 >= k1_50 && len1_50 <= k2_50 || current_50[k_50] == '0')
38                 ans_50.push_back(current_50);
39             }
40         else Try(k_50 + 1);
41         len1_50 = tmp_50;
42         current_50 = s_50;
43     }
44 }

```

Hình 9. 1. Test case 1 và 2 bài 9.

```

Ba13_9_Tuan36.Lab3.cpp > main()
14 string current_50 = "";
15
16 void Try(int k_50){
17     for (int i_50 = 0; i_50 <= 1; i_50++){
18         //tmp và s lưu lại trạng thái (backtrack đỡ phức tạp)
19         int tmp_50 = len1_50;
20         string s_50 = current_50;
21
22         if (i_50 == 0){
23             //chưa làm việc dù thi ko dc nghỉ
24             if (len1_50 < k1_50 && k_50 != 0) continue;
25             current_50 += to_string(i_50);
26             len1_50 = 0;
27         }
28         else{
29             //còn có thể làm việc
30             if (len1_50 < k2_50){
31                 current_50 += to_string(i_50);
32                 len1_50++;
33             }
34         }
35         if (k_50 == n_50 - 1){
36             //check trường hợp những ngày cuối làm việc không đủ trong khoảng [k1, k2]
37             if (current_50[k_50] == '1' && len1_50 >= k1_50 && len1_50 <= k2_50 || current_50[k_50] == '0')
38                 ans_50.push_back(current_50);
39             }
40         else Try(k_50 + 1);
41         len1_50 = tmp_50;
42         current_50 = s_50;
43     }
44 }
45
46 int main(){
47     cout << "Ho Minh Dũng - 5050\n";
48     cin >> n_50 >> k1_50 >> k2_50;
49     Try(0);
50
51     //sắp xếp
52     sort(ans_50.begin(), ans_50.end());
53     cout << "output = \n";
54     for (string x_50 : ans_50) cout << x_50 << endl;
55 }

```

Hình 9. 2. Test case 3 bài 9.

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.

```

Bai3_10_Tuan36.cpp > res_50
1 /* Bai 3.10 Khoảng cách Hamming
2 Khoảng cách Hamming giữa hai xâu cùng độ dài là số vị trí mà
3 ký tự tại vị trí đó là khác nhau trên hai xâu.
4 Cho S là xâu gồm n ký tự 0. Hãy liệt kê tất cả các xâu nhị phân độ dài n,
5 có khoảng cách Hamming với S bằng H.
6 Các xâu phải được liệt kê theo thứ tự từ điển.
7 Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
8 #include <bits/stdc++.h>
9 using namespace std;
10 int n_50, h_50;
11 int cnt_50; //đếm số bit 1 hiện có
12 vector<string> res_50;
13 string current_50;
14
15 void Try(int k_50){
16     for (int i_50 = 0; i_50 <= 1; i_50++){
17         //tmp để lưu trạng thái ban đầu
18         char tmp_50 = current_50[k_50];
19         current_50[k_50] = (i_50 == 0) ? '0' : '1';
20         if (i_50 == 1) cnt_50++;
21
22         if (k_50 == n_50 - 1) {
23             //nếu số bit 1 bằng h thì ghi nhận kết quả
24             if (cnt_50 == h_50) res_50.push_back(current_50);
25         } else if (cnt_50 <= h_50) {
26             Try(k_50 + 1);
27         }
28         //trả về trạng thái ban đầu
29         current_50[k_50] = tmp_50;
30         if (i_50 == 1) cnt_50--;
31     }
32 }
33
34 int main(){
35     cout << "Ho Minh Dũng - 5050\noutput = \n";
36     ifstream inFile_50("input.txt"); //input từ file
37     int t_50; inFile_50 >> t_50;
38     while (t_50--){
39         inFile_50 >> n_50 >> h_50;
40         res_50.clear();
41         current_50 = string(n_50, '0');
42         Try(0);
43         for (string x_50 : res_50) cout << x_50 << endl;
44         cout << endl;
45     }
46 }

```

Input.txt	Ky_thuat_lap_trinh\Hamming\Lab02_Tuan36\Bai3_10_Tuan36.cpp
1 7	\$1 { g++ Bai3_10_Tuan36_Lab3.cpp -o Bai3_10_Tuan36_Lab3 ; if (\$?) { .\Bai3_10_Tuan36_Lab3 }
2 4 2	Ho Minh Dũng - 5050
3 1 0	output =
4 1 1	0011
5 2 0	0101
6 2 1	0110
7 5 2	1001
8 5 3	1010
9	1100
	0
	1
	00
	01
	10
	00011
	00101
	00110
	01001
	01010
	01100
	10001
	10010
	10100
	11000
	00111
	01011
	01101
	01110
	10011
	10101
	10110
	11001
	11010
	11100

Hình 10. 1. Test case bài 10.

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.

Hồ Minh Dũng - 20235050

