

Bài thực hành số 3

Mục lục

Bài thực hành số 3	1
Bài 3.1	2
Bài 3.2	3
Bài 3.3	5
Bài 3.4	7
Bài 3.5	9
Bài 3.6	12
Bài 3.7	13
Bài 3.9	15
Bài 3.10	16
Bài 3.11	17
Bài 3.12	19

Bài 3.1

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

Time left 23:48:06

```
1 // Nguyen Duong Anh 20210082
2 int lucas(int n) {
3     if(n == 0) return 2;
4     else if(n==1) return 1;
5     return lucas(n-1) + lucas(n-2);
6 }
7 // Nguyen Duong Anh 20210082
```

Precheck

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

Passed all tests! ✓

```
// Nguyen Duong Anh 20210082
int lucas(int n) {
    if(n == 0) return 2;
    else if(n==1) return 1;
    return lucas(n-1) + lucas(n-2);
}
// Nguyen Duong Anh 20210082
```

Bài 3.2

```

1 // Nguyễn Dương Ánh 20210082
2 #include<stdio.h>
3 #include<stdlib.h>
4 #include <iostream>
5 using namespace std;
6
7 int n;
8 int X[100], Y[100]; // Lưu tọa độ các bước đi chuyển của quân mã
9 int mark[100][100]; // Đánh dấu vị trí các ô mà quân mã đã đi chuyển qua
10
11 // Mảng hx, hy mô tả 8 vị trí quân mã có thể đi chuyển kể từ vị trí hiện tại
12 const int hx[] = {1, 1, 2, 2, -1, -1, -2, -2};
13 const int hy[] = {2, -2, 1, -1, 2, -2, 1, -1};
14
15 // In ra dãy các đi chuyển tìm được
16 void print_sol(){
17     for (int j = 1; j <= n * n; ++j)
18         printf("(%d %d)\n", X[j], Y[j]);
19     exit(0);
20 }
21
22 bool check(int xx, int yy){
23     if(mark[xx][yy] == 1) return false; // đi qua rồi
24     if(xx < 1 || xx > n || yy < 1 || yy > n) return false; // đi ra ngoài tọa độ 1-n
25     //cout << xx << " " << yy << endl;
26     return true;
27 }
28
29 // Thuật toán quay lui
30 void TRY(int k){
31     for(int i = 0; i < 8; i++){
32         int xx = X[k-1] + hx[i]; // đi chuyển theo trục x
33         int yy = Y[k-1] + hy[i]; // đi chuyển theo trục y
34
35         if(check(xx,yy)){
36             mark[xx][yy] = 1;
37             X[k] = xx;
38             Y[k] = yy;
39
40             if(k == n*n ) print_sol();
41             else TRY(k+1);
42             mark[xx][yy] = 0;
43         }
44     }
45 }
46
47 int main(){
48     cin >> n;
49     mark[1][1] = 1;
50     X[1] = Y[1] = 1;
51     TRY(2);
52     return 0;
53 }
54 // Nguyễn Dương Ánh 20210082

```

Precheck [Check](#)

	Input	Expected	Got	
✓	5	(1 1) (2 3) (3 5) (5 4) (4 2) (2 1) (3 3) (1 4) (2 2) (4 1) (5 3) (4 5) (2 4) (1 2) (3 1) (5 2) (4 4) (2 5) (1 3) (3 2) (5 1) (4 3) (5 5) (3 4) (1 5)	(1 1) (2 3) (3 5) (5 4) (4 2) (2 1) (3 3) (1 4) (2 2) (4 1) (5 3) (4 5) (2 4) (1 2) (3 1) (5 2) (4 4) (2 5) (1 3) (3 2) (5 1) (4 3) (5 5) (3 4) (1 5)	✓
✓	6	(1 1) (2 3) (3 5) (4 3) (5 5) (6 3) (5 1) (3 2) (5 3) (6 1) (4 2)	(1 1) (2 3) (3 5) (4 3) (5 5) (6 3) (5 1) (3 2) (5 3) (6 1) (4 2)	✓

