

BÁO CÁO THỰC HÀNH TUẦN 38
HỌC PHẦN: IT3040 - KỸ THUẬT LẬP TRÌNH - 20221

Mục lục

Bài thực hành số 3

Bài 3.1. Dãy Lucas được định nghĩa bởi $L_n = L_{n-1} + L_{n-2}$ với $L_0 = 2, L_1 = 1$. Hãy viết hàm tính số Lucas thứ n	1
Bài 3.2. 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 (ô (1, 1) được xem là đã đi qua).....	1
Bài 3.3. 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_{ij} . Hãy tìm tổng chi phí nhỏ nhất có thể	2
Bài 3.4. 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	4
Bài 3.5. Tính hệ số tổ hợp $C(n, k)$	7
Bài 3.6. Tìm ước chung lớn nhất của hai số nguyên a, b cho trước.	8
Bài 3.7. 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	9
Phần 3. Bài tập về nhà	10
Bài tập 9: Lập lịch cho y tá	10
Bài tập 10: Khoảng cách Hamming	11
Bài tập 11: Lịch trình chụp ảnh	12
Bài tập 12: Đếm đường đi	14

Bài 3.1. Dãy Lucas được định nghĩa bởi $L_n = L_{n-1} + L_{n-2}$ với $L_0 = 2, L_1 = 1$. Hãy viết hàm tính số Lucas thứ n .

My courses Theme colours This course 3 Brak Lihou

Grade 100.00 out of 100.00

Question 1
Correct
Mark 10.00 out of 10.00
Flag question

Bài 3.1. Dãy Lucas được định nghĩa bởi $L_n = L_{n-1} + L_{n-2}$ với $L_0 = 2, L_1 = 1$. Hãy viết hàm tính số Lucas thứ n .

For example:

Test	Result
cout << lucas(5);	11

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

```

1 #include<iostream>
2 using namespace std;
3
4 int lucas(int n) {
5     if(n==0) return 2;
6     if(n==1) return 1;
7
8     else return lucas(n-1)+lucas(n-2);
9 }
10
11

```

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

bail.cpp

```

1 #include<iostream>
2 using namespace std;
3
4 int lucas(int n) {
5     if(n==0) return 2;
6     if(n==1) return 1;
7
8     else return lucas(n-1)+lucas(n-2);
9 }
10
11
12 int main(){
13
14     printf("-- Brak Lihou --\n");
15     printf("-- 20200836--\n\n");
16
17     int n;
18     cout<<lucas(5);
19
20     return 0;
21 }

```

C:\Users\Anh Xa\Desktop\vt\Brak_Lihou_20200836_715026_3\bail.exe

```

-- Brak Lihou --
-- 20200836--
11
-----
Process exited after 0.1109 seconds with return value 0
Press any key to continue . . .

```

Compile Log Debug Find Results

Bài 3.2. 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 (ô (1, 1) được xem là đã đi qua).

```

Theme colours This course 3 Brak Lihou
(3 4)
(1 5)

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

1 #include <iostream>
2 using namespace std;
3 int n;
4 int X[100], Y[100];
5 int mark[100][100];
6 const int hx[] = {1, 1, 2, 2, -1, -1, -2, -2};
7 const int hy[] = {2, -2, 1, -1, 2, -2, 1, -1};
8
9 void print_sol(){
10     for (int j = 1; j <= n * n; ++j)
11         printf("%d %d\n", X[j], Y[j]);
12     exit(0);
13 }
14 bool check(int xx, int yy){
15     if(mark[xx][yy] == 1) return false;
16     if(xx < 1 || xx > n || yy < 1 || yy > n) return false;
17     return true;
18 }
19
20 void TRY(int k){
21     for(int i = 0; i < 8; i++){
22         int xx = X[k-1] + hx[i];
23         int yy = Y[k-1] + hy[i];
24         if(check(xx,yy)){
25             mark[xx][yy] = 1;
26             X[k] = xx;
27             Y[k] = yy;
28             if(k == n*n)
29                 print_sol();
30             else TRY(k+1);
31             mark[xx][yy] = 0;
32         }
33     }
34 }
35
36 int main(){
37     cin >> n;
38     mark[1][1] = 1;
39     X[1] = Y[1] = 1;
40     TRY(2);
41     return 0;
42 }