```
Bai3_11_Tuan36_Lab3.cpp > ...
1 /* Bai 3.11 Superior là một hòn đảo tuyệt đẹp với n địa điểm chụp ảnh
2 và các đường mòn chiều nối các điểm chụp ảnh với nhau.
3 Đoàn khách tham quan có r người với sở thích chụp ảnh khác nhau.
4 Theo đó, mỗi người sẽ đưa ra danh sách các địa điểm mà họ muốn chụp.
5 Bạn cần giúp mỗi người trong danh sách di chuyển sao cho
6 đi qua các điểm họ yêu cầu đúng một lần, không đi qua điểm nào khác,
7 bắt đầu tại điểm đầu tiên và kết thúc tại điểm cuối cùng trong danh sách
8 mà họ đưa ra, và có tổng khoảng cách di lại là nhỏ nhất.
9 Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
10 #include <bits/stdc++.h>
11 using namespace std;
12
13 int n_50, r_50;
14 int c_50[1000][1000]; //ma trận chỉ phí
15 vector<int> want_50; //lưu danh sách các điểm muốn thăm
16 bool visited_50[1000]; //điểm đã thăm
17 int min_cost_50;
18 int current_50 = 0;
19 int pre_50; //điểm thăm trước đó
20 int cmin_50; //chi phí nhỏ nhất
21
22 void Try(int k_50){
23     for (int i_50 = 1; i_50 < want_50.size() - 1; i_50++){
24         // chưa thăm và có đường đi đến
25         if (!visited_50[want_50[i_50]] && c_50[pre_50][want_50[i_50]] != 0){
26             //lưu trạng thái ban đầu
27             int tmp_50 = pre_50;
28             visited_50[want_50[i_50]] = true;
29             current_50 += c_50[pre_50][want_50[i_50]];
30             //cập nhật pre
31             pre_50 = want_50[i_50];
32             // không tính điểm cuối
33             if (k_50 == want_50.size() - 2){
34                 // có đường từ điểm want[i] đến điểm cuối
35                 if (c_50[want_50[i_50]][want_50.back()] != 0){
36                     min_cost_50 = min(min_cost_50, current_50 + c_50[want_50[i_50]][want_50.back()]);
37                 }
38             } else {
39                 //branch and bound để tối ưu
40                 int g_50 = current_50 + cmin_50 * (n_50 - k_50 + 1);
41                 if (g_50 < min_cost_50) Try(k_50 + 1);
42             }
43         }
44         //trả lại trạng thái ban đầu
45         current_50 -= c_50[tmp_50][want_50[i_50]];
46         visited_50[want_50[i_50]] = false;
47         pre_50 = tmp_50;
48     }
49 }
50
51
52 int main(){
53     cout << "Ho Minh Dũng - 5050\noutput = \n";
54     //input
55     ifstream inFile_50("input.txt");
56     inFile_50 >> n_50 >> r_50;
57     cmin_50 = INT_MAX;
58     for (int i_50 = 0; i_50 < n_50; i_50++){
59         for (int j_50 = 0; j_50 < n_50; j_50++){
60             inFile_50 >> c_50[i_50][j_50];
61             if (c_50[i_50][j_50] > 0) cmin_50 = min(cmin_50, c_50[i_50][j_50]);
62         }
63     }
64
65     inFile_50.ignore();
66     for (int i_50 = 0; i_50 < r_50; i_50++){
67         memset(visited_50, false, sizeof(visited_50));
68         want_50.clear();
69     }
70 }
```

Ky_thuat_lap_trinh\Bai3_11_Tuan36_Lab3.cpp > ...
n36> cd "G:\HUST\Ky_thuat_lap_trinh\Bai3_11_Tuan36_Lab3" ; if (\$?) {
g++ Bai3_11_Tuan36_Lab3.cpp -o Bai3_11_Tuan36_Lab3 ; if (\$?) { ./Bai3_11_Tuan36_Lab3 }
Ho Minh Dũng - 5050
output =
5
0
7
PS G:\HUST\Ky_thuat_lap_trinh\Bai3_11_Tuan36_Lab3

Hình 11. 1. Test case I bài 11.

