Contents

В	ài thực hành số 3 – Tuần 362
	Bài tập 3.1. Dãy Lucas được định nghĩa bởi Ln=Ln-1+Ln-2 và bắt đầu bởi L0=2, L1=1. Viết hàm tính số Lucas thứ n
	Bài tập 3.2 . 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)4
	Bài tập 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 ci,j. Hãy tìm tổng chi phí nhỏ nhất có thể
	Bài tập 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
	Bài tập 3.5. Tính Ckn
	Bài tập 3.6. Tính ước chung lớn nhất của hai số cho trước
	Bài tập 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
	Bài tập 3.8. 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
	Bài tập 3.9. Một y tá cần lập lịch làm việc trong N ngày, mỗi ngày chỉ có thể là làm việc hay nghỉ ngơi. Một lịch làm việc là tốt nếu không có hai ngày nghỉ nào liên tiếp và mọi chuỗi ngày tối đại làm việc liên tiếp đều có số ngày thuộc đoạn [K1,K2]. Hãy liệt kê tất cả các cách lập lịch tốt, với mỗi lịch in ra trên một dòng một xâu nhị phân độ dài n với bit 0/1 tương ứng là nghỉ/làm việc. Các xâu phải được in ra theo thứ tự từ điển
	Bài tập 3.10. 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
	Bài tập 3.11. 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.

Bài thực hành số 3 – Tuần 36

Bài tập 3.1. Dãy Lucas được định nghĩa bởi Ln=Ln-1+Ln-2 và bắt đầu bởi L0=2, L1=1. Viết hàm tính số Lucas thứ n.

```
Đáp án: (penalty regime: 10, 20, ... %)
      // Nguyen Van Duy - 20215334
   2 v /*
   3
      Bài 3.1. Dãy Lucas được định nghĩa bởi Ln=Ln-1+Ln-2
      và bắt đầu bởi L0=2, L1=1.
      Viết hàm tính số Lucas thứ n.
   5
      */
   6
      #include <iostream>
   7
   8
      using namespace std;
   9
  10
  11 | int lucas(int n) {
  12 v
  13
           # YOUR CODE HERE #
  14
          if (n == 0) return 2; // lucas(0) = 2
  15
          if (n == 1) return 1; // lucas(1) = 1
  16
  17
  18
          // tinh lucas bang de quy
  19
          return lucas(n-1) + lucas(n-2);
  20
  21
      // Nguyen Van Duy - 20215334
              Chấm điểm
 Chấm thử
```

	Test	Kết quả đúng	Kết quả chương trình	
~	cout << lucas(5);	11	11	~
~	cout << lucas(10);	123	123	~
~	cout << lucas(30);	1860498	1860498	~

// Nguyen Van Duy - 20215334 /*

Bài 3.1. Dãy Lucas được định nghĩa bởi Ln=Ln-1+Ln-2 và bắt đầu bởi L0=2, L1=1.

```
Viết hàm tính số Lucas thứ n.
*/
#include <iostream>
using namespace std;
int lucas(int n) {
  /**********
  # YOUR CODE HERE #
  ********
  if (n == 0) return 2; // lucas(0) = 2
  if (n == 1) return 1; // lucas(1) = 1
  // tinh lucas bang de quy
  return lucas(n-1) + lucas(n-2);
}
// Nguyen Van Duy - 20215334
```

Bài tập 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 (0, 1) được xem là đã đi qua)

```
// Nguyen Van Duy - 20215334
 2 v
    Bài 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 chunky của mã sao cho mỗi ô trên bàn cờ đều
 5
    được đi qua đúng 1 lần (ô (1, 1) được xem là đã đi qua)
 6
 7
    #include <iostream>
 8
   using namespace std;
9
10
   int n;
11
   int X[100], Y[100]; //# Lưu toa độ các bước di chuyển của quân mã
12
   int mark[100][100]; //# Đánh dấu vi trí các ô mà quân mã đã di chuyển qua
13
14
    //# Mảng hx, hy mô tả 8 vi trí quân mã có thể di chuyển kể từ vi trí hiện tại
15
    const int hx[] = {1, 1, 2, 2, -1, -1, -2, -2};
    const int hy[] = \{2, -2, 1, -1, 2, -2, 1, -1\};
16
17
    //# In ra dãy các di chuyển tìm được
18
19 +
    void print_sol(){
20
        for (int j = 1; j <= n * n; ++j)
            printf("(%d %d)\n", X[j], Y[j]);
21
22
        exit(0);
```

```
23 }
24
    //# Thuật toán quay lui
25
26 void TRY(int k){
27 v
        for(int i = 0; i < 8; i++){
28
            int xx = X[k-1] + hx[i];
29
            int yy = Y[k-1] + hy[i];
30 +
31
            # YOUR CODE HERE #
            *************/
32
33
34 +
            if (0 < xx && xx <= n && 0 < yy && yy <= n && mark[xx][yy] != 1) {
                // neu (xx yy) van nam trong o co
35
36
                // va
37
                // vi tri do chua duoc di
38
                X[k] = xx; // luu toa do x cua quan ma
                Y[k] = yy; // luu toa do y cua quan ma
39
40
                mark[xx][yy] = 1; // danh dau o co da duoc di
41 1
            } else {
                // backtrack
42
```

```
43
                 continue;
44
45
             if (k == n*n) {
46 v
47
                 // ghi nhan ket qua
48
                 print_sol();
49 *
             } else {
   // tim buoc di tiep theo
50
51
                 TRY(k+1);
52
53
54
             // bo danh dau
55
             mark[xx][yy] = 0;
56
57
58
59 v int main(){
60
         cin >> n;
         mark[1][1] = 1;
61
62
         X[1] = Y[1] = 1;
63
          TRY(2);
  64
          return 0;
  65
  66 // Nguyen Van Duy - 20215334
```