```

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

bai1.cpp
bai2.cpp

```

1 #include <iostream>
2 using namespace std;
3 int n;
4 int X[100], Y[100];
5 int mark[100][100];
6 const int hx[] = {1, 1, 2, 2, -1, -1, -2, -2};
7 const int hy[] = {2, -2, 1, -1, 2, -2, 1, -1};
8
9 void print_sol(){
10     for (int j = 1; j <= n * n; ++j)
11         printf("%d %d\n", X[j], Y[j]);
12     exit(0);
13 }
14 bool check(int xx, int yy){
15     if(mark[xx][yy] == 1) return false;
16     if(xx < 1 || xx > n || yy < 1 || yy > n) return false;
17     return true;
18 }
19 void TRY(int k){
20     for(int i = 0; i < 8; i++){
21         int xx = X[k-1] + hx[i];
22         int yy = Y[k-1] + hy[i];
23         if(check(xx,yy)){
24             mark[xx][yy] = 1;
25             X[k] = xx;
26             Y[k] = yy;
27             if(k == n*n)
28                 print_sol();
29             else TRY(k+1);
30             mark[xx][yy] = 0;
31         }
32     }
33 }
34
35 int main(){
36     cin >> n;
37     mark[1][1] = 1;
38     X[1] = Y[1] = 1;
39     TRY(2);
40     return 0;
41 }

```

Brak Lihou
 20200836


```

5
(1 1)
(2 3)
(3 5)
(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)

```

3.3. 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_{ij} . Hãy tìm tổng chi phí nhỏ nhất có thể

Dữ liệu vào:

Dòng 1: Chứa số nguyên n ($1 \leq n \leq 16$)

n dòng tiếp theo: Chứa ma trận c ($0 \leq c_{ij} \leq 10000000$)

Kết quả: Ghi tổng chi phí nhỏ nhất có thể

es ▶ Theme colours ▶ This course ▶ 3 Brak Lihou ▶

4 0 1 2
2 1 0 3
3 4 2 0

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

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define MAX 100
4 int n, c[MAX][MAX];
5 int cmin = INT_MAX;
6 int best = INT_MAX;
7 int curr;
8 int mark[MAX];
9 int x[MAX];
10 void input(){
11     cin >> n;
12     for (int i = 1; i <= n; ++i)
13         for (int j = 1; j <= n; ++j){
14             cin >> c[i][j];
15             if (c[i][j] > 0) cmin = min(cmin, c[i][j]);
16         }
17     for (int i = 0; i < MAX; ++i)
18         mark[i] = 0;
19     curr = 0;
```

Theme colours ▶ This course ▶ 3 Brak Lihou ▶

4 0 1 2
2 1 0 3
3 4 2 0

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

```
18         mark[i] = 0;
19         curr = 0;
20     }
21     bool check(int k, int i){
22         if(mark[i] == 1) return false;
23         return true;
24     }
25     void sol(){
26         best = min(best,curr);
```

es ▶ Theme colours ▶ This course ▶ 3 Brak Lihou ▶

4 0 1 2
2 1 0 3
3 4 2 0

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

```
37         curr += c[x[n]][1];
38         sol();
39         curr -= c[x[n]][1];
40     }
41     else TRY(k+1);
42
43     mark[i] = 0;
44     curr -= c[x[k-1]][i];
45 }
46 }
47 }
48 }
49
50 int main() {
51     input();
```

es ▶ Theme colours ▶ This course ▶ 3 Brak Lihou ▶

	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! ✓

Correct

Marks for this submission: 15.00/15.00.

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define MAX 100
4  int n, c[MAX][MAX];
5  int cmin = INT_MAX;
6  int best = INT_MAX;
7  int curr;
8  int mark[MAX];
9  int x[MAX];
10 void input(){
11     cin >> n;
12     for (int i = 1; i <= n; ++i)
13         for (int j = 1; j <= n; ++j){
14             cin >> c[i][j];
15             if (c[i][j] > 0) cmin = min(cmin, c[i][j]);
16         }
17     for (int i = 0; i < MAX; ++i)
18         mark[i] = 0;
19     curr = 0;
20 }
21 bool check(int k, int i){
22     if(mark[i] == 1) return false;
23     return true;
24 }
25 void sol(){
26     best = min(best, curr);
27 }
28 void TRY(int k){
29     for(int i = 2; i <= n; i++){
30         if(check(k,i)){
31             mark[i] = 1;
32             curr += c[x[k-1]][i];
33             x[k] = i;
34         }
35     }
36     if(k == n) sol();
37     TRY(k+1);
38 }
39
40 int main() {
41     printf("-- Brak Lihou --\n");
42     printf("-- 20200836 --\n\n");
43
44     input();
45     x[1] = 1;
46     TRY(2);
47     cout << best;
48     return 0;
49 }

```

```

-- Brak Lihou --
-- 20200836 --

4
0 2 1 3
4 0 1 2
2 1 0 3
3 4 2 0
7

Process exited after 25.74 seconds with return value 0
Press any key to continue . . .