```
Bai3_11_Tuan36_Lab3.cpp > ...
1 void Try(int k_50){
2     for (int i_50 = 1; i_50 < want_50.size() - 1; i_50++){
3         // chưa thăm và có đường đi đến
4         if (!visited_50[want_50[i_50]] && c_50[pre_50][want_50[i_50]] != 0){
5             //lưu trạng thái ban đầu
6             int tmp_50 = pre_50;
7             visited_50[want_50[i_50]] = true;
8             current_50 += c_50[pre_50][want_50[i_50]];
9             //cập nhật pre
10            pre_50 = want_50[i_50];
11            // không tính điểm cuối
12            if (k_50 == want_50.size() - 2){
13                // có đường từ điểm want[i] đến điểm cuối
14                if (c_50[want_50[i_50]][want_50.back()] != 0){
15                    min_cost_50 = min(min_cost_50, current_50 + c_50[want_50[i_50]][want_50.back()]);
16                }
17            } else {
18                //branch and bound để tối ưu
19                int g_50 = current_50 + cmin_50 * (n_50 - k_50 + 1);
20                if (g_50 < min_cost_50) Try(k_50 + 1);
21            }
22        }
23        //trả lại trạng thái ban đầu
24        current_50 -= c_50[tmp_50][want_50[i_50]];
25        visited_50[want_50[i_50]] = false;
26        pre_50 = tmp_50;
27    }
28 }
29
30
31 int main(){
32     cout << "Ho Minh Dũng - 5050\noutput = \n";
33     //input
34     ifstream inFile_50("input.txt");
35     inFile_50 >> n_50 >> r_50;
36     cmin_50 = INT_MAX;
37     for (int i_50 = 0; i_50 < n_50; i_50++){
38         for (int j_50 = 0; j_50 < n_50; j_50++){
39             inFile_50 >> c_50[i_50][j_50];
40             if (c_50[i_50][j_50] > 0) cmin_50 = min(cmin_50, c_50[i_50][j_50]);
41         }
42     }
43
44     inFile_50.ignore();
45     for (int i_50 = 0; i_50 < r_50; i_50++){
46         memset(visited_50, false, sizeof(visited_50));
47         want_50.clear();
48     }
49 }
```

Ky_thuat_lap_trinh\Bai3_11_Tuan36_Lab3.cpp > ...
n36> cd "G:\HUST\Ky_thuat_lap_trinh\Bai3_11_Tuan36_Lab3" ; if (\$?) {
g++ Bai3_11_Tuan36_Lab3.cpp -o Bai3_11_Tuan36_Lab3 ; if (\$?) { ./Bai3_11_Tuan36_Lab3 }
Ho Minh Dũng - 5050
output =
5
0
7
PS G:\HUST\Ky_thuat_lap_trinh\Bai3_11_Tuan36_Lab3

Hình 11. 2. Test case I bài 11.

Hồ Minh Dũng - 20235050

```

Bài3_11_Tuan36_Lab3.cpp > ...
22 void Try(int k_50){
23     for (int i_50 = 1; i_50 < want_50.size() - 1; i_50++){
25         if (!visited_50[want_50[i_50]] && c_50[pre_50][want_50[i_50]] != 0){
49     }
50 }
51
52 int main(){
53     cout << "Ho Minh Dũng - 5050\noutput = \n";
54     //input
55     ifstream inFile_50("input.txt");
56     inFile_50 >> n_50 >> r_50;
57     cmin_50 = INT_MAX;
58     for (int i_50 = 0; i_50 < n_50; i_50++){
59         for (int j_50 = 0; j_50 < n_50; j_50++){
60             inFile_50 >> c_50[i_50][j_50];
61             if (c_50[i_50][j_50] > 0) cmin_50 = min(cmin_50, c_50[i_50][j_50]);
62         }
63     }
64
65     inFile_50.ignore();
66     for (int i_50 = 0; i_50 < r_50; i_50++){
67         memset(visited_50, false, sizeof(visited_50));
68         want_50.clear();
69         string s_50;
70         //xử lý đầu vào của danh sách điểm đến
71         getline(inFile_50, s_50);
72         stringstream ss_50(s_50);
73         string tmp_50;
74         while (ss_50 >> tmp_50){
75             want_50.push_back(stoi(tmp_50) - 1);
76         }
77
78         visited_50[want_50[0]] = true; // điểm bắt đầu
79         pre_50 = want_50[0];
80         min_cost_50 = INT_MAX;
81         current_50 = 0;
82         Try(1);
83         //nếu min_cost ko dc cập nhật thì cập nhật giá trị = 0
84         if (min_cost_50 == INT_MAX) min_cost_50 = 0;
85         cout << min_cost_50 << endl;
86     }
87 }

```

Hình 11. 3. Test case 1 bài 11.

