

x001. TÍNH TOÁN SONG SONG .....	3
v002. BẢNG SỐ .....	4
v003. CARGO.....	5
v004. DÂY CON.....	6
v005. XÂU FIBINACCI.....	7
v006. VÒNG SỐ NGUYÊN TỐ.....	8
v007. ĐÔI BẠN .....	9
008. CỬA SỔ VĂN BẢN.....	10
009. VÒNG TRÒN CON.....	11
v010. BỐ TRÍ PHÒNG HỌP .....	12
011. MUA VÉ TÀU HOẢ .....	13
v012. XIN CHỮ KÝ .....	15
013. LẮC NẠM KIM CƯƠNG .....	16
014. RẢI SỎI .....	17
015. ĐIỆP VIÊN .....	18
016. KHOẢNG CÁCH GIỮA HAI XÂU .....	19
017. XẾP LẠI BẢNG SỐ.....	20
018. THĂM KHU TRIỂN LÃM.....	21
019. DÒ MÌN .....	23
020. XẾP LẠI DÂY SỐ.....	24
021. CO DÂY BÁT PHÂN.....	25
v022. TUYẾN BAY.....	26
023. MÔ PHỎNG CÁC PHÉP TOÁN .....	27
025. TỔNG CÁC CHỮ SỐ.....	29
026. ĐƯỜNG ĐI NHIỀU ĐIỂM NHẤT .....	30
027. KẾ HOẠCH THUÊ NHÂN CÔNG.....	31
028. DÂY CÁC HÌNH CHỮ NHẬT .....	32
029. SƠN CỘT.....	33
030. CẮT VẢI.....	34
031. CHIA KẸO.....	35
032. BẢNG QUAN HỆ .....	36
033. ĐONG NƯỚC.....	37
034. TRẢ TIỀN.....	38
035. HOÁN VỊ CHỮ CÁI .....	39
036. DỰ TIỆC BÀN TRÒN .....	40
v037. TRÁO BÀI.....	41
038. ĐỐI XỨNG HOÁ .....	42
039. MẠNG MÁY TÍNH.....	43
040. LẠT ĐÔ MI NÔ .....	44
041. SỐ NHỊ PHÂN LỚN NHẤT .....	45
042. SƠN CÁC HÌNH CHỮ NHẬT.....	46
043. PHÂN HOẠCH TAM GIÁC .....	47
044. CÁC THÀNH PHẦN LIÊN THÔNG MẠNH .....	48
045. MÃ GRAY.....	49
046. DỰ ÁN XÂY CẦU.....	50
047. BẢO TỒN ĐỘNG VẬT HOANG DÃ .....	51
048. PHÁ TƯỜNG .....	52
049. TRUYỀN TIN TRÊN MẠNG .....	53
051. ĐOÀN XE QUA CẦU.....	55
052. SỐ LƯỢNG .....	56
053. THÁM HIỂM LÒNG ĐẤT .....	57
054. THỨ TỰ TỪ ĐIỂN.....	58

055. DÂY LỆCH.....	59
056. RÚT GỌN DÂY SỐ .....	60
057. BUỒN TIỀN .....	61
058. DÂY NGOẶC.....	62
059. THẲNG BỜM VÀ PHÚ ÔNG .....	63
060. SỐ THẬP PHẦN .....	64
061. DANH SÁCH VÒNG.....	65
062. TÍNH DIỆN TÍCH .....	66
063. THANG MÁY .....	67
064. TRỌNG SỐ XÂU .....	68
065. PHỎ MAY MẮN .....	69
066. TÍN HIỆU GIAO THÔNG .....	70
067. PHÂN NHÓM.....	71
068. TUA DU LỊCH RẺ NHẤT .....	72
069. DU LỊCH NHIỀU TUA NHẤT .....	73
070. PHÂN CÔNG .....	74
071. NHẮN TIN .....	75
072. CÁC SỐ ĐIỆN THOẠI .....	76
073. GIÁ TRỊ LỚN NHẤT.....	77
074. NÚT GIAO THÔNG TRỌNG ĐIỂM .....	78
075. TẬP KẾT .....	79
076. MỜI KHÁCH DỰ TIỆC.....	80
077. KHÔI PHỤC NGOẶC.....	81
078. DÂY XÍCH .....	82
079. PHÂN CÔNG .....	83
080. DÂY CUNG.....	84
081. MỀ CUNG .....	85
082. DU LỊCH KIỂU ÚC .....	86
083. SỬA ĐƯỜNG.....	87
084. ĐI THI.....	88
085. MÈO KIỂU ÚC.....	89
086. THÀNH PHỐ TRÊN SAO HOẢ.....	90
087. RÔ BÓT XÂY NHÀ.....	91
088. TƯ DUY KIỂU ÚC .....	92
089. 8-3, TẶNG HOA KIỂU ÚC .....	93
090. MÃ HOÁ BURROWS WHEELER.....	94
091. BAO LỖI.....	95
092. GIAI THỪA.....	96
093. PHỦ SÓNG.....	97
094. DÂY NGHỊCH THỂ.....	98
095. MUA HÀNG.....	99
096. XÂU CON CHUNG DÀI NHẤT .....	100
097. DÂY CON NGẮN NHẤT .....	101
098. BIẾN ĐỔI DÂY SỐ .....	102
099. GIÁ TRỊ NHỎ NHẤT .....	103
100. NÓI DÂY .....	104