```

Bài 3.4. Cho dãy a

có n phần tử. Một dãy con của a là dãy thu được bằng cách xóa đi một số phần tử của a và giữ nguyên thứ tự các phần tử còn lại (có thể không xóa phần tử nào). Hãy tìm dãy con tăng dài nhất của a .

Dữ liệu vào:

Dòng 1: Chứa số nguyên n ($1 \leq n \leq 1000$)

Dòng 2: Chứa n số nguyên a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$)

Kết quả:

Dòng đầu tiên chứa độ dài dãy con tăng dài nhất

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

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

es ▶ Theme colours ▶ This course ▶ 3 3 ▶ Brak Lihou ▶

2 1 5 4 3 6 | 2 5 6

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

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int a[1000], n;
4 int mem[1000];
5
6 void init(){
7     memset(mem, -1, sizeof(mem));
8 }
9
10 void sol(){
11     mem[0] = 1;
12
13     for(int i=1; i<n; i++){
14         mem[i] = 1;
15         for(int j=0; j<i; j++){
16             if(a[j] < a[i]){
17                 mem[i] = max(mem[i], mem[j] + 1);
18             }
19         }
20     }
21 }
22
23 int lis(int i) {
24     return mem[i];
25 }
26
27 //*****
28
29
```

es ▶ Theme colours ▶ This course ▶ 3 3 ▶ Brak Lihou ▶

2 1 5 4 3 6 | 2 5 6

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

```
18     mem[i] = mem[j] + 1;
19 }
20 }
21 }
22 }
23
24 int lis(int i) {
25     return mem[i];
26 }
27 //*****
28
29
```

es ▶ Theme colours ▶ This course ▶ 3 3 ▶ Brak Lihou ▶

2 1 5 4 3 6 | 2 5 6

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

```
31 void trace(int i){
32     for(int j = 0; j < i; j++){
33         if (a[j] < a[i] && mem[i] == 1 + mem[j]){
34             trace(j);
35             break;
36         }
37     }
38     cout << a[i] << " ";
39 }
40
41 int main(){
42     init();
43     cin >> n;
44     for(int i = 0; i < n; i++) cin >> a[i];
45
46     sol();
47
48     int res = 1, pos = 0;
49     for(int i = 1; i < n; i++){
50         if (res < lis(i)){
51             res = lis(i);
52             pos = i;
53         }
54     }
55     cout << res << endl;
56     trace(pos);
57     return 0;
58 }
```

es ▶ Theme colours ▶ This course ▶ 3 3 ▶ Brak Lihou ▶

2 1 5 4 3 6 | 2 5 6

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

```
43 sol();
44
45 int res = 1, pos = 0;
46 for(int i = 1; i < n; i++){
47     if (res < lis(i)){
48         res = lis(i);
49         pos = i;
50     }
51 }
52 cout << res << endl;
53 trace(pos);
54 return 0;
55 }
```

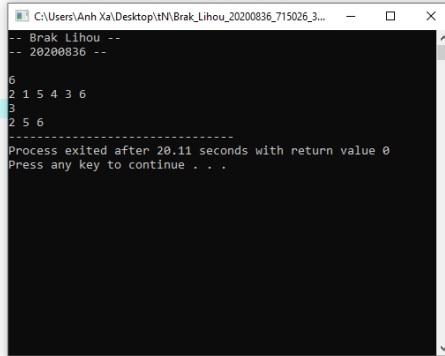
Check

	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