	Dữ liệu đầu vào	Kết quả đúng	Kết quả chương trình	
~	5	(1 1)	(1 1)	~
		(2 3)	(2 3)	
		(3 5)	(3 5)	
		(5 4)	(5 4)	
		(4 2)	(4 2)	
		(2 1)	(2 1)	
		(3 3)	(3 3)	
		(1 4)	(1 4)	
		(2 2)	(2 2)	
		(4 1)	(4 1)	
		(5 3)	(5 3)	
		(4 5)	(4 5)	
		(2 4)	(2 4)	
		(1 2)	(1 2)	
		(3 1)	(3 1)	
		(5 2)	(5 2)	
		(4 4)	(4 4)	
		(2 5)	(2 5)	
		(1 3)	(1 3)	
		(3 2)	(3 2)	
		(5 1)	(5 1)	
		(4 3)	(4 3)	
		(5 5)	(5 5)	
		(3 4)	(3 4)	
		(1 5)	(1 5)	
~	6	(1 1)	(1 1)	~
		(2 3)	(2 3)	
		(3 5)	(3 5)	

	Nguyễn Văn Duy – 2021533
(3 3)	(3-3)
(4 3)	(4 3)
(5 5)	(5 5)
(6 3)	(6 3)
(5 1)	(5 1)
(3 2)	(3 2)
(5 3)	(5 3)
(6 1)	(6 1)
(4 2)	(4 2)
(2 1)	(2 1)
(1 3)	(1 3)
(2 5)	(2 5)
(4 6)	(4 6)
(6 5)	(6 5)
(4 4)	(4 4)
(5 6)	(5 6)
(6 4)	(6 4)
(5 2)	(5 2)
(3 1)	(3 1)
(1 2)	(1 2)
(3 3)	(3 3)
(4 5)	(4 5)
(6 6)	(6 6)
(5 4)	(5 4)
(6 2)	(6 2)
(4 1)	(4 1)
(2 2)	(2 2)
(1 4)	(1 4)
(2 6)	(2 6)
(3 4)	(3 4)
(1 5)	(1 5)
(3 6)	(3 6)
(2 4)	(2 4)
(1 6)	(1 6)

			Nguyễn Văn Duy – 2	.0213
~	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)	
		(2 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)	(17)	
		(2 //	(- /)	

// Nguyen Van Duy - 20215334

/*

```
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 chunky 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)
*/
#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 i = 1; i \le n * n; ++i)
     printf("(%d %d)\n", X[j], Y[j]);
  exit(0);
}
//# Thuật toán quay lui
void TRY(int k){
  for(int i = 0; i < 8; i++){
     int xx = X[k-1] + hx[i];
     int yy = Y[k-1] + hy[i];
```

/**********

}

```
# YOUR CODE HERE #
**************/
if (0 < xx && xx <= n && 0 < yy && yy <= n && mark[xx][yy] != 1) {
  // neu (xx yy) van nam trong o co
  // va
  // vi tri do chua duoc di
  X[k] = xx; // luu toa do x cua quan ma
  Y[k] = yy; // luu toa do y cua quan ma
  mark[xx][yy] = 1; // danh dau o co da duoc di
} else {
  // backtrack
  continue;
}
if (k == n*n) {
  // ghi nhan ket qua
  print_sol();
} else {
  // tim buoc di tiep theo
  TRY(k+1);
}
// bo danh dau
mark[xx][yy] = 0;
```

```
int main(){
    cin >> n;
    mark[1][1] = 1;
    X[1] = Y[1] = 1;
    TRY(2);
    return 0;
}
// Nguyen Van Duy - 20215334
```

Bài tập 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 ci,j. Hãy tìm tổng chi phí nhỏ nhất có thể

```
1 // Nguyen Van Duy - 20215334
3
   Bài 3.3. Một người xuất phát tại thành phố 1, muốn đi thăm tất cả
 4 các thành phố khác, mỗi thành phố đúng 1 lần và quay về 1.
   Chi phí để đi từ thành phố i sang thành phố j là ci,j.
 6
    Hãy tìm tổng chi phí nhỏ nhất có thể
   #include <bits/stdc++.h>
   using namespace std;
#define MAX 100
9
10
11
12 int n, c[MAX][MAX]; //# số thành phố và ma trận chi phí
13
   int cmin = INT_MAX; //# chi phí đi lại nhỏ nhất giữa hai thành phố khác nhau
int best = INT_MAX; //# tổng chi phí nhỏ nhất cần tìm, ban đầu đặt bằng giá trị vô cùng lớn INT_MAX = 2^31-1
15 int curr; //# tổng chi phí tới thời điểm hiện tại
int mark[MAX]; //# đánh dấu những thành phố đã đi
tx[MAX]; //# lưu giữ các thành phố đã đi
18
19 //# Đọc dữ liệu vào
20 void input(){
21
        cin >> n;
        for (int i = 1; i <= n; ++i)
23 1
               for (int j = 1; j \le n; ++j){
 24
                   cin >> c[i][j];
 25
                   if (c[i][j] > 0) cmin = min(cmin, c[i][j]);
 26
               }
 27
 28
 29
     //# Thuật toán quay lui
 30 - void TRY(int k){
          for(int i = 2; i <= n; i++){
 31 v
 32 v
 33
               # YOUR CODE HERE #
               **************/
 34
 35 v
               if(mark[i] == 0 \&\& c[x[k-1]][i] != 0){
 36
                   x[k] = i; // luu thanh pho da di
 37
                   mark[i] = 1; // danh dau thanh pho i da duoc di qua
 38
                   curr += c[x[k-1]][i]; // cap nhat chi phi khi di qua thanh pho i
 39
                   if(k == n){
 40
                        // neu da di het n thanh pho
 41
                        if(c[x[k]][1] != 0){
 42
```

```
43
                         // thanh pho cuoi cung di ve lai duoc thanh pho ban dau
44
                         best = min(best, curr + c[x[k]][1]); // cap nhat ket qua
45
                    }
46
                 }else{
47
                     int g = curr + (n-k+1)*cmin; // chi phi toi thieu neu tiep tuc di
48 1
                     if(g < best){</pre>
49
                         // neu co the nho hon chi phi tot nhat cua hien tai
50
                         // thi tiep tuc
51
                         TRY(k+1);
52
53
54
55
                 mark[i] = 0; // bo danh dau
56
                 curr -= c[x[k-1]][i]; // xoa chi phi da di
57
58
59
60
61 +
    int main() {
62
        input();
```

	Dữ liệu đầu vào	Kết quả đúng	Kết quả chương trình	
~	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	*