```

Bài3_11_Tuan36_Lab3.cpp > main()
22 void Try(int k_50){
23     for (int i_50 = 1; i_50 < want_50.size() - 1; i_50++){
25         if (!visited_50[want_50[i_50]] && c_50[pre_50][want_50[i_50]] != 0){
49     }
50 }
51
52 int main(){
53     cout << "Ho Minh Dũng - 5050\noutput = \n";
54     //input
55     ifstream inFile_50("input.txt");
56     inFile_50 >> n_50 >> r_50;
57     cmin_50 = INT_MAX;
58     for (int i_50 = 0; i_50 < n_50; i_50++){
59         for (int j_50 = 0; j_50 < n_50; j_50++){
60             inFile_50 >> c_50[i_50][j_50];
61             if (c_50[i_50][j_50] > 0) cmin_50 = min(cmin_50, c_50[i_50][j_50]);
62         }
63     }
64
65     inFile_50.ignore();
66     for (int i_50 = 0; i_50 < r_50; i_50++){
67         memset(visited_50, false, sizeof(visited_50));
68         want_50.clear();
69         string s_50;
70         //xử lý đầu vào của danh sách điểm đến
71         getline(inFile_50, s_50);
72         stringstream ss_50(s_50);
73         string tmp_50;
74         while (ss_50 >> tmp_50){
75             want_50.push_back(stoi(tmp_50) - 1);
76         }
77
78         visited_50[want_50[0]] = true; // điểm bắt đầu
79         pre_50 = want_50[0];
80         min_cost_50 = INT_MAX;
81         current_50 = 0;
82         Try(1);
83         //nếu min_cost ko dc cập nhật thì cập nhật giá trị = 0
84         if (min_cost_50 == INT_MAX) min_cost_50 = 0;
85         cout << min_cost_50 << endl;
86     }
87 }

```

Hình 11. 4. Test case 2 bài 11.

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.

Hồ Minh Dũng - 20235050

```

Bai3_12.Tuan36.Lab3.cpp > Try(int, int)
1 /* Bai 3.12 Cho đồ thị vô hướng G,
2   hay đèn số dương đi dì qua k cạnh và không đi qua đỉnh nào quá một lần.
3   Hồ Minh Dũng - 20235050 - 750829 - Tuan 36*/
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 int n_50, k_50, m_50;
8 pair<int, int> edge_50[100];
9 vector<int> adj_50[100]; //danh sách kè
10 bool visited_50[100]; //các đỉnh đã đi qua
11 int res_50; //kết quả (chưa có kết quả trùng VD 1-2-3 và 3-2-1 là trùng)
12
13 //DFS
14 void Try(int a, int depth){
15     //đuynết danh sách kè
16     for (int e : adj_50[a]){
17         if (!visited_50[e]){
18             visited_50[e] = true;
19             //tăng độ sâu
20             depth++;
21             if (depth == k_50) res_50++;
22             else Try(e, depth);
23             visited_50[e] = false;
24             depth--;
25         }
26     }
27 }
28
29 int main(){
30     cout << "Ho Minh Dũng - 5050\noutput = \n";
31     //input
32     ifstream inFile("input.txt");
33     inFile >> n_50 >> k_50 >> m_50;
34     for (int i = 0; i < m_50; i++){
35         int x, y; inFile >> x >> y;
36         x--; y--;
37         edge_50[i] = pair(x, y);
38         adj_50[x].push_back(y);
39         adj_50[y].push_back(x);
40     }
41 }

```

Kỹ thuật lập trình|Đỗ Minh Hùng|M_Lab03.Tuan36> cd "G:\HUST\Kỹ thuật lập trình\Đỗ Minh Hùng\M_Lab03.Tuan36" ; if (\$?) { g++ Bai3_12.Tuan36.Lab3.cpp -o Bai3_12.Tuan36.Lab3 } ; if (\$?) { ./Bai3_12.Tuan36.Lab3 } Ho Minh Dũng - 5050 output = 468 PS G:\HUST\Kỹ thuật lập trình\Đỗ Minh Hùng\M_Lab03.Tuan36>

Hình 12. 1 Test case 1 bài 12.