## 001. TÍNH TOÁN SONG SONG

**Biểu thức đủ** là một dãy ký tự gồm các biến ký hiệu bằng chữ cái thường tiếng Anh: a..z, các phép toán cộng ký hiệu +, nhân ký hiệu \* và các dấu ngoặc (.). Được định nghĩa như sau:

i) Mỗi biến a,b,...,z là một biểu thức đủ

ii) Nếu X và Y là biểu thức đủ thì  $(X+Y)$  và  $(X*Y)$  cũng là biểu thức đủ.

iii) Những biểu thức nào không xây dựng được theo 2 nguyên tắc trên không là biểu thức đủ.

VD: Theo cách định nghĩa trên thì  $(a+(b+(c+d)))$  hoặc  $((a+b)+(c*d))$  là các biểu thức đủ.

**Cho biết thời gian tính phép + là P, thời gian tính phép \* là Q, người ta định nghĩa thời gian tính toán một biểu thức đủ như sau:**

- Nếu biểu thức đủ chỉ gồm 1 biến (a..z) thì thời gian tính toán là 0
- Nếu X và Y là 2 biểu thức đủ; thời gian tính X là TX thời gian tính Y là TY thì thời gian tính  $(X+Y)$  là  $\max(TX, TY)+P$  thời gian tính  $(X*Y)$  là  $\max(TX, TY)+Q$

**Từ 1 biểu thức đủ người ta có thể biến đổi về một biểu thức tương đương bằng các luật:**

- Giao hoán:  $(X+Y) \Leftrightarrow (Y+X)$ ;  $(X*Y) \Leftrightarrow (Y*X)$
- Kết hợp:  $(X+(Y+Z)) \Leftrightarrow ((X+Y)+Z)$ ;  $(X*(Y*Z)) \Leftrightarrow ((X*Y)*Z)$

**Yêu cầu: Cho trước một biểu thức đủ E dưới dạng xâu ký tự hãy viết chương trình:**

1. Tìm thời gian tính toán biểu thức E
2. Hãy biến đổi biểu thức E thành biểu thức E' tương đương với nó sao cho thời gian tính E' là ít nhất có thể.

**Dữ liệu vào được đặt trong file văn bản PO.INP như sau:**

- Dòng thứ nhất ghi 2 số P, Q cách nhau 1 dấu cách ( $P, Q \leq 100$ )
- Tiếp theo là một số dòng, mỗi dòng ghi 1 biểu thức đủ.

Kết quả ra đặt trong file văn bản PO.OUT như sau:

**Với mỗi biểu thức E trong file PO.INP ghi ra file PO.OUT 3 dòng**

- Dòng thứ nhất: Ghi thời gian tính toán E
- Dòng thứ hai: Ghi biểu thức E'
- Dòng thứ ba: Ghi thời gian tính toán E'

**Chú ý: Để cho gọn, mỗi biểu thức đủ trong input/output file có thể viết mà không cần đến cặp dấu ngoặc ngoài cùng, dữ liệu vào được coi là đúng đắn và không cần kiểm tra**

**Ví dụ:**

PO . INP	PO . OUT
1 1	7
$a+(a+(a+(a+(a+(a+a))))$	$((a+a)+(a+a))+((a+a)+(a+a))$
$((a+(b+(c+d)))*e)*f)$	3
$((((a*b)*c)*d)+e)+(f*g)$	5
	$(e*f)*((a+b)+(c+d))$
	3
	5
	$((a*b)*(c*d))+(e+(f*g))$
	3

## 002. BẢNG SỐ

Cho một bảng hình chữ nhật kích thước  $M \times N$  với  $M, N$  nguyên dương.  $M, N \leq 50$ . Hình chữ nhật này được chia thành  $M \times N$  ô vuông bằng nhau với kích thước đơn vị bởi các đường song song với các cạnh, trên ô vuông  $[i, j]$  ghi số nguyên  $A[i, j]$  ( $2 \leq A[i, j] \leq 50$ ).

Từ mảng  $A$  ta lập mảng  $B$  mà  $B[i, j]$  được xây dựng như sau:

Biểu diễn số  $A[i, j]$  thành tổng các số nguyên tố với ràng buộc: trong biểu diễn đó có nhiều nhất chỉ một số nguyên tố xuất hiện hai lần. Trong các cách biểu diễn, chọn ra biểu diễn nhiều hạng tử nhất thì  $B[i, j]$  bằng số số hạng của biểu diễn này kể cả bội (nếu có).