```

bai1.cpp bai2.cpp bai3.cpp bai4.cpp
1 #include <bits/stdc++.h>
2 using namespace std;
3 int a[1000], n;
4 int mem[1000];
5
6 void init()
7 {
8     memset(mem, -1, sizeof(mem));
9 }
10 void sol()
11 {
12     mem[0] = 1;
13     for(int i=1; i<n; i++){
14         mem[i] = 1;
15         for(int j=0; j<i; j++){
16             if(a[j] < a[i]){
17                 if(mem[i] < mem[j] + 1){
18                     mem[i] = mem[j] + 1;
19                 }
20             }
21         }
22     }
23 }
24
25 int lis(int i) {
26     return mem[i];
27     /******
28 }
29
30 // Truy vết lời giải
31 void trace(int i){
32     for(int j = 0; j < i; j++){
33         if (a[j] < a[i] && mem[i] == 1 + mem[j]){
34             trace(j);
35             break;
36         }
37     }
38     cout << a[i] << " ";
39 }
40
41 int main()
42 {
43     printf("-- Brak Lihou --\n");
44     printf("-- 20200836 --\n\n");
45
46     cout << a[i] << " ";
47 }
48
49 int main()
50 {
51     printf("-- Brak Lihou --\n");
52     printf("-- 20200836 --\n\n");
53     init();
54     cin >> n;
55     for(int i = 0; i < n; i++) cin >> a[i];
56
57     sol();
58
59     int res = 1, pos = 0;
60     for(int i = 1; i < n; i++){
61         if (res < lis(i)){
62             res = lis(i);
63             pos = i;
64         }
65     }
66     cout << res << endl;
67     trace(pos);
68     return 0;
69 }

```



Bài 3.5. Tính hệ số tổ hợp $C(n, k)$

es ▶ Theme colours ▶ This course ▶ 3 ▶ Brak Lihou ▶

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

```

1 #include <iostream>
2 using namespace std;
3
4 int binom(int n, int k) {
5     if (k > n) return 0;
6     if (k == 0) return 1;
7     return binom(n-1, k) + binom(n-1, k-1);
8 }
9
10 int binom2(int n, int k){
11
12     int res = 1;
13     for (int i = 1; i <= k; i++, n--){
14         res = res * n/i;
15     }
16     return res;
17 }
18
19
20 int main() {
21     int m;
22     cin >> m;
23     for (int n = 1; n <= m; ++n){
24         for (int k = 0; k <= n; ++k)
25             printf("%d ", binom(n, k));
26         printf("\n");
27     }
28     for (int n = 1; n <= m; ++n){
29         for (int k = 0; k <= n; ++k)
30             printf("%d ", binom2(n, k));
31         printf("\n");
32     }
33     return 0;
34 }

```

Check

	Input	Expected
✓	4	<pre> 1 1 1 2 1 1 3 3 1 </pre>

bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp

```

1 #include <iostream>
2 using namespace std;
3
4 int binom(int n, int k) {
5     if (k > n) return 0;
6     if (k == 0) return 1;
7     return binom(n-1, k) + binom(n-1, k-1);
8 }
9
10 int binom2(int n, int k){
11     int res = 1;
12     for (int i = 1; i <= k; i++, n--){
13         res = res * n/i;
14     }
15     return res;
16 }
17 int main() {
18     printf("-- Brak Lihou --\n");
19     printf("-- 20200836 --\n\n");
20     int m;
21     cin >> m;
22     for (int n = 1; n <= m; ++n){
23         for (int k = 0; k <= n; ++k)
24             printf("%d ", binom(n, k));
25

```

C:\Users\Anh Xa\Desktop\tn\Brak_Lihou_20200836_715026_31... — □ ×

```

-- Brak Lihou --
-- 20200836 --

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

-----
Process exited after 1.793 seconds with return value 0
Press any key to continue . . .

```



```

bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp
11     int res = 1;
12     for (int i = 1; i <= k; i++, n--){
13         res = res * n/i;
14     }
15     return res;
16 }
17 int main() {
18
19     printf("-- Brak Lihou --\n");
20     printf("-- 20200836 --\n\n");
21     int m;
22     cin >> m;
23     for (int n = 1; n <= m; ++n){
24         for (int k = 0; k <= n; ++k)
25             printf("%d ", binom(n, k));
26         printf("\n");
27     }
28     for (int n = 1; n <= m; ++n){
29         for (int k = 0; k <= n; ++k)
30             printf("%d ", binom2(n, k));
31         printf("\n");
32     }
33     return 0;
34 }

```

Bài
3.6. Tìm

ước chung lớn nhất của hai số nguyên a, b cho trước.

res ▶ Theme colours ▶ This course ▶ 3 ▶ Brak Lihou ▶

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