Hoàn thành được tất cả các bô mẫu 🗸

```
// Nguyen Van Duy - 20215334
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 ci,j.
Hãy tìm tổng chi phí nhỏ nhất có thể
*/
#include <bits/stdc++.h>
using namespace std;
#define MAX 100
int n, c[MAX][MAX]; //# số thành phố và ma trận chi phí
int cmin = INT MAX; //# chi phí đi lại nhỏ nhất giữa hai thành phố khác nhau
int best = INT MAX; //# tổng chi phí nhỏ nhất cần tìm, ban đầu đặt bằng giá tri vô cùng
lớn INT_MAX = 2^31-1
int curr; //# tổng chi phí tới thời điểm hiện tại
int mark[MAX]; //# đánh dấu những thành phố đã đi
int x[MAX]; //# lưu giữ các thành phố đã đi
//# Đọc dữ liệu vào
void input(){
  cin >> n;
  for (int i = 1; i \le n; ++i)
     for (int j = 1; j \le n; ++j)
       cin >> c[i][j];
       if (c[i][j] > 0) cmin = min(cmin, c[i][j]);
     }
}
```

```
//# Thuật toán quay lui
void TRY(int k){
  for(int i = 2; i \le n; i++){
     /**********
     # YOUR CODE HERE #
     **************
     if(mark[i] == 0 \&\& c[x[k-1]][i] != 0){
       x[k] = i; // luu thanh pho da di
       mark[i] = 1; // danh dau thanh pho i da duoc di qua
       curr += c[x[k-1]][i]; // cap nhat chi phi khi di qua thanh pho i
       if(k == n)
         // neu da di het n thanh pho
         if(c[x[k]][1] != 0){
            // va
            // thanh pho cuoi cung di ve lai duoc thanh pho ban dau
            best = min(best, curr + c[x[k]][1]); // cap nhat ket qua
          }
       }else{
          int g = curr + (n-k+1)*cmin; // chi phi toi thieu neu tiep tuc di
         if(g < best)
            // neu co the nho hon chi phi tot nhat cua hien tai
            // thi tiep tuc
            TRY(k+1);
          }
```

Bài tập 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

```
1 // Nguyen Van Duy - 20215334
2 * /*
3 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
7
   #include <bits/stdc++.h>
8
9
   using namespace std;
   int a[1000], n;
10
   int mem[1000]; //# mảng ghi nhớ lời giải các bài toán con đã được giải
11
12
13 🔻
   void init(){
        memset(mem, -1, sizeof(mem));
14
15
16
17
    //# Quy hoạch động,
18 //# Hàm lis(i) trả về độ dài dãy con tăng dài nhất kết thúc bởi a[i]
19 v int lis(int i) {
20 +
        # YOUR CODE HERE #
21
22
        **************
23 v
        if (mem[i] != -1) {
24
            // tra ve ket qua da duoc giai
25
            return mem[i];
26
        }
27
28
        int res = 1; //# luu loi giai cua bai toan
29 +
        for(int j = 0; j < i; j++){
30
             // neu day con truoc do < a[i]
31 v
            if (a[j] < a[i]){
32
                // res = day con dai nhat phia truoc + 1
33
                res = max(res, lis(j) + 1);
34
            }
35
36
37
        mem[i] = res; // luu loi giai cua bai toan con da duoc giai
38
        return res;
39
40
    //# Truy vet loi giai
42 void trace(int i){
```

```
43 *
         for(int j = 0; j < i; j++){
44 *
             if (a[j] < a[i] && mem[i] == 1 + mem[j]){</pre>
45
                 trace(j);
46
                 break;
47
             }
48
49
         cout << a[i] << " ";
50
51
52 v int main(){
53
         init();
         cin >> n;
54
55
         for(int i = 0; i < n; i++) cin >> a[i];
56
         int res = 1, pos = 0;
57 ₹
         for(int i = 1; i < n; i++){
58 +
             if (res < lis(i)){
59
                 res = lis(i);
60
                 pos = i;
61
             }
62
         }
63
        cout << res << endl;
64
        trace(pos);
65
        return 0;
66
67 // Nguyen Van Duy - 20215334
```

	Dữ liệu đầu vào	Kết quả đúng	Kết quả chương trình	
*	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	~

```
// Nguyen Van Duy - 20215334
```

/*

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

*/

#include <bits/stdc++.h>

```
using namespace std;
int a[1000], n;
int mem[1000]; //# mång 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]
int lis(int i) {
  /******
  # YOUR CODE HERE #
  ********
  if (mem[i] != -1) {
    // tra ve ket qua da duoc giai
    return mem[i];
  }
  int res = 1; //# luu loi giai cua bai toan
  for(int j = 0; j < i; j++){
    // neu day con truoc do < a[i]
    if (a[j] < a[i]){
       // res = day con dai nhat phia truoc + 1
       res = max(res, lis(j) + 1);
  }
```

```
mem[i] = res; // luu loi giai cua bai toan con da duoc giai
  return res;
}
//# Truy vet loi giai
void trace(int i){
  for(int j = 0; j < i; j++){
     if (a[j] < a[i] && mem[i] == 1 + mem[j]){
       trace(j);
       break;
     }
  cout << a[i] << " ";
}
int main(){
  init();
  cin >> n;
  for(int i = 0; i < n; i++) cin >> a[i];
  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 Van Duy - 20215334</pre>
```