```

Bai3_12.Tuan36.Lab3.cpp > Try(int, int)
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int n_50, k_50, m_50;
5 pair<int, int> edge_50[100];
6 vector<int> adj_50[100]; //danh sách kè
7 bool visited_50[100]; //các đỉnh đã đi qua
8 int res_50; //kết quả (chưa có kết quả trùng VD 1-2-3 và 3-2-1 là trùng)
9
10 //DFS
11 void Try(int a, int depth){
12     //đuynết danh sách kè
13     for (int e : adj_50[a]){
14         if (!visited_50[e]){
15             visited_50[e] = true;
16             //tăng độ sâu
17             depth++;
18             if (depth == k_50) res_50++;
19             else Try(e, depth);
20             visited_50[e] = false;
21             depth--;
22         }
23     }
24 }
25
26 int main(){
27     cout << "Ho Minh Dũng - 5050\noutput = \n";
28     //input
29     ifstream inFile("input.txt");
30     inFile >> n_50 >> k_50 >> m_50;
31     for (int i = 0; i < m_50; i++){
32         int x, y; inFile >> x >> y;
33         x--; y--;
34         edge_50[i] = pair(x, y);
35         adj_50[x].push_back(y);
36         adj_50[y].push_back(x);
37     }
38
39     for (int i = 0; i < n_50; i++){
40         visited_50[i] = true;
41         Try(i, 0);
42         visited_50[i] = false;
43     }
44     //logi bỏ các kết quả trùng (vì là đồ thị vô hướng)
45     cout << res_50/2 << endl;
46 }
47
48 }

```

Kỹ thuật lập trình|Đỗ Minh Hùng|M_Lab03.Tuan36> cd "G:\HUST\Kỹ thuật lập trình\Đỗ Minh Hùng\M_Lab03.Tuan36" ; if (\$?) { g++ Bai3_12.Tuan36.Lab3.cpp -o Bai3_12.Tuan36.Lab3 } ; if (\$?) { ./Bai3_12.Tuan36.Lab3 } Ho Minh Dũng - 5050 output = 16825 PS G:\HUST\Kỹ thuật lập trình\Đỗ Minh Hùng\M_Lab03.Tuan36>

Hình 12. 2.Test case 2 bài 12.

Hồ Minh Dũng - 20235050

```

1 // Bai 3.12 Chỗ đồ thị vô hướng G,
2 //này đếm số đường đi qua 1 cạnh và không đi qua đỉnh nào quá một lần.
3 //Ho Minh Dũng - 20235050 - 750829 - Tuan 36/*
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 int n_50, k_50, m_50;
8 pair<int, int> edge_50[100];
9 vector<int> adj_50[100]; //danh sách kè
10 bool visited_50[100]; //các đỉnh đã đi qua
11 int res_50; //kết quả (chứa cả kết quả trùng VD 1-2-3 và 3-2-1 là trùng)
12
13 //DFS
14 void Try(int a, int depth){
15     //điều kiện dừng
16     for (int &e : adj_50[a]){
17         if (!visited_50[e]){
18             visited_50[e] = true;
19             //tăng độ sâu
20             depth++;
21             if (depth == k_50) res_50++;
22             else Try(e, depth);
23             visited_50[e] = false;
24             depth--;
25         }
26     }
27 }
28
29 int main(){
30     cout << "Ho Minh Dũng - 5050\noutput = \n";
31     //input
32     ifstream inFile("input.txt");
33     inFile >> n_50 >> k_50 >> m_50;
34     for (int i = 0; i < m_50; i++){
35         int x, y; inFile >> x >> y;
36         x--; y--;
37         edge_50[i] = pair(x, y);
38         adj_50[x].push_back(y);
39         adj_50[y].push_back(x);
40     }
41
42     for (int i = 0; i < n_50; i++){
43         visited_50[i] = true;
44         Try(i, 0);
45         visited_50[i] = false;
46     }
47     //logi bô các kết quả trùng (vì là đồ thị vô hướng)
48     cout << res_50/2 << endl;
49 }

```

n36> cd "G:\HUST\Ky_thuat_lap_trinh\Bai3_12_Tuan36_Lab3"; if (\$?) {
g++ Bai3_12_Tuan36_Lab3.cpp -o Bai3_12_Tuan36_Lab3 } ; if (\$?) { ./Bai3_12_Tuan36_Lab3 }
Ho Minh Dũng - 5050
output =
16565
PS G:\HUST\Ky_thuat_lap_trinh\Bai3_12_Tuan36_Lab3>

Hình 12. 3. Test case 3 bài 12.