```

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

```

Check

```

10 }
17
18 int main() {
19     int a, b;
20     cin >> a >> b;
21     cout << gcd(a, b) << endl << gcd2(a, b);
22     return 0;
23 }

```

Check

C

```

bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp
1 #include <iostream>
2 using namespace std;
3 int gcd(int a, int b){
4     if (b == 0) return a;
5     return gcd(b, a % b);
6 }
7
8 int gcd2(int a, int b){
9     while(b != 0){
10         int tmp = a % b;
11         a = b;
12         b = tmp;
13     }
14
15     return a;
16 }
17
18 int main() {
19     printf("-- Brak Lihou --\n");
20     printf("-- 20200836 --\n\n");
21     int a, b;
22     cin >> a >> b;
23     cout << gcd(a, b) << endl << gcd2(a, b);
24     return 0;
25 }

```

```

C:\Users\Anh Xa\Desktop\tn\Brak_Lihou_20200836_715026_3\bai6.exe
-- Brak Lihou --
-- 20200836 --

50 35
5
5
-----
Process exited after 9.563 seconds with return value 0
Press any key to continue . . .

```

Bài 3.7. Sử dụng phương pháp khử đệ quy bằng stack, hãy liệt kê các xâu nhị phân độ dài n không có k bit 1 nào liên tiếp

Dữ liệu vào:

Một dòng duy nhất chứa hai số nguyên n, k ($1 \leq k \leq n \leq 20$)

Kết quả:

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

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 struct state{
4     int i, j, old_L;
5
6     state(int _i = 0, int _j = 0, int _L = 0):
7         i(_i), j(_j), old_L(_L){}
8 };
9 int main() {
10     int n, k;
11     cin >> n >> k;
12     int x[n+1];
13     stack<state> s;
14     int L = 0;
15     s.push(state(1, 0));
16     while (!s.empty()) {
17         state &top = s.top();
18         if (top.i > n) {
19             for (int i = 1; i <= n; ++i)
20                 cout << x[i] << " ";
21             cout << "\n";
22             s.pop();
23             continue;
24         }
25         if (top.j > 0) L = top.old_L;
26         if (top.j > 1) {
27             s.pop();
28             continue;
29         }
30         if (L+1 < k || top.j == 0) {
31             x[top.i] = top.j;
32             top.old_L = L;
33             if (top.i) {
34                 s.push(state(top.i+1, 0));
35                 top.j++;
36             }
37         }
38     }
39     return 0;
40 }

```

```

18 if (top.i > n) {
19     for (int i = 1; i <= n; ++i)
20         cout << x[i] << " ";
21     cout << "\n";
22     s.pop();
23     continue;
24 }
25 if (top.j > 0) L = top.old_L;
26 if (top.j > 1) {
27     s.pop();
28     continue;
29 }
30 if (L+1 < k || top.j == 0) {
31     x[top.i] = top.j;
32     top.old_L = L;
33     if (top.i) {
34         s.push(state(top.i+1, 0));
35         top.j++;
36     }
37 }
38 }
39 return 0;
40 }

```

```

33 if (top.j) {
34     L = L + 1;
35 } else { L = 0; }
36 s.push(state(top.i+1, 0));
37 top.j++;
38 }
39 }
40 return 0;
41 }
42 }

```

	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 0 0 0 0 0 0 1 0 0 0 1 0	✓

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 struct state{
4     int i, j, old_L;
5
6     state(int _i = 0, int _j = 0, int _L = 0):
7         i(_i), j(_j), old_L(_L){}
8 };
9 int main() {
10     printf("-- Brak Lihou --\n");
11     printf("-- 20200836 --\n");
12     int n, k;
13     cin >> n >> k;
14     int x[n+1];
15     stack<state> s;
16     int L = 0;
17     s.push(state(1, 0));
18     while (!s.empty()){
19         state &top = s.top();
20         if (top.i > n){
21             for (int i = 1; i <= n; ++i)
22                 cout << x[i] << " \n"[i == n];
23             s.pop();
24             continue;
25         }
26         if(top.j>0) L = top.old_L;
27
28         if(top.j>1){
29             s.pop();
30             continue;
31         }
32         if(L+1<k || top.j==0){
33             x[top.i] = top.j;
34             top.old_L = L;
35             if(top.j){
36                 L = L + 1;
37             } else {L = 0;}
38             s.push(state(top.i+1,0));
39         }
40         top.j++;
41     }
42     return 0;
43 }

```

Terminal Output:

```

-- Brak Lihou --
-- 20200836 --

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

Process exited after 7.483 seconds with return value 0
Press any key to continue . . .

```

Phần 3. Bài tập về nhà

Bài tập 9: Lập lịch cho y tá

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 const int MAX = 1000;
4 const int du = 1000000000 + 7;
5 int n, k1, k2;
6 int x[MAX];
7 int cnt = 0;
8 int so0 = 0, so1 = 0;
9
10 void inputData(){
11     cin >> n >> k1 >> k2;
12 }
13
14 bool check(int a, int i){
15     if(a==1) return true;
16     else {
17         // neu ngay nay nghi -> kiem tra ngay truuoc do, neu da nghi thi khong duoc nghi nua
18         if(i==0){
19             if(x[a-1] == 0) return false;
20             else { // x[a-1] == 1
21                 if(so1<k1) return false;
22             }
23         } else { // i == 1
24             if(x[a-1] == 0){
25                 if(n-a+1 < k1) return false;

```

Terminal Output:

```

6 2 3
011011
110110
110111
110111
111011

Process exited after 12.24 seconds with return value 0
Press any key to continue . . .

```

```

bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp bai7.cpp bai9.cpp
22 }
23 } else { // i == 1
24 if(x[a-1] == 0){
25 if(n-a+1 < k1) return false;
26 } else {
27 if(sol>=k2) return false;
28 }
29 }
30
31 return true;
32 }
33 }
34
35 void solution(){/*
36 int sol = 0;
37 for(int i=n; i>0; i--){
38 if(x[i] == 1) sol++;
39 else break;
40 if(sol!=0 && sol<k1) return; */
41
42 for(int i=1; i<=n; i++)
43 cout << x[i];
44
45 //if(cnt == du) cnt = 0;
46 }
47
48
49 void TRY(int a){
50 for(int i=0; i<=1; i++){
51 if(check(a,i)){
52 x[a] = i; // ngay thu a lam viec hoac khong lam viec
53 int pre = sol;
54 if(i == 1){
55 if(x[a-1] == 1) sol++;
56 else sol = 1;
57 } else {
58 sol = 0;
59 }
60
61 if(a==n) solution();
62 else TRY(a+1);
63
64 sol = pre;
65 }
66 }
67 }
68
69 int main(){
70 inputData();
71
72 Compile Log Debug Find Results

```

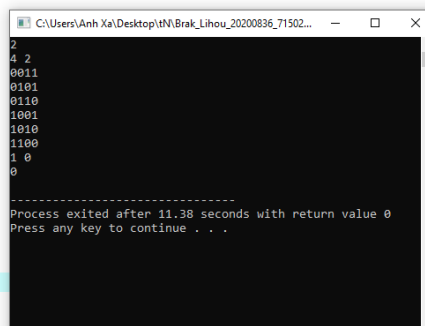
Bài tập
10:
Khoảng

cách Hamming

```

bai1.cpp bai2.cpp bai3.cpp bai4.cpp bai5.cpp bai6.cpp bai7.cpp bai9.cpp bai10.cpp
1 #include<bits/stdc++.h>
2 using namespace std;
3 const int MAX = 20;
4 int N, H;
5 int x[MAX];
6 int S[MAX]; // Xau mac dinh de so sanh
7 int cnt = 0;
8
9 void input(){
10 cin >> N >> H;
11 for(int i=0; i<N; i++){
12 S[i] = 0;
13 for(int i=0; i<N; i++){
14 x[i] = 0;
15 }
16
17 // Kiem tra 2 co khoang cach Hamming la bnh
18 int checkHamming(int str1[], int str2[]){
19 int lens = N;
20 int cnt = 0;
21 for(int i=0; i<lens; i++){
22 if(str1[i] != str2[i]) cnt++;
23 }
24 return cnt;
25 }
26
27 Compile Log Debug Find Results

```



```

22         if(str1[i] != str2[i]) cnt++;
23     }
24     return cnt;
25 }
26
27 bool check(int a, int i){
28     return true;
29 }
30
31 void solution(){
32     if(checkHamming(x,S) == H){
33         for(int i=0; i<N; i++){
34             cout << x[i];
35             cout << endl;
36         }
37     }
38 }
39 void TRY(int a){
40     for(int i=0; i<=1; i++){
41         x[a] = i;
42         if(a == N-1) solution();
43         else
44             TRY(a+1);
45     }
46 }