Bài tập 3.5. Tính Ckn

```
// Nguyen Van Duy - 20215334
 2 v /*
 3
    Bài 3.5. Tính Ckn
 4
 5
    #include <iostream>
 6
    using namespace std;
 8 | int binom(int n, int k) {
 9
        if (k > n) return 0;
        if (k == 0) return 1;
10
11
        return binom(n-1, k) + binom(n-1, k-1);
12
    }
13
14 | int binom2(int n, int k){
15
        if (k > n) return 0;
        //# Khử đệ quy
16
        /**********
17 ,
        # YOUR CODE HERE #
18
19
        ****************
20
        long long int giaiThua_n = 1; // = n!
21
        long long int giaiThua_nk = 1; // = (n-k)!
        long long int giaiThua_k = 1; // = k!
22
23
24
        for (long long int i = 1; i \le n; ++i) {
25
            giaiThua_n *= i;
26
27 1
            if (i <= n-k) {
                 giaiThua_nk *= i;
28
29
30
31 1
            if (i \leftarrow k) {
32
                 giaiThua_k *= i;
33
34
35
36
        // binom(n, k) = n! / ((n-k)! * k!)
37
        return giaiThua_n / (giaiThua_nk * giaiThua_k);
38
    }
39
40 v int main() {
41
        int m;
42
        cin >> m;
43 +
        for (int n = 1; n <= m; ++n){
44
            for (int k = 0; k \le n; ++k)
45
                 printf("%d ", binom(n, k));
46
            printf("\n");
47
48 1
        for (int n = 1; n \le m; ++n){
49
            for (int k = 0; k \le n; ++k)
                printf("%d ", binom2(n, k));
50
51
            printf("\n");
52
53
        return 0;
54
55 // Nguyen Van Duy - 20215334
```

	0.7-						
	Dữ liệu đầu vào	Kết quả đúng	Kết quả chương trình				
~	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 1 1 1 1 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 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 10 45 120 210 252 210 1 1 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 10 45 120 210 252 210 1				

```
20
     1 1
                                                                                                               1 1
     1 2 1
                                                                                                               1 2 1
     1331
                                                                                                               1331
     14641
                                                                                                               14641
     1 5 10 10 5 1
                                                                                                               1 5 10 10 5 1
     1 6 15 20 15 6 1
                                                                                                               1 6 15 20 15 6 1
     1 7 21 35 35 21 7 1
                                                                                                               1 7 21 35 35 21 7 1
     1 8 28 56 70 56 28 8 1
                                                                                                               1 8 28 56 70 56 28 8 1
     1 9 36 84 126 126 84 36 9 1
                                                                                                               1 9 36 84 126 126 84 36 9
     1 10 45 120 210 252 210 120 45 10 1
                                                                                                               1 10 45 120 210 252 210 1
     1 11 55 165 330 462 462 330 165 55 11 1
                                                                                                               1 11 55 165 330 462 462 3
     1 12 66 220 495 792 924 792 495 220 66 12 1
                                                                                                               1 12 66 220 495 792 924 7
     1 13 78 286 715 1287 1716 1716 1287 715 286 78 13 1
                                                                                                               1 13 78 286 715 1287 1716
     1 14 91 364 1001 2002 3003 3432 3003 2002 1001 364 91 14 1
                                                                                                               1 14 91 364 1001 2002 300
     1 15 105 455 1365 3003 5005 6435 6435 5005 3003 1365 455 105 15 1
                                                                                                               1 15 105 455 1365 3003 50
     1 16 120 560 1820 4368 8008 11440 12870 11440 8008 4368 1820 560 120 16 1
                                                                                                               1 16 120 560 1820 4368 80
     1 17 136 680 2380 6188 12376 19448 24310 24310 19448 12376 6188 2380 680 136 17 1
                                                                                                               1 17 136 680 2380 6188 12
     1 18 153 816 3060 8568 18564 31824 43758 48620 43758 31824 18564 8568 3060 816 153 18 1
                                                                                                               1 18 153 816 3060 8568 18
     1 19 171 969 3876 11628 27132 50388 75582 92378 92378 75582 50388 27132 11628 3876 969 171 19 1
                                                                                                               1 19 171 969 3876 11628 2
     1 20 190 1140 4845 15504 38760 77520 125970 167960 184756 167960 125970 77520 38760 15504 4845 1140 190 20 1 1 20 190 1140 4845 15504
```

```
1 1
1 2 1
                                                                                                            1 2 1
1331
                                                                                                           1331
14641
                                                                                                           14641
1 5 10 10 5 1
                                                                                                           1 5 10 10 5 1
1 6 15 20 15 6 1
                                                                                                           1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
                                                                                                           1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
                                                                                                           1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
                                                                                                           1 9 36 84 126 126 84 36 9
1 10 45 120 210 252 210 120 45 10 1
                                                                                                           1 10 45 120 210 252 210 1
1 11 55 165 330 462 462 330 165 55 11 1
                                                                                                           1 11 55 165 330 462 462 3
1 12 66 220 495 792 924 792 495 220 66 12 1
                                                                                                           1 12 66 220 495 792 924 7
1 13 78 286 715 1287 1716 1716 1287 715 286 78 13 1
                                                                                                            1 13 78 286 715 1287 1716
1 14 91 364 1001 2002 3003 3432 3003 2002 1001 364 91 14 1
                                                                                                           1 14 91 364 1001 2002 300
1 15 105 455 1365 3003 5005 6435 6435 5005 3003 1365 455 105 15 1
                                                                                                           1 15 105 455 1365 3003 50
1 16 120 560 1820 4368 8008 11440 12870 11440 8008 4368 1820 560 120 16 1
                                                                                                           1 16 120 560 1820 4368 80
1 17 136 680 2380 6188 12376 19448 24310 24310 19448 12376 6188 2380 680 136 17 1
                                                                                                           1 17 136 680 2380 6188 12
1 18 153 816 3060 8568 18564 31824 43758 48620 43758 31824 18564 8568 3060 816 153 18 1
                                                                                                           1 18 153 816 3060 8568 18
1 19 171 969 3876 11628 27132 50388 75582 92378 92378 75582 50388 27132 11628 3876 969 171 19 1
                                                                                                           1 19 171 969 3876 11628 2
1 20 190 1140 4845 15504 38760 77520 125970 167960 184756 167960 125970 77520 38760 15504 4845 1140 190 20 1 1 20 190 1140 4845 15504
```

Hoàn thành được tất cả các bộ mẫu 🛷

```
// Nguyen Van Duy - 20215334

/*

Bài 3.5. Tính Ckn

*/

#include <iostream>
using namespace std;

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) {
   if (k > n) return 0;
   //# Khử đệ quy
```

```
/******
  # YOUR CODE HERE #
  *************
  long long int giaiThua_n = 1; // = n!
  long long int giaiThua_nk = 1; // = (n-k)!
  long long int giaiThua_k = 1; // = k!
  for (long long int i = 1; i \le n; ++i) {
    giaiThua_n *= i;
    if (i \le n-k) {
       giaiThua_nk *= i;
     }
    if (i \le k) {
       giaiThua_k *= i;
     }
  }
  // binom(n, k) = n! / ((n-k)! * k!)
  return giaiThua_n / (giaiThua_nk * giaiThua_k);
}
int main() {
  int m;
  cin \gg m;
  for (int n = 1; n \le m; ++n){
```