		(1 6)	(1 6)	
✓	7	(1 1)	(1 1)	✓
		(2 3)	(2 3)	
		(3 5)	(3 5)	
		(4 7)	(4 7)	
		(5 5)	(5 5)	
		(6 7)	(6 7)	
		(7 5)	(7 5)	
		(6 3)	(6 3)	
		(7 1)	(7 1)	
		(5 2)	(5 2)	
		(6 4)	(6 4)	
		(7 6)	(7 6)	
		(5 7)	(5 7)	
		(6 5)	(6 5)	
		(7 7)	(7 7)	
		(5 6)	(5 6)	
		(4 4)	(4 4)	
		(3 6)	(3 6)	
		(2 4)	(2 4)	
		(1 6)	(1 6)	
		(3 7)	(3 7)	
		(4 5)	(4 5)	
		(5 3)	(5 3)	
		(7 2)	(7 2)	
		(5 1)	(5 1)	
		(3 2)	(3 2)	
		(1 3)	(1 3)	
		(2 1)	(2 1)	
		(4 2)	(4 2)	
		(6 1)	(6 1)	
		(7 3)	(7 3)	
		(5 4)	(5 4)	
		(6 6)	(6 6)	
		(7 4)	(7 4)	
		(6 2)	(6 2)	
		(4 3)	(4 3)	
		(3 1)	(3 1)	
		(1 2)	(1 2)	
		(3 3)	(3 3)	
		(4 1)	(4 1)	
		(2 2)	(2 2)	
		(1 4)	(1 4)	
		(1 6)	(2 6)	
		(3 4)	(3 4)	
		(1 5)	(1 5)	
		(2 7)	(2 7)	
		(4 6)	(4 6)	
		(2 5)	(2 5)	
		(1 7)	(1 7)	

```
// Nguyen Duong Anh 20210082
#include<stdio.h>
#include<stdlib.h>
#include <iostream>
using namespace std;

int n;
int X[100], Y[100]; // Lưu tọa độ các bước di chuyển của quân mã
int mark[100][100]; // Đánh dấu vị trí các ô mà quân mã đã di chuyển qua

// Mảng hx, hy mô tả 8 vị trí quân mã có thể di chuyển kể từ vị trí hiện tại
const int hx[] = {1, 1, 2, 2, -1, -1, -2, -2};
const int hy[] = {2, -2, 1, -1, 2, -2, 1, -1};

// In ra dãy các di chuyển tìm được
void print_sol(){
    for (int j = 1; j <= n * n; ++j)
        printf("(%d %d)\n", X[j], Y[j]);
    exit(0);
}

bool check(int xx, int yy){
    if(mark[xx][yy] == 1) return false; // đi qua rồi
    if(xx < 1 || xx > n || yy < 1 || yy > n) return false; // đi ra ngoài tọa độ 1-n
    return true;
}

// Thuật toán quay lui
void TRY(int k){
    for(int i = 0; i < 8; i++){
        int xx = X[k-1] + hx[i]; // di chuyển theo trục x
```

```

        int yy = Y[k-1] + hy[i];    // di chuyển theo trục y

        if(check(xx,yy)){
            mark[xx][yy] = 1;
            X[k] = xx;
            Y[k] = yy;

            if(k == n*n ) print_sol();
            else TRY(k+1);
            mark[xx][yy] = 0;
        }
    }
}

int main(){
    cin >> n;
    mark[1][1] = 1;
    X[1] = Y[1] = 1;
    TRY(2);
    return 0;
}
// Nguyen Duong Anh 20210082