```

```

33     for(int i=0; i<N; i++){
34         cout << x[i];
35         cout << endl;
36     }
37 }
38
39 void TRY(int a){
40     for(int i=0; i<=1; i++){
41         x[a] = i;
42         if(a == N-1) solution();
43         else
44             TRY(a+1);
45     }
46 }
47
48 int main(){
49     int T;
50     cin >> T;
51     while(T > 0){
52         input();
53         TRY(0);
54         T--;
55     }
56 }

```

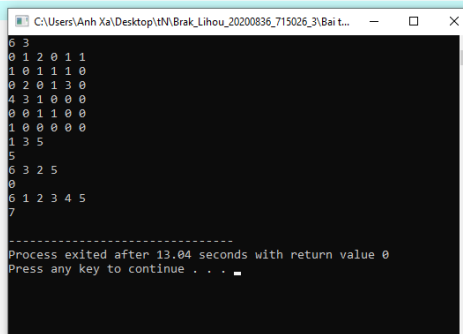
Bài tập
11: Lịch
trình

chụp ảnh

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 const int MAX = 10000;
4 int n, r;
5 int price[MAX][MAX];
6 int x[MAX];
7 bool visited[MAX];
8 vector<int> vt;
9 int min_price;
10 int sum_price;
11 int start, destination, numberOfPoint;
12
13 void input(){
14     cin >> n >> r;
15     for(int i=0; i<n; i++){
16         for(int j=0; j<n; j++){
17             cin >> price[i][j];
18         }
19     }
20 }
21 bool check(int a, int i){
22     if(visited[vt[i]]) return false;
23     if(price[x[a-1]][vt[i]] == 0) return false;
24     return true;
25 }

```



```

25 }
26
27 void solution(){
28     if(price[x[numberOfPoint-2]][destination] == 0) return;
29     min_price = min(min_price, sum_price + price[x[numberOfPoint-2]][destination]);
30 }
31
32 void TRY(int a){
33     for(int i=1; i<numberOfPoint-1; i++){
34         if(check(a, i)){
35             visited[v[i]] = true;
36             sum_price += price[x[a-1]][v[i]];
37
38             x[a] = v[i];
39             if(a == numberOfPoint-2) solution();
40             else TRY(a+1);
41
42             visited[v[i]] = false;
43             sum_price -= price[x[a-1]][v[i]];
44         }
45     }
46 }
47
48 int main(){
49     string str;
50
51     while(r > 0){
52         min_price = INT_MAX;
53         sum_price = 0;
54
55         getline(cin, str);
56
57         // Tach str thanh cac so va ghi vao vector vt
58         /*
59         int pre = 0;
60         for(int i=0; i<str.length(); i++){
61             if(str[i] == ' '){
62                 string tmp = str.substr(pre,i);
63                 pre = i + 1;
64                 stringstream convert(tmp);
65                 int tmp_int = 0;
66                 convert >> tmp_int;
67                 vt.push_back(tmp_int - 1);
68             }
69         }
70
71         string tmp = str.substr(pre,str.length());
72         stringstream convert(tmp);
73         int tmp_int = 0;
74         convert >> tmp_int;
75         vt.push_back(tmp_int - 1);
76         */
77         while (!str.empty()){
78             stringstream convert(str.substr(0, str.find(" ")));
79             int tmp = 0;
80             convert >> tmp;
81             vt.push_back(tmp - 1);
82
83             if (str.find(" ") > str.size()){
84                 break;
85             } else {
86                 str.erase(0, str.find(" ") + 1); // Update string
87             }
88         }
89
90         // Bat dau khai tao cac du lieu can thiet truuoc khi quay lui
91         start = vt[0]; // diem bat dau dau

```

```

88     }
89
90     // Bat dau khoi tao cac du lieu can thiet truoc khi quay lui
91     start = vt[0]; // diem bat dau dau
92     destination = vt[vt.size()-1]; // diem dich
93     numberOfPoint = vt.size(); // so diem phai di qua
94     x[0] = start; x[numberOfPoint-1] = destination;
95     for(int i=0; i<n; i++){
96         visited[i] = false;
97     }
98
99     TRY(1);
100
101     // In ra ket qua
102     if(min_price == INT_MAX) cout << "0" << endl;
103     else cout << min_price << endl;
104
105     // Xoa vector va chuyen sang khach tiep theo
106     vt.erase(vt.begin(), vt.end());
107     r--;
108 }
109
110
111

```

Bài tập 12: Đếm đường đi