Bài tập 3.6. Tính ước chung lớn nhất của hai số cho trước

```
1 // Nguyen Van Duy - 20215334
3
   Bài 3.6. Tính ước chung lớn nhất của hai số cho trước
4
   #include <iostream>
 5
 6
    using namespace std;
8 | int gcd(int a, int b){
 9
        if (b == 0) return a;
10
        return gcd(b, a % b);
11
   }
12
13 v int gcd2(int a, int b){
14
15
        //# Khử đệ quy
        /***********
16 +
        # YOUR CODE HERE #
17
        ************/
18
19
        // lap den khi a = 0 hoac b = 0
20 +
        while(a && b) {
21
            // so lon = so lon % so be
22 *
            if (a > b) {
23
                 a = a \% b;
            } else {
 24 •
 25
                 b = b \% a;
             }
 26
 27
 28
         // tra ve so khac 0 trong hai so a va b
 29
 30
         return a ? a : b;
 31
 32
 33 v int main() {
 34
         int a, b;
         cin \gg a \gg b;
 35
         cout << gcd(a, b) << end1 << gcd2(a, b);
 36
 37
         return 0;
 38
 39 // Nguyen Van Duy - 20215334
```

0 35			
0 55	5 5	5 5	~
17 413	7	7 7	~
		-	7 413 7 7 7

```
// Nguyen Van Duy - 20215334
/*
```

Bài 3.6. Tính ước chung lớn nhất của hai số cho trước

```
*/
#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
  /**********
  # YOUR CODE HERE #
  *************
  // lap den khi a = 0 hoac b = 0
  while(a && b) {
    // so lon = so lon % so be
    if (a > b) {
       a = a \% b;
    } else {
       b = b \% a;
    }
  }
  // tra ve so khac 0 trong hai so a va b
  return a?a:b;
```

```
int main() {
  int a, b;
  cin >> a >> b;
  cout << gcd(a, b) << endl << gcd2(a, b);
  return 0;
}
// Nguyen Van Duy - 20215334</pre>
```

Bài tập 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

```
1 // Nguyen Van Duy - 20215334
 2 *
 3
    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
 4
 5
    #include <bits/stdc++.h>
 6
 7
    using namespace std;
8
9 * struct state{
10
       int i, j, old_L;
11
        //# constructor
12
        state(int _i = 0, int _j = 0, int _L = 0):
13
               i(_i), j(_j), old_L(_L){}
14
    };
15
16 v int main() {
17
        int n, k;
18
        cin >> n >> k;
19
        int x[n+1];
20
        stack<state> s;
21
        //# number of consecutive suffix 1
22
        int L = 0;
23
         s.push(state(1, 0));
24 v
         while (!s.empty()){
25
            state &top = s.top();
26
             //# if a new binary sequence is found
27 1
             if (top.i > n){
28
                for (int i = 1; i <= n; ++i)
29
                    cout << x[i] << " \n"[i == n];
30
                 s.pop();
31
                 continue;
32
             }
33
34
             //# Khử đệ quy
             /**********
35 v
36
             # YOUR CODE HERE #
37
38
39
             // luu cac gia tri o ngan dau tien stack
             int i = top.i, j = top.j, old_L = top.old_L;
40
41
             s.pop();
42
```

```
43
            x[i] = j; // ghi gia tri j vao vi tri thu i
44
45 v
            if (j) {
46
                // neu j = 1 thi so bit 1 o cuoi day tang them 1
47
                L = old_L + 1;
48 -
            } else {
                // neu j = 0 thi khong co bit 1 nao o cuoi day
49
50
                L = 0;
51 1
                if (old_L + 1 < k) {
52
                    // dua vao stack 1 state (ghi gia tri 1 vao vi tri thu i)
53
                    // thay the so 0 o vi tri hien tai thanh so 1
54
                    s.push(state(i, 1, old_L));
55
                }
56
57
58
            // dua vao stack 1 state (ghi gia tri 0 vao vi tri thu i+1)
59
            // vi tri tiep theo se nhan gia tri 0
60
            s.push(state(i + 1, 0, L));
61
62
        return 0;
```

63 } 64 // Nguyen Van Duy - 20215334

	Dữ liệu đầu vào	Kết quả đúng	Kết quả chương trình	
~	4 2	0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 1 1 1 0 0 0 1 1 1 0 1 0 1 0 1 1 0 1	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	>
~	5 3	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 0 0 0 1	*
~	7 3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	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 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0 1 1 0 1 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1	*

			Nguyê	n Văn	Duy –
001	0 0	1 1	0010	011	
001	0 1	0 0	0010	100	
001	0 1	0 1	0010		
001	0 1	10	0010		
001	10	0 0	0011		
001	10	0 1	0011	001	
001	10	10	0011	010	
001	10	11	0011		
010	00	00	0100		
010	00	10	0100		
010	00	11	0100		
010	01	0 0	0100		
010	01	0 1	0100		
010	0 1	1 0	0100		
010	10	0 0	0101	000	
010	10	0 1	0101	001	
010	10	10	0101	010	
010	10	1 1	0101	011	
010	1 1	0 0	0101	100	
010	1 1	0 1	0101	101	
011	0 0	0 0	0110	000	
011	0 0	0 1	0110	001	
011	0 0	10	0110	010	
011	0 0	1 1	0110	011	
011	0 1	0 0	0110	100	
011	0 1	0 1	0110	101	
0 1 1	0 1	10	0110		
100	00	0 0	1000		
100	00	01	1000		
100	0 0	10	1000		
100	00 01	11	1000		
100	01	01	1000		
100	01	10	1000	110	
100	10	0 0	1001	000	
100	10	01	1001	001	
100	1 0	1 0	1001		
100	10	1 1	1001	011	
100	1 1	0 0	1001	100	
100	1 1	0 1	1001	101	
101	0 0	0 0	1010	000	
101	0 0	0 1	1010	001	
101	0 0	10	1010		
101	0 0	1 1	1010		
101	0 1	0 0	1010		
101	0 1	0 1	1010		
101	0 1	10	1010		
101	10	00	1011		
101	10	01	1011		
101	10	10	1011		
101	1000	11	1011		
110	00	01	1100		
110	00	10	1100		
110	00	11	1100		
	_				

```
1100100 1100100
1100101 1100101
1100110 1100110
1101000 1101000
1101001 1101001
1101010 1101010
1101011 1101011
1101100 1101100
1101101 1101100
```