```

Bài 3.3

```

1 // Nguyen Duong Anh 20210082
2 #include <iostream>
3 using namespace std;
4 #define MAX 100
5
6 int n, c[MAX][MAX]; // n số thành phố và m trên chỉ phí
7 int min = INT_MAX; // n chỉ phí đã lại nhỏ nhất giữa hai thành phố khác nhau
8 int best = INT_MAX; // n tổng chỉ phí nhỏ nhất cần tìm, ban đầu đặt bằng giá trị vô cùng lớn INT_MAX = 2^31-1
9 int curr; // n tổng chỉ phí tối thiểu hiện tại
10 int mark[MAX]; // n đánh dấu những thành phố đã đi
11 int x[MAX]; // n lưu giữ các thành phố đã đi
12
13 // Đọc dữ liệu vào
14 void input(){
15     cin >> n;
16     for (int i = 1; i <= n; ++i)
17         for (int j = 1; j <= n; ++j){
18             cin >> c[i][j];
19             if (c[i][j] > 0) min = min(min, c[i][j]);
20         }
21     for (int i = 0; i < MAX; ++i)
22         mark[i] = 0;
23     curr = 0;
24 }
25
26 bool check(int k, int i){
27     if(mark[i] == 1) return false; // tp đã đi qua
28     return true;
29 }
30
31 // cập nhật lại lúc mới
32 void solution(){
33     best = min(best, curr);
34 }
35
36 // Thuật toán quay lui
37 void TRY(int k){
38     for (int i = 2; i <= n; ++i){
39         if(check(i,1)){
40             mark[i] = 1;
41             curr += c[x[k-1]][i];
42             // di tiếp từ x[k]
43             x[k] = i;
44
45             // lần đầu chạy từ đầu đến cuối, lấy 1 kg đầu để so sánh
46             if(k == n) {
47                 curr += c[x[n]][1];
48                 solution();
49                 curr -= c[x[n]][1];
50             }
51             else TRY(k+1);
52
53             mark[i] = 0;
54             curr -= c[x[k-1]][i];
55         }
56     }
57 }
58
59 int main() {
60     input();
61     x[1] = 1;
62     TRY(2);
63     cout << best;
64     return 0;
65 }
66 // Nguyen Duong Anh 20210082

```

Precheck
 Check

Time left 42:15/

66 }
67 // Nguyen Duong Anh 20210082

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 4 0 9 2 8 5 13 2 2 4 0 3 6 9 1 5 3 3 2 0 2 3 7 3 2 7 3 9 0 4 1 9 2 1 4 5 6 0 5 4 1 6 7 2 4 5 0 10 12 2 3 1 4 5 6 0	14	14	✓

Passed all tests! ✓

```
// Nguyen Duong Anh 20210082
#include<stdio.h>
#include<stdlib.h>
#include <iostream>
using namespace std;

int n;
int X[100], Y[100]; // Lưu tọa độ các bước di chuyển của quân mã
int mark[100][100]; // Đánh dấu vị trí các ô mà quân mã đã di chuyển qua

// Mảng hx, hy mô tả 8 vị trí quân mã có thể di chuyển kể từ vị trí hiện tại
const int hx[] = {1, 1, 2, 2, -1, -1, -2, -2};
const int hy[] = {2, -2, 1, -1, 2, -2, 1, -1};

// In ra dãy các di chuyển tìm được
void print_sol(){
    for (int j = 1; j <= n * n; ++j)
        printf("(%d %d)\n", X[j], Y[j]);
    exit(0);
}

bool check(int xx, int yy){
    if(mark[xx][yy] == 1) return false; // đi qua rồi
    if(xx < 1 || xx > n || yy < 1 || yy > n) return false; // đi ra ngoài tọa độ 1-n
    //cout << xx << " " << yy << endl;
    return true;
}

// Thuật toán quay lui
void TRY(int k){
    for(int i = 0; i < 8; i++){
        int xx = X[k-1] + hx[i]; // di chuyển theo trục x
        int yy = Y[k-1] + hy[i]; // di chuyển theo trục y

        if(check(xx,yy)){
            mark[xx][yy] = 1;
            X[k] = xx;