Ví dụ:

Nếu  $A[i, j] = 10 = 2 + 3 + 5$  thì  $B[i, j] = 3$ ;

Nếu  $A[i, j] = 12 = 2 + 2 + 3 + 5$  thì  $B[i, j] = 4$ ;

Chú ý: Không được biểu diễn  $A[i, j] = 10 = 2 + 2 + 2 + 2 + 2$  để có  $B[i, j] = 5$  vì như vậy không thoả mãn ràng buộc

a) Dữ liệu vào được cho bởi Text file INP.B1 trong đó:

- Dòng đầu ghi hai số  $M, N$
- $M$  dòng sau, dòng thứ  $i$  ghi  $N$  phần tử trên dòng  $i$  của bảng  $A$ :  $A[i, 1], A[i, 2], \dots, A[i, N]$  hai phần tử liên tiếp cách nhau ít nhất một dấu trống.

b) Kết quả ghi ra Text file OUT.B1:

Giá trị bảng  $B$ , mỗi dòng của bảng ghi trên một dòng của file, hai phần tử liên tiếp cách nhau ít nhất một dấu trống.

c) Hãy tìm hình chữ nhật lớn nhất được tạo bởi các ô mang giá trị bằng nhau của bảng  $B$ . Ghi tiếp ra file OUT.B1 một dòng gồm 5 số là: diện tích lớn nhất tìm được, tọa độ trên trái và dưới phải của hình chữ nhật có diện tích lớn nhất đó.

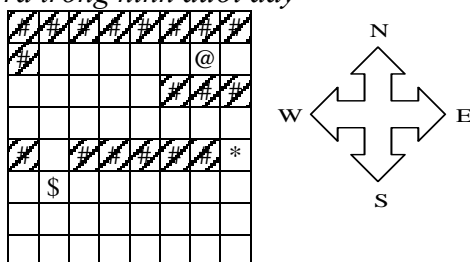
### 003. CARGO

Bản đồ một kho hàng hình chữ nhật kích thước  $m \times n$  được chia thành các ô vuông đơn vị (m hàng, n cột: các hàng đánh số từ trên xuống dưới, các cột đánh số từ trái qua phải). Trên các ô của bản đồ có một số ký hiệu:

- Các ký hiệu # đánh dấu các ô đã có một kiện hàng xếp sẵn,
- Một ký hiệu \*: Đánh dấu ô đang có một xe đẩy
- Một ký hiệu \$: Đánh dấu ô chứa kiện hàng cần xếp
- Một ký hiệu @: Đánh dấu vị trí ô mà cần phải xếp kiện hàng B vào ô đó
- Các ký hiệu dấu chấm ".": Cho biết ô đó trống

**Cần phải dùng xe đẩy ở \* để đẩy kiện hàng ở \$ đến vị trí @ sao cho trong quá trình di chuyển cũng như đẩy hàng, không chạm vào những kiện hàng đã được xếp sẵn. (Xe đẩy có thể di chuyển sang một trong 4 ô chung cạnh với ô đang đứng). Nếu có nhiều phương án thì chỉ ra một phương án sao cho xe đẩy phải di chuyển qua ít bước nhất.**

Các hướng di chuyển được chỉ ra trong hình dưới đây



**Dữ liệu:** Vào từ file văn bản CARGO.INP

- Dòng 1: Ghi hai số nguyên dương  $m, n$  cách nhau một dấu cách ( $m, n \leq 80$ )
- $m$  dòng tiếp theo, dòng thứ  $i$  ghi đủ  $n$  ký hiệu trên hàng thứ  $i$  của bản đồ theo đúng thứ tự từ trái qua phải. Các ký hiệu được ghi liền nhau

**Kết quả:** Ghi ra file văn bản CARGO.OUT

- Dòng 1: Ghi số bước di chuyển xe đẩy để thực hiện mục đích yêu cầu, nếu không có phương án khả thi thì dòng này ghi số -1
- Dòng 2: Nếu có phương án khả thi thì dòng này ghi các ký tự liền nhau thể hiện hướng di chuyển của xe đẩy R (East, West, South, North). Các chữ cái thường (e,w,s,n) thể hiện bước di chuyển không đẩy hàng, các chữ cái in hoa (E,W,S,N) thể hiện bước di chuyển có đẩy hàng.

**Ví dụ:**

CARGO . INP	CARGO . OUT
8 8	23
#####	sswwwwwNNNwnEseNwnEEEE
#.....@.	
.....###	
.....	
#.#####*	
.\$.....	
.....	
.....	

CARGO . INP	CARGO . OUT
5 9	22
@.....	eeNNNsseeeennnnwwwWWW
.##.###.#	
.....#..	
##\$###.#	
.*. ....	