```

1 // Bai 3.12 Chỗ đồ thị vô hướng G,
2 //này đếm số đường đi qua 1 cạnh và không đi qua đỉnh nào quá một lần.
3 //Ho Minh Dũng - 20235050 - 750829 - Tuan 36/*
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 int n_50, k_50, m_50;
8 pair<int, int> edge_50[100];
9 vector<int> adj_50[100]; //danh sách kè
10 bool visited_50[100]; //các đỉnh đã đi qua
11 int res_50; //kết quả (chứa cả kết quả trùng VD 1-2-3 và 3-2-1 là trùng)
12
13 //DFS
14 void Try(int a, int depth){
15     //điều kiện dừng
16     for (int &e : adj_50[a]){
17         if (!visited_50[e]){
18             visited_50[e] = true;
19             //tăng độ sâu
20             depth++;
21             if (depth == k_50) res_50++;
22             else Try(e, depth);
23             visited_50[e] = false;
24             depth--;
25         }
26     }
27 }
28
29 int main(){
30     cout << "Ho Minh Dũng - 5050\noutput = \n";
31     //input
32     ifstream inFile("input.txt");
33     inFile >> n_50 >> k_50 >> m_50;
34     for (int i = 0; i < m_50; i++){
35         int x, y; inFile >> x >> y;
36         x--; y--;
37         edge_50[i] = pair(x, y);
38         adj_50[x].push_back(y);
39         adj_50[y].push_back(x);
40     }
41
42     for (int i = 0; i < n_50; i++){
43         visited_50[i] = true;
44         Try(i, 0);
45         visited_50[i] = false;
46     }
47     //logi bô các kết quả trùng (vì là đồ thị vô hướng)
48     cout << res_50/2 << endl;
49 }

```

n36> cd "G:\HUST\Ky_thuat_lap_trinh\Bai3_12_Tuan36_Lab3"; if (\$?) {
g++ Bai3_12_Tuan36_Lab3.cpp -o Bai3_12_Tuan36_Lab3 } ; if (\$?) { ./Bai3_12_Tuan36_Lab3 }
Ho Minh Dũng - 5050
output =
43781
PS G:\HUST\Ky_thuat_lap_trinh\Bai3_12_Tuan36_Lab3>

Hình 12. 4. Test case 4 bài 12.

Hồ Minh Dũng - 20235050

```

Bai3_12_Tuan36_Lab3.cpp > TryInt, int
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 int n_50, k_50, m_50;
8 pair<int, int> edge_50[100];
9 vector<int> adj_50[100]; //danh sách kề
10 bool visited_50[100]; //các đỉnh đã đi qua
11 int res_50; //kết quả (chứa cả kết quả trùng VD 1-2-3 và 3-2-1 là trùng)
12
13 //DFS
14 void Try(int a, int depth){
15     //đuýt danh sách kề
16     for (int e : adj_50[a]){
17         if (!visited_50[e]){
18             visited_50[e] = true;
19             //tăng độ sâu
20             depth++;
21             if (depth == k_50) res_50++;
22             else Try(e, depth);
23             visited_50[e] = false;
24             depth--;
25         }
26     }
27 }
28
29 int main(){
30     cout << "Ho Minh Dũng - 5050\noutput = \n";
31     //input
32     ifstream inFile("input.txt");
33     inFile >> n_50 >> k_50 >> m_50;
34     for (int i = 0; i < m_50; i++){
35         int x, y; inFile >> x >> y;
36         x--; y--;
37         edge_50[i] = pair(x, y);
38         adj_50[x].push_back(y);
39         adj_50[y].push_back(x);
40     }
41
42     for (int i = 0; i < n_50; i++){
43         visited_50[i] = true;
44         Try(i, 0);
45         visited_50[i] = false;
46     }
47     //loại bỏ các kết quả trùng (vì là đồ thị vô hướng)
48     cout << res_50/2 << endl;
49 }

```

```

Ky_thuat_lap_trinh\Bai3_12_Tuan36_Lab3>n36> cd "g:\HUST\Ky_thuat_lap_trinh\bai3_12_tuan36_lab3"; if ($?) { g++ Bai3_12_Tuan36_Lab3.cpp -o Bai3_12_Tuan36_Lab3 } ; if ($?) { .\Bai3_12_Tuan36_Lab3 }
Ho Minh Dũng - 5050
output =
479511
PS G:\HUST\Ky_thuat_lap_trinh\Bai3_12_Tuan36_Lab3>

```

Hình 12. 5 Test case 5 bài 12.

```

Bai3_12_Tuan36_Lab3.cpp > ...
1 /* Bai 3.12 Cho đồ thị vô hướng G,
2   hãy đếm số đường đi đi qua k cạnh và không đi qua đỉnh nào quá một lần.
3   Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 int n_50, k_50, m_50;
8 pair<int, int> edge_50[100];
9 vector<int> adj_50[100]; //danh sách kề
10 bool visited_50[100]; //các đỉnh đã đi qua
11 int res_50; //kết quả (chứa cả kết quả trùng VD 1-2-3 và 3-2-1 là trùng)
12
13 //DFS
14 void Try(int a, int depth){
15     //đuýt danh sách kề
16     for (int e : adj_50[a]){
17         if (!visited_50[e]){
18             visited_50[e] = true;
19             //tăng độ sâu
20             depth++;
21             if (depth == k_50) res_50++;
22             else Try(e, depth);
23             visited_50[e] = false;
24             depth--;
25         }
26     }
27 }
28
29 int main(){
30     cout << "Ho Minh Dũng - 5050\noutput = \n";
31     //input
32     ifstream inFile("input.txt");
33     inFile >> n_50 >> k_50 >> m_50;
34     for (int i = 0; i < m_50; i++){
35         int x, y; inFile >> x >> y;
36         x--; y--;
37         edge_50[i] = pair(x, y);
38         adj_50[x].push_back(y);
39         adj_50[y].push_back(x);
40     }
41
42     for (int i = 0; i < n_50; i++){
43         visited_50[i] = true;
44         Try(i, 0);
45         visited_50[i] = false;
46     }
47 }

```