```

```

        Y[k] = yy;

        if(k == n*n ) print_sol();
        else TRY(k+1);
        mark[xx][yy] = 0;
    }
}

int main(){
    cin >> n;
    mark[1][1] = 1;
    X[1] = Y[1] = 1;
    TRY(2);
    return 0;
}
// Nguyen Duong Anh 20210082

```

Bài 3.4

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

Time left 15:49:51

```

1 // Nguyen Duong Anh 20210082
2 #include <bits/stdc++.h>
3 using namespace std;
4 int a[1000], n;
5 int mem[1000]; // mảng ghi nhớ lời giải các bài toán con đã được giải
6
7 void init(){
8     memset(mem, -1, sizeof(mem));
9 }
10
11 // Quy hoạch động.
12 // Hàm lis(i) trả về độ dài dãy con tăng dài nhất kết thúc bởi a[i]
13 void incsubseq(){
14     mem[0] = 1;
15
16     for(int i=1; i<n; i++){
17         mem[i] = 1;
18         for(int j=0; j<i; j++){
19             if(a[j] < a[i]){
20                 if(mem[i] < mem[j] + 1){
21                     mem[i] = mem[j] + 1;
22                 }
23             }
24         }
25     }
26 }
27 int lis(int i) {
28     return mem[i];
29 }
30
31 // Truy vết lời giải
32 void trace(int i){
33     for(int j = 0; j < i; j++){
34         if (a[j] < a[i] && mem[i] == 1 + mem[j]){
35             trace(j);
36             break;
37         }
38     }
39     cout << a[i] << " ";
40 }
41
42 int main(){
43     init();
44     cin >> n;
45     for(int i = 0; i < n; i++) cin >> a[i];
46
47     incsubseq();
48
49     int res = 1, pos = 0;
50     for(int i = 1; i < n; i++){
51         if (res < lis(i)){
52             res = lis(i);
53             pos = i;
54         }
55     }
56     cout << res << endl;
57     trace(pos);
58     return 0;
59 }
60 // Nguyen Duong Anh 20210082

```

59	}		
60	// Nguyen Duong Anh 20210082		

Precheck

	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! ✓

```
// Nguyen Duong Anh 20210082
#include <bits/stdc++.h>
using namespace std;
int a[1000], n;
int mem[1000]; // # mảng ghi nhớ lời giải các bài toán con đã được giải

void init(){
    memset(mem, -1, sizeof(mem));
}

// # Quy hoạch động,
// # Hàm lis(i) trả về độ dài dãy con tăng dài nhất kết thúc bởi a[i]
void incsubseq(){
    mem[0] = 1;

    for(int i=1; i<n; i++){
        mem[i] = 1;
        for(int j=0; j<i; j++){
            if(a[j] < a[i]){
                if(mem[i] < mem[j] + 1){
                    mem[i] = mem[j] + 1;
                }
            }
        }
    }
}

int lis(int i) {
    return mem[i];
}

// # Truy vết lời giải
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(){
    init();
    cin >> n;
    for(int i = 0; i < n; i++) cin >> a[i];

    incsubseq();

    int res = 1, pos = 0;
    for(int i = 1; i < n; i++){
        if (res < lis(i)){
            res = lis(i);
            pos = i;
        }
    }
    cout << res << endl;
    trace(pos);
    return 0;
}
// Nguyen Duong Anh 20210082

```

Bài 3.5

Time left 15:46:34

```

1 // Nguyen Duong Anh 20210082
2 #include<stdio.h>
3 #include <iostream>
4 using namespace std;
5 const int MAX = 1000;
6
7 int binom(int n, int k) {
8     if (k > n) return 0;
9     if (k == 0) return 1;
10    return binom(n-1, k) + binom(n-1, k-1);
11 }
12
13 int binom2(int n, int k){
14     // * Khử đệ quy
15     long result = 1; //(nC0 = 1)
16     for (int i = 1; i <= k; i++)
17     {
18         result = result * n / i;
19         n--;
20     }
21     return result;
22 }
23
24
25 int main() {
26     int m;
27     cin >> m;
28     for (int n = 1; n <= m; ++n){
29         for (int k = 0; k <= n; ++k)
30             printf("%d ", binom(n, k));
31         printf("\n");
32     }
33     for (int n = 1; n <= m; ++n){
34         for (int k = 0; k <= n; ++k)
35             printf("%d ", binom2(n, k));
36         printf("\n");
37     }
38     return 0;
39 }
40 // Nguyen Duong Anh 20210082