## 004. DÃY CON

Cho một dãy gồm  $n$  ( $n \leq 1000$ ) số nguyên dương  $A_1, A_2, \dots, A_n$  và số nguyên dương  $k$  ( $k \leq 50$ ). Hãy tìm dãy con gồm nhiều phần tử nhất của dãy đã cho sao cho tổng các phần tử của dãy con này chia hết cho  $k$ .

**Dữ liệu vào: file văn bản DAY.INP**

- Dòng đầu tiên chứa hai số  $n, k$  ghi cách nhau bởi ít nhất 1 dấu trống.
- Các dòng tiếp theo chứa các số  $A_1, A_2, \dots, A_n$  được ghi theo đúng thứ tự cách nhau ít nhất một dấu trống hoặc xuống dòng (CR-LF).

**Kết quả: ghi ra file văn bản DAY.OUT**

- Dòng đầu tiên ghi  $m$  là số phần tử của dãy con tìm được.
- Các dòng tiếp theo ghi dãy  $m$  chỉ số các phần tử của dãy đã cho có mặt trong dãy con tìm được. Các chỉ số ghi cách nhau ít nhất một dấu trắng hoặc một dấu xuống dòng.

**Ví dụ:**

DAY . INP	DAY . OUT
10 3	9
2 3 5 7	1 3 2 4 5
9 6 12 7	6 7 10 8
11 15	

cuu duong than cong . com

cuu duong than cong . com

## 005. XÂU FIBINACCI

Xét dãy các xâu  $F_1, F_2, F_3, \dots, F_N, \dots$  trong đó:

$F_1 = 'A'$

$F_2 = 'B'$

$F_{K+1} = F_K + F_{K-1} \ (K \geq 2).$

Ví dụ:

$F_1 = 'A'$

$F_2 = 'B'$

$F_3 = 'BA'$

$F_4 = 'BAB'$

$F_5 = 'BABBA'$

$F_6 = 'BABBABAB'$

$F_7 = 'BABBABABBABBA'$

$F_8 = 'BABBABABBABBABABBABAB'$

$F_9 = 'BABBABABBABBABABBABABBABBABABBABBA'$

Cho xâu  $S$  độ dài không quá 25, chỉ bao gồm các ký tự 'A' và 'B'. Hãy xác định số lần xuất hiện xâu  $S$  trong xâu  $F_N$ ,  $N \leq 35$ . Chú ý: hai lần xuất hiện của  $S$  trong  $F_N$  không nhất thiết phải là các xâu rời nhau hoàn toàn.

**Dữ liệu:** vào từ file văn bản FIBISTR.INP, bao gồm nhiều dòng, mỗi dòng có dạng  $N \ S$ . Giữa  $N$  và  $S$  có đúng 1 dấu cách. Dữ liệu vào là chuẩn, không cần kiểm tra.

**Kết quả:** Đưa ra file văn bản FIBISTR.OUT, mỗi dòng dữ liệu ứng với một dòng kết quả ra

Ví dụ:

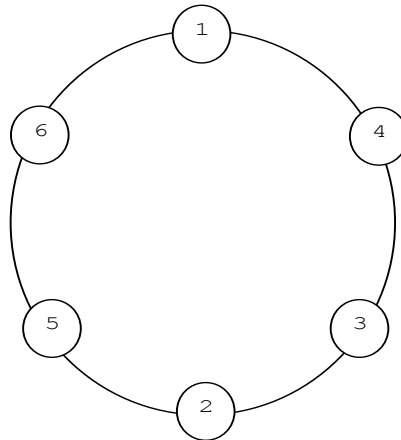
FIBISTR.INP
3 A
3 AB
8 BABBAB

FIBISTR.OUT
1
0
4

cuu duong than cong . com

## 006. VÒNG SỐ NGUYÊN TỐ

Một vòng tròn chứa  $2n$  vòng tròn nhỏ (Xem hình vẽ). Các vòng tròn nhỏ được đánh số từ 1 đến  $n$  theo chiều kim đồng hồ. Cần điền các số tự nhiên từ 1 đến  $2n$  mỗi số vào một vòng tròn nhỏ sao cho tổng của hai số trên hai vòng tròn nhỏ liên tiếp là số nguyên tố. Số điền ở vòng tròn nhỏ 1 luôn là số 1.