```

Ky_thuat_lap_trinh\Bai3_12_Tuan36_Lab3>n36> cd "g:\HUST\Ky_thuat_lap_trinh\bai3_12_tuan36_lab3"; if ($?) { g++ Bai3_12_Tuan36_Lab3.cpp -o Bai3_12_Tuan36_Lab3 } ; if ($?) { .\Bai3_12_Tuan36_Lab3 }
Ho Minh Dũng - 5050
output =
892
PS G:\HUST\Ky_thuat_lap_trinh\Bai3_12_Tuan36_Lab3>

```

Hình 12. 6 Test case 6 bài 12.

Hồ Minh Dũng - 20235050

```

Bai3_12_Tuan36_Lab3.cpp > ...
1 /* Bai 3.12 Cho đồ thị vô hướng G,
2   hay demás đường đi dì qua k cạnh và không dì qua đỉnh nào quá một lần.
3   Hồ Minh Dũng - 20235050 - 750829 - Tuan 36*/
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 int n_50, k_50, m_50;
8 pair<int, int> edge_50[100];
9 vector<int> adj_50[100]; //danh sách kè
10 bool visited_50[100]; //các đỉnh đã dì qua
11 int res_50; //kết quả (chưa có kết quả trùng VD 1-2-3 và 3-2-1 là trùng)
12
13 //DFS
14 void Try(int a, int depth){
15     //đuynh danh sách kè
16     for (int &e : adj_50[a]){
17         if (!visited_50[e]){
18             visited_50[e] = true;
19             //tăng độ sâu
20             depth++;
21             if (depth == k_50) res_50++;
22             else Try(e, depth);
23             visited_50[e] = false;
24             depth--;
25         }
26     }
27 }
28
29 int main(){
30     cout << "Hồ Minh Dũng - 5050\noutput = \n";
31     //Input
32     ifstream inFile("input.txt");
33     inFile >> n_50 >> k_50 >> m_50;
34     for (int i = 0; i < m_50; i++){
35         int x, y; inFile >> x >> y;
36         x--; y--;
37         edge_50[i] = pair(x, y);
38         adj_50[x].push_back(y);
39         adj_50[y].push_back(x);
40     }
41
42     for (int i = 0; i < n_50; i++){
43         visited_50[i] = true;
44         Try(i, 0);
45         visited_50[i] = false;
46     }
47 }

```

Ky_thuat_lap_trinh\ĐungH\Lab03_Tua
n36> cd "G:\HUST\Ky_thuat_lap_trinh\ĐungH\Lab03_Tuan36"; if (\$?) {
g++ Bai3_12_Tuan36_Lab3.cpp -o Bai3_12_Tuan36_Lab3 }; if (\$?) { ./Bai3_12_Tuan36_Lab3 }
Hồ Minh Dũng - 5050
output =
1105855
PS G:\HUST\Ky_thuat_lap_trinh\ĐungH\Lab03_Tuan36>

Hình 12. 7. Test case 7 bài 12