```

Precheck
Check

Nguyễn Dương Ánh - 20210082

```
35         printf("%d ", binom2(n, k));
36     }
37     printf("\n");
38     return 0;
39 }
40 // Nguyen Duong Anh 20210082
```

Precheck

	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 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	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

		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 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	
✓	20	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 1001 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 1 17 136 680 2380 6188 12376 19448 24310 24310 19448 12376 6188 2380 680 136 17 1 1 18 153 816 3060 8568 18564 31824 43758 48620 43758 31824 18564 8568 3060 816 153 18 1 1 19 171 960 3876 11628 27132 50388 75582 92378 92378 75582 50388 27132 11628 3876 960 171 19 1 1 20 190 1140 4845 15504 38760 77520 125970 167960 184756 167960 125970 77520 38760 15504 4845 1140 190 20 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 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 1001 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 1 17 136 680 2380 6188 12376 19448 24310 24310 19448 12376 6188 2380 680 136 17 1 1 18 153 816 3060 8568 18564 31824 43758 48620 43758 31824 18564 8568 3060 816 153 18 1 1 19 171 960 3876 11628 27132 50388 75582 92378 92378 75582 50388 27132 11628 3876 960 171 19 1 1 20 190 1140 4845 15504 38760 77520 125970 167960 184756 167960 125970 77520 38760 15504 4845 1140 190 20 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 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 1001 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 1 17 136 680 2380 6188 12376 19448 24310 24310 19448 12376 6188 2380 680 136 17 1 1 18 153 816 3060 8568 18564 31824 43758 48620 43758 31824 18564 8568 3060 816 153 18 1 1 19 171 960 3876 11628 27132 50388 75582 92378 92378 75582 50388 27132 11628 3876 960 171 19 1 1 20 190 1140 4845 15504 38760 77520 125970 167960 184756 167960 125970 77520 38760 15504 4845 1140 190 20 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 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 1001 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 1 17 136 680 2380 6188 12376 19448 24310 24310 19448 12376 6188 2380 680 136 17 1 1 18 153 816 3060 8568 18564 31824 43758 48620 43758 31824 18564 8568 3060 816 153 18 1 1 19 171 960 3876 11628 27132 50388 75582 92378 92378 75582 50388 27132 11628 3876 960 171 19 1 1 20 190 1140 4845 15504 38760 77520 125970 167960 184756 167960 125970 77520 38760 15504 4845 1140 190 20 1	✓

Passed all tests! ✓

```
// Nguyen Duong Anh 20210082
#include<stdio.h>
#include <iostream>
using namespace std;
const int MAX = 1000;

int binom(int n, int k) {
    if (k > n) return 0;
    if (k == 0) return 1;
    return binom(n-1, k) + binom(n-1, k-1);
}
```

```
}

int binom2(int n, int k){

    // # Khử đệ quy
    long result = 1; //(nC0 = 1)
    for ( int i = 1; i <= k ; i++)
    {
        result = result * n / i;
        n--;
    }
    return result;
}

int main() {
    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;
}

// Nguyen Duong Anh 20210082
```

Bài 3.6

```

1 // Nguyen Duong Anh 20210082
2 #include <iostream>
3 using namespace std;
4
5 int gcd(int a, int b){
6     if (b == 0) return a;
7     return gcd(b, a % b);
8 }
9
10 int gcd2(int a, int b){
11
12     //# Khử đệ quy
13     while(b != 0){
14         int tmp = a % b;
15         a = b;
16         b = tmp;
17     }
18
19     return a;
20 }
21
22 int main() {
23     int a, b;
24     cin >> a >> b;
25     cout << gcd(a, b) << endl << gcd2(a, b);
26     return 0;
27 }
28 // Nguyen Duong Anh 20210082