**Dữ liệu:** Vào từ file văn bản CIRCLE.INP chứa số nguyên dương  $n$  ( $1 < n < 10$ )

**Kết quả:** Ghi ra file văn bản CIRCLE.OUT:

- Dòng đầu tiên ghi số lượng các cách điền số tìm được ( $k$ ).
- Dòng thứ  $i$  trong số  $k$  dòng tiếp theo ghi các số trong các vòng tròn nhỏ bắt đầu từ vòng tròn nhỏ 1 đọc theo thứ tự của các vòng tròn nhỏ

cuu duong than cong . com

**Ví dụ:**

CIRCLE . INP	CIRCLE . OUT
3	2 1 4 3 2 5 6 1 6 5 2 3 4

CIRCLE . INP	CIRCLE . OUT
4	4 1 2 3 8 5 6 7 4 1 2 5 8 3 4 7 6 1 4 7 6 5 8 3 2 1 6 7 4 3 8 5 2

cuu duong than cong . com



## 007. ĐÔI BẠN

Trước kia Tuấn và Mai là hai bạn cùng lớp còn bây giờ hai bạn học khác trường nhau. Cứ mỗi sáng, đúng 6 giờ cả hai đều đi từ nhà tới trường của mình theo con đường mất ít thời gian nhất (có thể có nhiều con đường đi mất thời gian bằng nhau và đều ít nhất). Nhưng hôm nay, hai bạn muốn gặp nhau để bàn việc họp lớp cũ nhân ngày 20-11.

Cho biết sơ đồ giao thông của thành phố gồm  $N$  nút giao thông được đánh số từ 1 đến  $N$  và  $M$  tuyến đường phố (mỗi đường phố nối 2 nút giao thông). Vị trí nhà của Mai và Tuấn cũng như trường của hai bạn đều nằm ở các nút giao thông. Cần xác định xem Mai và Tuấn có cách nào đi thoả mãn yêu cầu nêu ở trên, đồng thời họ lại có thể gặp nhau ở nút giao thông nào đó trên con đường tới trường hay không? (Ta nói Tuấn và Mai có thể gặp nhau tại một nút giao thông nào đó nếu họ đến nút giao thông này tại cùng một thời điểm). Nếu có nhiều phương án thì hãy chỉ ra phương án để Mai và Tuấn gặp nhau sớm nhất.

**Dữ liệu vào được đặt trong tệp FRIEND.INP:**

- Dòng đầu tiên chứa 2 số nguyên dương  $N, M$  ( $1 \leq N \leq 100$ );
- Dòng tiếp theo chứa 4 số nguyên dương  $H_a, S_a, H_b, S_b$  lần lượt là số hiệu các nút giao thông tương ứng với: Nhà Tuấn, trường của Tuấn, nhà Mai, trường của Mai.
- Dòng thứ  $i$  trong số  $M$  dòng tiếp theo chứa 3 số nguyên dương  $A, B, T$ . Trong đó  $A$  &  $B$  là hai đầu của tuyến đường phố  $i$ . Còn  $T$  là thời gian (tính bằng giây  $\leq 1000$ ) cần thiết để Tuấn (hoặc Mai) đi từ  $A$  đến  $B$  cũng như từ  $B$  đến  $A$ .

Giả thiết là sơ đồ giao thông trong thành phố đảm bảo để có thể đi từ một nút giao thông bất kỳ đến tất cả các nút còn lại.

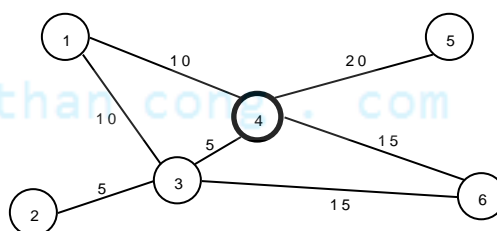
**Kết quả : Ghi ra tệp văn bản FRIEND.OUT**

- Dòng 1: Ghi từ YES hay NO tùy theo có phương án giúp cho hai bạn gặp nhau hay không. Trong trường hợp có phương án:
  - ♦ Dòng 2: Ghi thời gian ít nhất để Tuấn tới trường
  - ♦ Dòng 3: Ghi các nút giao thông theo thứ tự Tuấn đi qua
  - ♦ Dòng 4: Ghi thời gian ít nhất để Mai tới trường
  - ♦ Dòng 5: Ghi các nút giao thông theo thứ tự Mai đi qua
  - ♦ Dòng 6: Ghi số hiệu nút giao thông mà hai bạn gặp nhau
  - ♦ Dòng 7: Thời gian sớm nhất tính bằng giây kể từ 6 giờ sáng mà hai bạn có thể gặp nhau.

**Các số trên một dòng của Input/Output file ghi cách nhau ít nhất một dấu cách.**

Ví dụ : Với sơ đồ giao thông sau: ( $N=6, M=7, H_a=1, S_a=6, H_b=2, S_b=5$ )

Dòng	FRIEND . INP	FRIEND . OUT
1	6 7	YES
2	1 6 2 5	25
3	1 3 10	1 4 6
4	1 4 10	30
5	2 3 5	2 3 4 5
6	3 4 5	4
7	3 6 15	10
8	4 5 20	
9	4 6 15	



## 008. CỬA SỔ VĂN BẢN

Xét văn bản  $T$  gồm  $N$  ký tự ( $N \leq 1000000$ ,  $N$  không cho trước) và văn bản  $P$  gồm  $M$  ký tự ( $0 < M \leq 100$ ). Cửa sổ độ dài  $W$  là một đoạn văn bản gồm  $W$  ký tự liên tiếp của  $T$  ( $M < W \leq 1000$ ). Nói cửa sổ  $W$  chứa mẫu  $P$  nếu tồn tại một cách xoá một số ký tự liên tiếp của  $W$  để nhận được  $P$ .