```

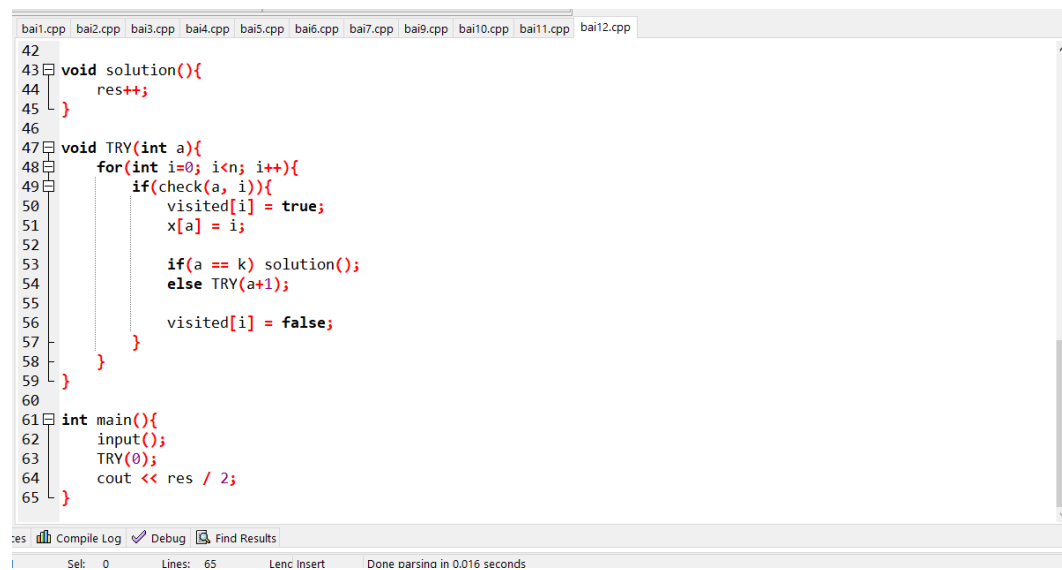
1  #include<bits/stdc++.h>
2  using namespace std;
3  const int MAX = 100;
4  int n, k;
5  int m;
6  vector<vector<int> > vt; // chuyen tu luu tru cac canh sang
7  int x[MAX];
8  bool visited[MAX];
9  int res;
10
11 void input(){
12     cin >> n >> k;
13     cin >> m;
14     vt.resize(n+1);
15     for(int i=0; i<m; i++){
16         int tmp1, tmp2;
17         cin >> tmp1 >> tmp2;
18     }
19     vt[tmp1-1].push_back(tmp2-1); // dinh tmp2 ke voi dinh tmp1
20
21 }
22
23 res = 0;
24
25 }
26
27 }
28
29 bool check(int a, int i){
30     if(a == 0) return true;
31     if(visited[i]) return false;
32
33     int index = 0;
34     for(int j=0; j<vt[a-1].size(); j++){
35         if(i == vt[a-1][j]) index++;
36     }
37     if(index == 0) return false;
38
39     return true;
40 }
41
42
43 void solution(){
44     res++;
45 }
46
47 void TRY(int a){
48     for(int i=0; i<n; i++){
49         if(check(a, i)){

```

```

C:\Users\Anh Xa\Desktop\TN_Brak_Lihou_20200836_715026_...
4 3
5
1 2
1 3
1 4
2 3
3 4
6
-----
Process exited after 19.4 seconds with return value 0
Press any key to continue . . .

```



```
42
43 void solution(){
44     res++;
45 }
46
47 void TRY(int a){
48     for(int i=0; i<n; i++){
49         if(check(a, i)){
50             visited[i] = true;
51             x[a] = i;
52
53             if(a == k) solution();
54             else TRY(a+1);
55
56             visited[i] = false;
57         }
58     }
59 }
60
61 int main(){
62     input();
63     TRY(0);
64     cout << res / 2;
65 }
```

tes | Compile Log | Debug | Find Results

Sel: 0 | Lines: 65 | Lenc Insert | Done parsing in 0.016 seconds