Hoàn thành được tất cả các bộ mẫu 🗸

```
// Nguyen Van Duy - 20215334
/*
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
*/
#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(\underline{i}), j(\underline{j}), old_L(\underline{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 \le n; ++i)
       cout << x[i] << " \n"[i == n];
    s.pop();
     continue;
  }
  //# Khử đệ quy
  /**********
  # YOUR CODE HERE #
  ********
  // luu cac gia tri o ngan dau tien stack
  int i = top.i, j = top.j, old_L = top.old_L;
  s.pop();
  x[i] = j; // ghi gia tri j vao vi tri thu i
  if (j) {
```

```
// neu j = 1 thi so bit 1 o cuoi day tang them 1
       L = old_L + 1;
     } else {
       // neu j = 0 thi khong co bit 1 nao o cuoi day
        L = 0;
       if (old_L + 1 < k) {
          // dua vao stack 1 state (ghi gia tri 1 vao vi tri thu i)
          // thay the so 0 o vi tri hien tai thanh so 1
          s.push(state(i, 1, old_L));
        }
     }
     // dua vao stack 1 state (ghi gia tri 0 vao vi tri thu i+1)
     // vi tri tiep theo se nhan gia tri 0
     s.push(state(i + 1, 0, L));
  }
  return 0;
}
// Nguyen Van Duy - 20215334
```

Bài tập 3.8. 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.

```
// Nguyen Van Duy - 20215334
1
 2
     Bài 3.8. B?n dang mu?n ki?m tra xem m?t v?t cho tru?c có dúng n?ng M nhu ngu?i ta nói hay không.
 3
 4
     Có m?t cân thang b?ng và n qu? cân. Qu? th? i n?ng mi. Hãy ch? ra m?t cách cân th?a mãn.
 5
     Quy cách in ra dã du?c tích h?p trong mã ngu?n du?i.
 6
 7
     #include <bits/stdc++.h>
 8
     using namespace std;
9 □ struct state{
10
         int i, j;
11
         state(int _i = 0, int _j = 0): i(_i), j(_j) {}
12 L };
13 ☐ int main() {
         int n, M;
15
         cin >> n >> M;
16
         int m[n+1];
17
         for (int i = 1; i <= n; ++i) cin >> m[i];
18
         int x[n+1];
19
         stack<state> s;
20
         //# sum of selected weights
21
         int sum = 0;
22
         s.push(state(1, -1));
23 🖨
         while (!s.empty()){
24
             state &top = s.top();
25 🖨
             if (top.i > n){
26 🛱
                 if (sum == M){}
27 白
                     for (int i = 1; i \le n; ++i){
28
                           if (x[i] == -1) cout << '-' << m[i];</pre>
29
                           if (x[i] == 1) cout << '+' << m[i];</pre>
30
31
                       cout << "=" << M;
32
                       exit(0);
33
34
                   s.pop();
35
                   continue;
36
37
38
              //# Kh? d? guy
39
               /***********
40
              # YOUR CODE HERE #
               **************/
41
42
43
              // luu cac gia tri o ngan dau tien stack
44
              int i = top.i, j = top.j;
45
              s.pop();
46
47
              // neu truong hop truoc do da tung su dung qua can i
48 🖃
              if (x[i] == 1) {
49
                  // huy bo trang thai truoc do
                  sum -= m[i]; // tinh lai sum
50
51
```

```
52
              x[i] = j; // ghi gia tri j vao vi tri thu i (1 hoac -1)
53
54
55 🖨
              if (j == 1) {
56
                  // su dung qua can thu i
57
                  sum += m[i];
              } else {
58
                  //j = -1
59
                  // khong su dung gua can i
60
                  s.push(state(i, 1)); // dua vao stack 1 state (su dung qua can i)
61
62
63
              // dua vao stack 1 state (khong su dung qua can i)
64
              s.push(state(i + 1, -1));
65
66
67
          cout << -1;
68
69
          return 0;
70
      // Nguyen Van Duy - 20215334
71
  1 2 3 4 5
 7-1+2+3-4+5=10
Process exited after 47.85 seconds with return value 0
Press any key to continue . . .
// Nguyen Van Duy - 20215334
```

/*

Bài 3.8. 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.

```
*/
#include <bits/stdc++.h>
using namespace std;
struct state{
  int i, j;
  state(int _i = 0, int _j = 0): i(_i), j(_j) {}
};
```

```
int main() {
  int n, M;
  cin >> n >> M;
  int m[n+1];
  for (int i = 1; i \le n; ++i) cin >> m[i];
  int x[n+1];
  stack<state> s;
  //# sum of selected weights
  int sum = 0;
  s.push(state(1, -1));
  while (!s.empty()){
     state &top = s.top();
    if (top.i > n){
       if (sum == M)
          for (int i = 1; i \le n; ++i){
            if (x[i] == -1) cout << '-' << m[i];
            if (x[i] == 1) cout << '+' << m[i];
          }
         cout << "=" << M;
          exit(0);
       s.pop();
       continue;
     }
    //# Khử đệ quy
     /*******
```

```
# YOUR CODE HERE #
********
// luu cac gia tri o ngan dau tien stack
int i = top.i, j = top.j;
s.pop();
// neu truong hop truoc do da tung su dung qua can i
if (x[i] == 1) {
  // huy bo trang thai truoc do
  sum -= m[i]; // tinh lai sum
}
x[i] = j; // ghi gia tri j vao vi tri thu i (1 hoac -1)
if (j == 1) {
  // su dung qua can thu i
  sum += m[i];
} else {
  //j = -1
  // khong su dung qua can i
  s.push(state(i, 1)); // dua vao stack 1 state (su dung qua can i)
}
// dua vao stack 1 state (khong su dung qua can i)
s.push(state(i + 1, -1));
```