Hai cửa sổ của  $T$  gọi là khác nhau nếu chúng bắt đầu từ những vị trí khác nhau trong  $T$ . Hãy xác định số cửa sổ khác nhau trong văn bản  $T$  chứa  $P$ .

### Dữ liệu:

- File văn bản WINDOWP.INP
  - ♦ Dòng đầu chứa hai số nguyên  $W, M$
  - ♦ Dòng thứ hai chứa  $M$  ký tự của văn bản  $P$ ;
- File WINDOWT.TXT chứa văn bản  $T$

### Kết quả:

Đưa ra file WINDOW.OUT một số nguyên xác định số cửa sổ tìm được theo yêu cầu.

Lưu ý: Đa số trường hợp, file WINDOWT.TXT không phải là Text file, có nghĩa là nó chứa các ký tự trong khoảng  $\#0..\#255$  (file of Char). Như vậy tính cả CR( $\#13$ ) và LF( $\#10$ )

### Ví dụ:

WINDOWP.INP
4 2
is

WINDOWT.TXT
This is a sample text for the
first task on the contest

WINDOW.OUT
8

cuu duong than cong . com

cuu duong than cong . com

## 009. VÒNG TRÒN CON

Cho hai dãy số nguyên  $a_1, a_2, \dots, a_m$  và  $b_1, b_2, \dots, b_n$  ( $2 \leq m, n \leq 100$ )

Các số này được xếp quanh hai vòng tròn A và B: các số  $a_i$  quanh vòng tròn A và các số  $b_j$  quanh vòng tròn B. Vòng tròn C được gọi với các số quanh nó  $c_1, c_2, \dots, c_p$  được gọi là vòng tròn con của A (hoặc của B) nếu tồn tại một cách xoá bớt các số của A (hoặc của B) để được vòng tròn C. Hãy tìm vòng tròn C là vòng tròn con của cả A và B với số phần tử (p) lớn nhất có thể.

**Chú ý:** Các số trên 3 vòng tròn A, B, C được xếp theo đúng thứ tự trong dãy theo cùng một chiều kim đồng hồ.

**Dữ liệu:** Vào từ file văn bản CIRCLE.INP

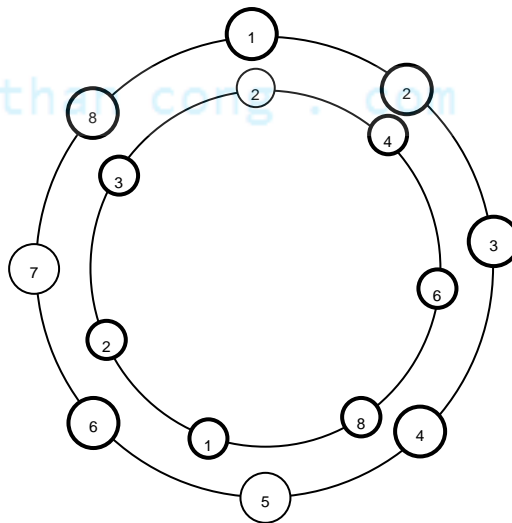
- Dòng đầu chứa hai số nguyên m, n cách nhau ít nhất một dấu cách.
- m dòng tiếp theo, dòng thứ i ghi số  $a_i$
- n dòng tiếp theo, dòng thứ j ghi số  $b_j$

**Kết quả:** Đưa ra file văn bản CIRCLE.OUT

- Dòng đầu ghi số nguyên p
- p dòng sau, dòng thứ k ghi số  $c_k$ .

**Ví dụ:**

CIRCLE . INP	CIRCLE . OUT
8 7	6
1	4
2	6
3	8
4	1
5	2
6	3
7	
8	
2	
4	
6	
8	
1	
2	
3	



cuu duong than cong . com

## 010. BỐ TRÍ PHÒNG HỌP

Có  $n$  cuộc họp đánh số từ 1 đến  $n$  đăng ký làm việc tại một phòng hội thảo. Cuộc họp  $i$  cần được bắt đầu ngay sau thời điểm  $s_i$  và kết thúc tại thời điểm  $f_i$ . Hỏi có thể bố trí phòng hội thảo phục vụ được nhiều nhất bao nhiêu cuộc họp, sao cho **khoảng** thời gian làm việc của hai cuộc họp bất kỳ là không giao nhau.