```

Precheck

Check

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

Passed all tests! ✓

```

// Nguyen Duong Anh 20210082
#include <iostream>
using namespace std;

int gcd(int a, int b){
    if (b == 0) return a;
    return gcd(b, a % b);
}

int gcd2(int a, int b){

    //# Khử đệ quy
    while(b != 0){
        int tmp = a % b;
        a = b;
        b = tmp;
    }

    return a;
}

int main() {
    int a, b;
    cin >> a >> b;
    cout << gcd(a, b) << endl << gcd2(a, b);
    return 0;
}
// Nguyen Duong Anh 20210082

```

Bài 3.7

```

1 // Nguyen Duong Anh 20210082
2 #include <bits/stdc++.h>
3 using namespace std;
4 struct state{
5     int i, j, old_L;
6     // constructor
7     state(int _i = 0, int _j = 0, int _L = 0):
8         i(_i), j(_j), old_L(_L){}
9 };
10
11 int main() {
12     int n, k;
13     cin >> n >> k;
14     int x[n+1];
15     stack<state> s;
16     // number of consecutive suffix 1
17     int L = 0;
18     s.push(state(1, 0));
19     while (!s.empty()){
20         state &top = s.top();
21         // if a new binary sequence is found
22         if (top.i > n){
23             for (int i = 1; i <= n; ++i)
24                 cout << x[i] << " \n"[i == n];
25             s.pop();
26             continue;
27         }
28
29         // Khử đệ quy
30         if(top.j>0) L = top.old_L;
31
32         if(top.j>1){
33             s.pop();
34             continue;
35         }
36
37         if(L+1<k || top.j==0){
38             x[top.i] = top.j;
39             top.old_L = L;
40             if(top.j){
41                 if(L+1<k || top.j==0){
42                     x[top.i] = top.j;
43                     top.old_L = L;
44                     if(top.j){
45                         L = L + 1;
46                     } else {L = 0;}
47                     s.push(state(top.i+1,0));
48                 }
49                 top.j++;
50             }
51             return 0;
52         }
53     }
54 // Nguyen Duong Anh 20210082
                    
```

Time left 15:42:28

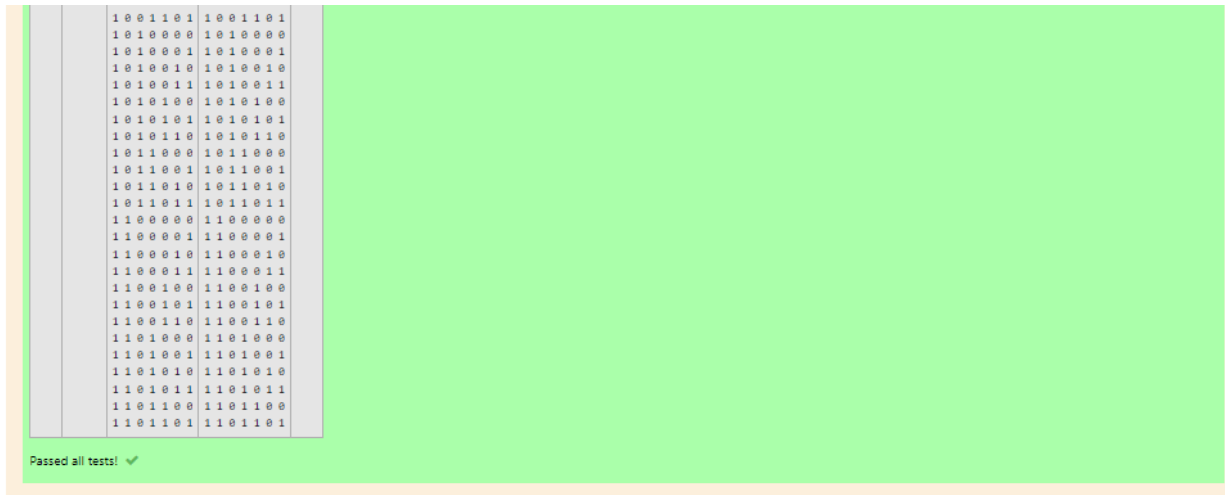
```

37         if(L+1<k || top.j==0){
38             x[top.i] = top.j;
39             top.old_L = L;
40             if(top.j){
41                 L = L + 1;
42             } else {L = 0;}
43             s.push(state(top.i+1,0));
44         }
45         top.j++;
46     }
47     return 0;
48 }
49 // Nguyen Duong Anh 20210082
                    
```