}

```
cout << -1;
return 0;
}
// Nguyen Van Duy - 20215334</pre>
```

Bài tập 3.9. Một y tá cần lập lịch làm việc trong N ngày, mỗi ngày chỉ có thể là làm việc hay nghỉ ngơi. Một lịch làm việc là tốt nếu không có hai ngày nghỉ nào liên tiếp và mọi chuỗi ngày tối đại làm việc liên tiếp đều có số ngày thuộc đoạn [K1,K2]. Hãy liệt kê tất cả các cách lập lịch tốt, với mỗi lịch in ra trên một dòng một xâu nhị phân độ dài n với bit 0/1 tương ứng là nghỉ/làm việc. Các xâu phải được in ra theo thứ tự từ điển

```
// Nguyen Van Duy - 20215334
 2
 3
     Bài 3.9. M?t y tá c?n l?p l?ch làm vi?c trong N ngày,
     m?i ngày ch? có th? là làm vi?c hay ngh? ngoi.
     M?t l?ch làm vi?c là t?t n?u không có hai ngày ngh? nào
     liên ti?p và m?i chu?i ngày t?i d?i làm vi?c liên ti?p
 7
     d?u có s? ngày thu?c do?n [K1,K2].
     Hãy li?t kê t?t c? các cách l?p l?ch t?t, v?i m?i l?ch
 8
 9
     in ra trên m?t dòng m?t xâu nh? phân d? dài n v?i
10
     bit 0/1 tuong ?ng là ngh?/làm vi?c.
11
     Các xâu ph?i du?c in ra theo th? t? t? di?n
12
13
     #include <bits/stdc++.h>
14
     using namespace std;
15
16 | struct state{
17
         int i, j, old_L;
18
         //# constructor
19
         state(int _i = 0, int _j = 0, int _L = 0):
20
                  i(_i), j(_j), old_L(_L){}
21
   L };
22
23 ☐ int main() {
24
          int n, k1, k2;
25
          cin >> n >> k1 >> k2;
26
         int x[n+1];
27
          stack<state> s;
          //# number of consecutive suffix 1
28
29
         int L = 0;
30
          s.push(state(1, 0));
31 🖹
         while (!s.empty()){
32
              state &top = s.top();
33
              //# if a new binary sequence is found
34 🗐
              if (top.i > n){
35 🖨
                  if (x[n] && top.old_L >= k1 || !x[n]) {
36
                      for (int i = 1; i <= n; ++i)
37
                          cout << x[i];
38
                      cout << endl;
39
40
                  s.pop();
41
                  continue;
42
43
44
              //# Kh? d? quy
              /********
45
              # YOUR CODE HERE #
46
47
48
49
              // luu cac gia tri o ngan dau tien stack
50
              int i = top.i, j = top.j, old_L = top.old_L;
```

```
51
              s.pop();
52
              x[i] = j; // ghi trang thai lam viec ngay i
53
54
55 🗀
              if (j) {
56
                  // case: j = 1
57
                  L = old L + 1; // so ngay lam viec lien tiep tang them 1
58 🖃
                  if (L >= k1) {
59
                      // dua vao stack 1 state (nghi vao ngay tiep theo)
                      s.push(state(i + 1, 0, L));
60
61
                  } else {
62
                      // dua vao stack 1 state (di lam vao ngay tiep theo)
63
                      s.push(state(i + 1, 1, L));
64
65
              } else {
66
                  // case: j = 0
67
                  L = 0; // reset so ngay lam viec truoc do
68 🖃
                  if (old_L + 1 <= k2) {</pre>
69
                      // chuoi ngay lam viec truoc do chua toi da
70
                      // dua vao stack 1 state (di lam vao ngay hien tai)
                      s.push(state(i, 1, old_L));
71
72
73
                  // dua vao stack 1 state (di lam vao ngay tiep theo)
74
                  s.push(state(i+1, 1, L));
75
76
77
78
          return 0;
79
80
      // Nguyen Van Duy - 20215334
81
011011
110110
110111
111011
Process exited after 2.016 seconds with return value 0
Press any key to continue . .
// Nguyen Van Duy - 20215334
/*
```

Bài 3.9. Một y tá cần lập lịch làm việc trong N ngày, mỗi ngày chỉ có thể là làm việc hay nghỉ ngơi.

Một lịch làm việc là tốt nếu không có hai ngày nghỉ nào liên tiếp và mọi chuỗi ngày tối đại làm việc liên tiếp đều có số ngày thuộc đoạn [K1,K2].

```
Hãy liệt kê tất cả các cách lập lịch tốt, với mỗi lịch
in ra trên một dòng một xâu nhị phân độ dài n với
bit 0/1 tương ứng là nghỉ/làm việc.
Các xâu phải được in ra theo thứ tự từ điển
*/
#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(\underline{i}), j(\underline{j}), old_L(\underline{L})\{\}
};
int main() {
  int n, k1, k2;
  cin >> n >> k1 >> k2;
  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){
```

Nguyễn Văn Duy - 20215334 if $(x[n] \&\& top.old_L >= k1 || !x[n]) {$ for (int i = 1; $i \le n$; ++i) $cout \ll x[i];$ cout << endl; } s.pop(); continue; } //# Khử đệ quy /********** # YOUR CODE HERE # ************** // luu cac gia tri o ngan dau tien stack int i = top.i, j = top.j, old_L = top.old_L; s.pop(); x[i] = j; // ghi trang thai lam viec ngay i

```
// dua vao stack 1 state (di lam vao ngay tiep theo)
          s.push(state(i + 1, 1, L));
        }
     } else {
       // case: j = 0
       L = 0; // reset so ngay lam viec truoc do
       if (old_L + 1 \le k2) {
          // chuoi ngay lam viec truoc do chua toi da
          // dua vao stack 1 state (di lam vao ngay hien tai)
          s.push(state(i, 1, old_L));
        }
       // dua vao stack 1 state (di lam vao ngay tiep theo)
       s.push(state(i+1, 1, L));
     }
  }
  return 0;
}
// Nguyen Van Duy - 20215334
```