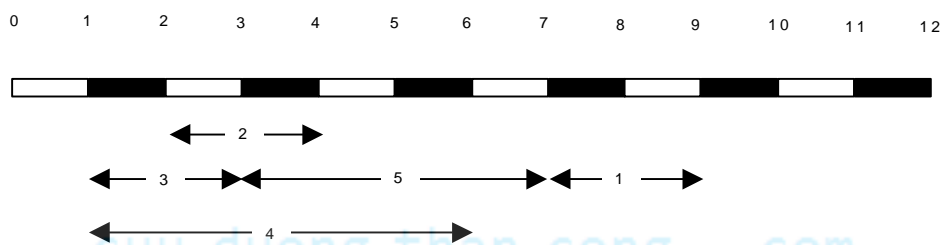
**Dữ liệu vào từ file văn bản *ACTIVITY.INP***

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $n \leq 10000$ )
- Dòng thứ  $i$  trong số  $n$  dòng tiếp theo chứa hai số nguyên dương  $s_i, f_i$  ( $s_i < f_i \leq 32000$ ) ( $\forall i: 1 \leq i \leq n$ ).

**Kết quả: Ghi ra file *ACTIVITY.OUT***

- Dòng đầu tiên ghi số  $K$  là số các cuộc họp được chấp nhận phục vụ
- $K$  dòng tiếp theo liệt kê số hiệu các cuộc họp được chấp nhận theo thứ tự từ cuộc họp đầu tiên tới cuộc họp cuối cùng, mỗi dòng ghi số hiệu một cuộc họp.

**Ví dụ:**



ACTIVITY.INP
5
7 9
2 4
1 3
1 6
3 7

ACTIVITY.OUT
3
3
5
1

## 011. MUA VÉ TÀU HOÀ

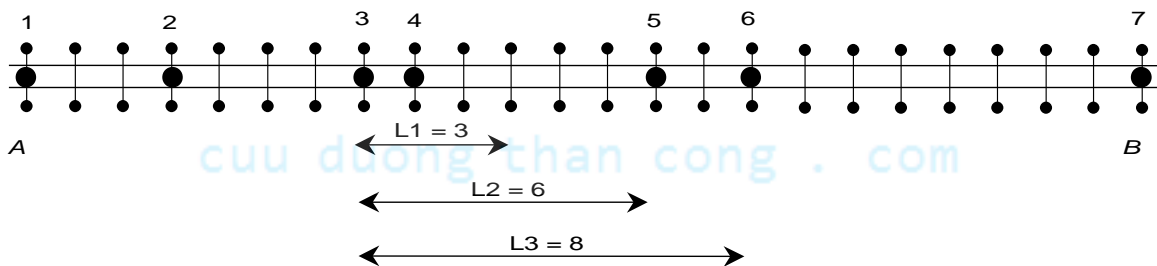
Tuyến đường sắt từ thành phố A đến thành phố B đi qua một số nhà ga. Tuyến đường có thể biểu diễn bởi một đoạn thẳng, các nhà ga là các điểm trên đó. Tuyến đường bắt đầu từ A và kết thúc ở B, vì thế các nhà ga sẽ được đánh số bắt đầu từ A (có số hiệu là 1) và B là nhà ga cuối cùng.

Giá vé đi lại giữa hai nhà ga chỉ phụ thuộc vào khoảng cách giữa chúng. Cách tính giá vé được cho trong bảng sau đây:

Khoảng cách giữa hai nhà ga (X)	Giá vé
$0 < X \leq L_1$	$C_1$
$L_1 < X \leq L_2$	$C_2$
$L_2 < X \leq L_3$	$C_3$

Vé để đi thẳng từ nhà ga này đến nhà ga khác chỉ có thể đặt mua nếu khoảng cách giữa chúng không vượt quá  $L_3$ . Vì thế nhiều khi để đi từ nhà ga này đến nhà ga khác ta phải đặt mua một số vé. Hơn thế nữa, nhân viên đường sắt yêu cầu hành khách chỉ được giữ đúng một vé khi đi trên tàu và vé đó sẽ bị huỷ khi hành khách xuống tàu.

Ví dụ, trên tuyến đường sắt cho như sau:



Để đi từ ga 2 đến ga 6 không thể mua vé đi thẳng. Có nhiều cách mua vé để đi từ ga 2 đến ga 6: Chẳng hạn đặt mua vé từ ga 2 đến ga 3 mất chi phí  $C_2$  sau đó mua vé từ ga 3 đến ga 6 mất chi phí  $C_3$ , và chi phí tổng cộng khi đi theo cách này là  $C_2 + C_3$ . Hoặc mua vé từ ga 2 đến ga 4 mất chi phí  $C_2$ , sau đó mua vé từ ga 4 đến ga 5 mất chi phí  $C_2$  và mua vé từ ga 5 đến ga 6 mất chi phí  $C_1$ , như vậy chi phí tổng cộng là  $2C_2 + C_1$ . Lưu ý rằng mặc dù khoảng cách giữa ga 2 và ga 6 bằng  $12 = 2L_2$  nhưng không được phép mua 2 vé với giá  $C_2$  để đi thẳng từ ga 2 đến ga 6.