```

Bai3_12_Tuan36_Lab3.cpp > g++ visited_50
1 /* Bai 3.12 Cho đồ thị vô hướng G,
2   hay đém số đường đi dì qua k cạnh và không dì qua đỉnh nào quá một lần.
3   Hồ Minh Dũng - 20235050 - 750829 - Tuan 36*/
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 int n_50, k_50, m_50;
8 pair<int, int> edge_50[100];
9 vector<int> adj_50[100]; //danh sách kè
10 bool visited_50[100]; //các đỉnh đã dì qua
11 int res_50; //kết quả (chưa có kết quả trùng VD 1-2-3 và 3-2-1 là trùng)
12
13 //DFS
14 void Try(int a, int depth){
15     //đuynh danh sách kè
16     for (int &e : adj_50[a]){
17         if (!visited_50[e]){
18             visited_50[e] = true;
19             //tăng độ sâu
20             depth++;
21             if (depth == k_50) res_50++;
22             else Try(e, depth);
23             visited_50[e] = false;
24             depth--;
25         }
26     }
27 }
28
29 int main(){
30     cout << "Hồ Minh Dũng - 5050\noutput = \n";
31     //Input
32     ifstream inFile("input.txt");
33     inFile >> n_50 >> k_50 >> m_50;
34     for (int i = 0; i < m_50; i++){
35         int x, y; inFile >> x >> y;
36         x--; y--;
37         edge_50[i] = pair(x, y);
38         adj_50[x].push_back(y);
39         adj_50[y].push_back(x);
40     }
41
42     for (int i = 0; i < n_50; i++){
43         visited_50[i] = true;
44         Try(i, 0);
45         visited_50[i] = false;
46     }
47 }

```

Ky_thuat_lap_trinh\ĐungH\Lab03_Tua
n36> cd "G:\HUST\Ky_thuat_lap_trinh\ĐungH\Lab03_Tuan36"; if (\$?) {
g++ Bai3_12_Tuan36_Lab3.cpp -o Bai3_12_Tuan36_Lab3 }; if (\$?) { ./Bai3_12_Tuan36_Lab3 }
Hồ Minh Dũng - 5050
output =
1865192
PS G:\HUST\Ky_thuat_lap_trinh\ĐungH\Lab03_Tuan36>

Hình 12. 8. Test case 8 bài 12.

Hồ Minh Dũng - 20235050

```
Bai3_12_Tuan36_Lab3.cpp > ...
1 /* Bai 3.12 Cho đồ thị vô hướng G,
2   hay đèn số dương đi dì qua k cạnh và không đi qua đỉnh nào quá một lần.
3   Ho Minh Dũng - 20235050 - 750829 - Tuan 36*/
4 #include <bits/stdc++.h>
5 using namespace std;
6
7 int n_50, k_50, m_50;
8 pair<int, int> edge_50[100];
9 vector<int> adj_50[100]; //danh sách kè
10 bool visited_50[100]; //các đỉnh đã đi qua
11 int res_50; //kết quả (chưa có kết quả trùng vD 1-2-3 và 3-2-1 là trùng)
12
13 //DFS
14 void Try(int a, int depth){
15     //đọc danh sách kè
16     for (int e : adj_50[a]){
17         if (!visited_50[e]){
18             visited_50[e] = true;
19             //tăng độ sâu
20             depth++;
21             if (depth == k_50) res_50++;
22             else Try(e, depth);
23             visited_50[e] = false;
24             depth--;
25         }
26     }
27 }
28
29 int main(){
30     cout << "Ho Minh Dũng - 5050\noutput = \n";
31     //Input
32     ifstream inFile("input.txt");
33     inFile >> n_50 >> k_50 >> m_50;
34     for (int i = 0; i < m_50; i++){
35         int x, y; inFile >> x >> y;
36         x--; y--;
37         edge_50[i] = pair(x, y);
38         adj_50[x].push_back(y);
39         adj_50[y].push_back(x);
40     }
41
42     for (int i = 0; i < n_50; i++){
43         visited_50[i] = true;
44         Try(i, 0);
45         visited_50[i] = false;
46     }
47     //Test bài xác định kết quả (vì có thể có nhiều)
0 0 0
```

input.txt	output
1 30 11	Kỹ thuật lập trình\Đỗng\Bai3_Tua...
2 60	n36> cd "g:\HUST\Kỹ thuật lập trình\Đỗng\Bai3_Tua...
3 13 29	\Đỗng\Bai3_Tua36"; if (\$?) {
4 1 15	g++ Bai3_12_Tuan36_Lab3.cpp -o Bai3_12_Tuan36_Lab3 ; if (\$?) { ./Bai3_12_Tuan36_Lab3 }
5 4 7	Ho Minh Dũng - 5050
6 17 20	output:
7 1 2	3431731
8 11 16	PS 6:\HUST\Kỹ thuật lập trình\Đỗng\Bai3_Tua36>
9 20 12	M_Lab3_Tua36>
10 29 27	
11 25 21	
12 24 23	
13 10 6	
14 9 3	
15 9 4	
16 1 25	
17 14 24	
18 21 11	
19 21 29	
20 27 14	
21 1 8	
22 30 1	
23 9 5	
24 1 28	
25 2 26	
26 18 25	
27 7 11	
28 2 29	
29 3 23	
30 1 13	
31 2 28	
32 15 14	
33 9 18	
34 27 19	
35 21 23	
36 15 29	
37 9 8	
38 14 9	
39 3 11	
40 5 6	
41 20 24	
42 1 17	
43 4 18	
44 27 23	
45 11 14	
46 13 12	
47 17 22	

Ln 12, Col 1 Spaces: 4 UTF-8 CRLF {} C++ 🖨 Win32

Hình 12. 9. Test case 9 bài 12.