Bài tập 3.10. 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 tư từ điển

```
// Nguyen Van Duy - 20215334

// Bài 3.10. Khoảng cách Hamming giữa hai xâu cùng độ dài là số

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

**/

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;
```

```
// in ra cac gia tri trong vector<int>
inline void print_sol(const vector<int>& v) {
    for (int i : v) {
        cout << i;
    }
    cout << endl;
}</pre>
```

```
// Nguyen Van Duy - 20215334

/*

Bài 3.10. 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

*/

#include <iostream>
```

```
#include <vector>
#include <algorithm>
using namespace std;
inline void print_sol(const vector<int>& v) {
  for (int i : v) {
     cout << i;
  }
  cout << endl;
}
int main() {
  int t;
  cin >> t;
  while (t--) {
     int n, h;
     cin >> n >> h;
     vector<int> v(n);
     for (int i = 0; i < n; ++i) {
      v[i] = i < h ? 1 : 0;
     }
```

```
vector<int> v_end(n);

v_end = v;

do {
    next_permutation(v.begin(), v.end());
    print_sol(v);
} while (v != v_end);
}

return 0;

// Nguyen Van Duy - 20215334
```

Bài tập 3.11. 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.

```
// Nguyen Van Duy - 20215334

// *

Bài 3.11. 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.
boàn 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 đứng 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.

/*/
#include <iostream>
#include <stack>
#include <stack>
#include <cstring>

using namespace std;
#define MAX 100
```

```
// o thu tu i, di den dia diem j

// hoac

// xoa dia diem da di o thu tu i, neu j = 0

dinline void write(int i, int j) {

mark[j] = 1; // danh dau da di qua dia diem j

if (i == 1 && j) {

// i la dia diem dau tien

x[i] = j; // luu gia da di dia diem i

return;

}

// neu i khong phai dia diem dau tien

if (j) {

x[i] = j; // luu gia da di dia diem i

curr += c[x[i-1]][x[i]]; // cap nhat khoang cach

return;

} else {

// j = 0

// xoa dia diem da di o thu tu i

curr -= c[x[i-1]][x[i]]; // xoa khoang cach da di

mark[x[i]] = 0; // danh dau dia diem chua di

x[i] = j;

return;

}

// The standard of the standard
```

```
stack<state> s;
memset( Dst mark, Val 0, Size sizeof(mark)); // gán giá trị 0 (chưa đi qua) cho tất cả địa điểm
int best = INT_MAX; //# tổng khoảng cách nhỏ nhất cần tìm
// ban đầu đặt bằng giá trị vô cùng lớn INT_MAX = 2^31-1
curr = 0; // khoảng cách thời điểm hiện tại ban đầu đặt bằng 0
bool not_found = true; // trạng thái đã tìm được đường đi hay chưa, ban đầu là chưa
```

```
C:\Users\Admin\CLionProjects\test\cmake-build-debug\test.exe
6 3
0 1 2 0 1 1
1 0 1 1 1 0
0 2 0 1 3 0
4 3 1 0 0 0
0 0 1 1 0 0
1 0 0 0 0 0
1 3 5
5
6 3 2 5
0
6 1 2 3 4 5
7

Process finished with exit code 0
```

```
// Nguyen Van Duy - 20215334
```

/*

Bài 3.11. 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.

*/

#include <iostream>

#include <vector>

#include <sstream>

#include <stack>

```
#include <cstring>
using namespace std;
#define MAX 100
int n, c[MAX][MAX]; //# số thành phố và ma trận chi phí
int curr; //# tổng chi phí tới thời điểm hiện tại
int mark[MAX]; //# đánh dấu những thành phố đã đi
int x[MAX]; //# lưu giữ các thành phố đã đi
int r;
inline void show_curr() {
  cout << "curr:" << curr << endl;
}
//# Đọc dữ liệu vào
inline void input(){
  cin >> n >> r;
  for (int i = 1; i <= n; ++i)
    for (int j = 1; j <= n; ++j)
       cin >> c[i][j];
```

}

```
struct state {
  int i;
  int j;
  state(int i, int j) : i(i), j(j) {}
};
inline void show_x(size_t n) {
  for (int i = 1; i < n; ++i) {
     cout << x[i] << " ";
  }
  cout << " <- x[]" << endl;
}
inline void write(int i, int j) {
  mark[j] = 1;
  if (i == 1) {
     x[i] = j;
     return;
  }
  if (j) {
```

```
x[i] = j;
     curr += c[x[i-1]][x[i]];
     return;
  } else {
     curr -= c[x[i-1]][x[i]];
     mark[x[i]] = 0;
     x[i] = j;
     return;
  }
}
int main() {
  input();
  cin.ignore();
  while (r--) {
     string line;
     getline(cin, line);
     stringstream ss(line);
     vector<int> v;
     v.push_back(0);
```

```
while (!ss.eof()) {
       int temp;
       ss >> temp;
       v.push_back(temp);
     }
     stack<state> s;
     memset(mark, 0, sizeof(mark));
     int best = INT_MAX; //# tổng chi phí nhỏ nhất cần tìm, ban đầu đặt bằng giá trị vô
cùng lớn INT_MAX = 2^31-1
     curr = 0;
     bool not_found = true;
     s.emplace(1, v[1]);
     while (!s.empty()) {
       state &top = s.top();
       int i = top.i, j = top.j;
       s.pop();
       write(i, j);
         show_x(v.size());
//
       if (!j) continue;
```

```
if (i == v.size() - 2 && c[j][v.back()]) {
          write(++i, v.back());
            cout << "--";
//
            show_x(v.size());
//
            cout << ">> ";
//
//
            show_x(v.size());
          if (curr < best) {
             best = curr;
             not_found = false;
          }
          write(i, 0);
            cout << "--";
//
            show_x(v.size());
//
          continue;
       }
       for (int k = 1; k < v.size() - 1; ++k) {
          if (!mark[v[k]] \&\& c[j][v[k]]) {
             s.emplace(i+1, 0);
             s.emplace(i+1, v[k]);
```

```
}
      }
    }
    if (not_found) {
      cout << "--> " << 0 << endl << endl;
    } else {
      cout << "--> " << best << endl << endl;
    }
  }
  return 0;
}
// Nguyen Van Duy - 20215334
```