Time left 15:42:06

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	✓
✓	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	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	✓



```
// Nguyen Duong Anh 20210082
#include <bits/stdc++.h>
using namespace std;
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() {
    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;
        }

        //# Khử đệ 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;
}
// Nguyen Duong Anh 20210082

```

Bài 3.9

```

// Nguyễn Dương Ánh 20210082
#include<stdio.h>
#include<stdlib.h>
#include <iostream>
using namespace std;

int n,k1,k2;
int *schedule; // mảng để chứa giá trị ngày làm việc (1) và ngày nghỉ (0)
int streak = 0; // chuỗi ngày làm việc liên tiếp

void print_sol(){
    for(int i=0; i<n;i++){
        cout<<schedule[i]<<" ";
    }
    cout<<"\n";
}

bool check(int prev, int current, int dayNumber){
    if (prev == 0 && current == 0 ) return false; // không được
    có 2 ngày nghỉ liên tiếp
    else if (prev == 1 && streak < k1 && current == 0) return false; // chưa đủ số
    ngày làm tối thiểu
    else if (streak >= k2 && current ==1) return false; // quá số ngày
    làm tối đa
    else if (dayNumber == n-1 && current == 1 && streak <k1-1) return false; // đến ngày
    cuối nhưng chưa đủ số ngày làm tối thiểu
    return true;
}

// ## Thuật toán quay lui
void TRY(int k){
    for(int i = 0; i < 2; i++){
        int current = i;
        int prevStreak = streak;

        // cout<< "dayNumber: "<<k<<" current: "<< current << " streak: "<<streak<<"\n";
        if(check(schedule[k-1],current, k)){
            if(i==1) streak ++;

```

```

        else if(i==0) streak =0;
        // cout<< "dayNumber: "<<k<<" current: "<< current << " streak: "<<streak<<"\n";

        schedule[k] = current;
        if(k==n-1) print_sol();    // nếu thử đến cuối rồi thì in ra
        else TRY(k+1);

        // khôi phục chuỗi trước đó
        streak = prevStreak;
    }
}

int main(){
    cin>> n>>k1>>k2;
    // cấp phát động cho schedule[]
    schedule=(int*)malloc(n * sizeof(int));

    for (int i=0; i<=1; i++){
        schedule[0] = i;
        streak = i;
        TRY(1);
    }
    return 0;
}
// Nguyễn Dương Ánh 20210082

```

Bài 3.10

```

// Nguyễn Dương Ánh 20210082
#include<iostream>
using namespace std;
const int MAX = 20;
int N, H;
int x[MAX];
int S[MAX]; // Xâu gốc để so sánh
int count = 0;

void input(){
    cin >> N >> H;
    for(int i=0; i<N; i++) S[i] = 0;
    for(int i=0; i<N; i++) x[i] = 0;
}

// Kiểm tra khoảng cách Hamming của 2 xâu
int check(int str1[], int str2[]){
    int lens = N;
    int count = 0;
    for(int i=0; i<lens; i++){
        if(str1[i] != str2[i]) count++;
    }
    return count;
}

```



```
bool check(int a, int i){
    return true;
}

void solution(){
    if(check(x,S) == H){
        for(int i=0; i<N; i++) cout << x[i];
        cout << endl;
    }
}

void TRY(int a){
    for(int i=0; i<=1; i++){
        x[a] = i;
        if(a == N-1) solution();
        else
            TRY(a+1);
    }
}

int main(){
    int k;
    cin >> k;
    while(k > 0){
        input();
        TRY(0);
        k--;
    }
}
// Nguyễn Dương Ánh 20210082
```

Bài 3.11

```
// Nguyễn Dương Ánh 20210082
#include<bits/stdc++.h>
using namespace std;
const int MAX = 10000;
int n, r;
int cost[MAX][MAX];
int x[MAX];
bool visited[MAX];
vector<int> vt;
int min_cost;
int sum_cost;
int start, destination, numberOfPoint;

void input(){
    cin >> n >> r;
    for(int i=0; i<n; i++)
        for(int j=0; j<n; j++){
            cin >> cost[i][j];
        }
}
```

```

}

bool check(int a, int i){
    if(visited[vt[i]]) return false;
    if(cost[x[a-1]][vt[i]] == 0) return false;
    return true;
}

void solution(){
    if(cost[x[numberOfPoint-2]][destination] == 0) return;
    min_cost = min(min_cost, sum_cost + cost[x[numberOfPoint-2]][destination]);
}

void TRY(int a){
    for(int i=1; i<numberOfPoint-1; i++){
        if(check(a, i)){
            visited[vt[i]] = true;
            sum_cost += cost[x[a-1]][vt[i]];

            x[a] = vt[i];
            if(a == numberOfPoint-2) solution();
            else TRY(a+1);

            visited[vt[i]] = false;
            sum_cost -= cost[x[a-1]][vt[i]];
        }
    }
}

int main(){
    string str;
    input(); getline(cin, str);

    while(r > 0){
        min_cost = INT_MAX;
        sum_cost = 0;

        getline(cin, str);

        while (!str.empty()){
            stringstream convert(str.substr(0, str.find(" ")));
            int tmp = 0;
            convert >> tmp;
            vt.push_back(tmp - 1);

            if (str.find(" ") > str.size()){
                break;
            } else {
                str.erase(0, str.find(" ") + 1);
            }
        }

        start = vt[0];
        destination = vt[vt.size()-1];
    }
}

```

```

        numberOfPoint = vt.size();
        x[0] = start; x[numberOfPoint-1] = destination;
        for(int i=0; i<n; i++)
            visited[i] = false;

        TRY(1);

        if(min_cost == INT_MAX) cout << "0" << endl;
        else cout << min_cost << endl;

        vt.erase(vt.begin(), vt.end());
        r--;
    }
}
// Nguyễn Dương Ánh 20210082

```

Bài 3.12

```

// Nguyễn Dương Ánh 20210082
#include<bits/stdc++.h>
using namespace std;
const int MAX = 100;
int n, k;
int m;
vector<vector<int> > vt;
int x[MAX];
bool visited[MAX];
int res;

void input(){
    cin >> n >> k;
    cin >> m;
    vt.resize(n+1);
    for(int i=0; i<m; i++){
        int t1, t2;
        cin >> t1 >> t2;

        vt[t1-1].push_back(t2-1);
        vt[t2-1].push_back(t1-1);
    }

    for(int i=0; i<n; i++){
        visited[i] = false;
    }
    res = 0;
}

bool check(int a, int i){
    if(a == 0) return true;
    if(visited[i]) return false;

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

```

```
        if(i == vt[x[a-1]][j]) index++;
    }
    if(index == 0) return false;

    return true;
}

void solution(){
    res++;
}

void TRY(int a){
    for(int i=0; i<n; i++){
        if(check(a, i)){
            visited[i] = true;
            x[a] = i;

            if(a == k) solution();
            else TRY(a+1);

            visited[i] = false;
        }
    }
}

int main(){
    input();
    TRY(0);
    cout << res / 2;
}
// Nguyễn Dương Ánh 20210082
```