**Yêu cầu:** Tìm cách đặt mua vé để đi lại giữa hai nhà ga cho trước với chi phí mua vé là nhỏ nhất.

**Dữ liệu vào từ file văn bản RTICKET.INP**

- Dòng đầu tiên ghi các số nguyên  $L_1, L_2, L_3, C_1, C_2, C_3$  ( $1 \leq L_1 < L_2 < L_3 \leq 10^9$ ;  $1 \leq C_1 < C_2 < C_3 \leq 10^9$ ) theo đúng thứ tự liệt kê ở trên.
- Dòng thứ hai chứa số lượng nhà ga  $N$  ( $2 \leq N \leq 10000$ ).
- Dòng thứ ba ghi hai số nguyên  $s, f$  là các chỉ số của hai nhà ga cần tìm cách đặt mua vé với chi phí nhỏ nhất để đi lại giữa chúng.
- Dòng thứ  $i$  trong số  $N - 1$  dòng tiếp theo ghi số nguyên là khoảng cách từ nhà ga A (ga 1) đến nhà ga thứ  $i + 1$ . Chi phí ít nhất từ nhà ga đầu tiên A đến nhà ga cuối cùng B không vượt quá  $10^9$ .

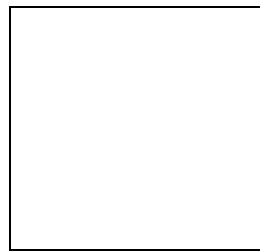
**Kết quả ghi ra file văn bản RTICKET.OUT chi phí nhỏ nhất tìm được.**

**Ví dụ:**

RTICKET . INP
3 6 8 20 30 40

RTICKET . OUT
70

7	
2	6
3	
7	
8	
13	
15	
23	



cuu duong than cong . com

cuu duong than cong . com

## 012. XIN CHỮ KÝ

Giám đốc một công ty trách nhiệm hữu hạn muốn xin chữ ký của ông Kiến trúc sư trưởng thành phố phê duyệt dự án xây dựng trụ sở làm việc của công ty. Ông kiến trúc sư trưởng chỉ ký vào giấy phép khi bà thư ký của ông ta đã ký duyệt vào giấy phép. Bà thư ký làm việc tại tầng thứ M của toà nhà trụ sở làm việc gồm M tầng của Văn phòng Kiến trúc sư trưởng thành phố. Các tầng của toà nhà được đánh số từ 1 đến M, từ thấp đến cao. Mỗi tầng của toà nhà có N phòng được đánh số từ 1 đến N từ trái qua phải. Trong mỗi phòng chỉ có một nhân viên làm việc. Giấy phép chỉ được bà thư ký ký duyệt khi đã có ít nhất một nhân viên ở tầng M đã ký xác nhận. Ngoài bà thư ký, một nhân viên bất kỳ chỉ ký xác nhận vào giấy phép khi có ít nhất một trong các điều kiện sau được thoả mãn:

- Nhân viên đó làm việc ở tầng 1
- Giấy phép đã được ký xác nhận bởi nhân viên làm việc ở cùng số phòng trong tầng sát dưới
- Giấy phép đã được ký xác nhận bởi nhân viên làm việc ở cùng số phòng trong tầng sát trên
- Giấy phép đã được ký xác nhận bởi nhân viên làm việc ở phòng bên cạnh

Mỗi một nhân viên (kể cả bà thư ký) khi ký xác nhận đều đòi một khoản lệ phí. Hãy chỉ ra cách xin được chữ ký của Kiến trúc sư trưởng đòi hỏi tổng lệ phí phải trả là nhỏ nhất (giả thiết rằng riêng chữ ký của Kiến trúc sư trưởng không mất lệ phí).

### **Dữ liệu vào từ file văn bản SIGN.INP**

- Dòng đầu tiên chứa ba số M, N, P ( $1 \leq M \leq 50$ ;  $1 \leq N \leq 100$ ;  $1 \leq P \leq N$ ) ở đây P là số phòng bà thư ký.
- Dòng thứ i trong số M dòng tiếp theo chứa N số nguyên dương theo thứ tự là lệ phí phải trả cho các nhân viên ở các phòng 1, 2, ..., N trên tầng i. Các số này không vượt quá  $10^9$  và giả thiết rằng tổng chi phí cần trả cũng không vượt quá  $10^9$ .

### **Kết quả: Ghi ra file văn bản SIGN.OUT**

Dòng đầu tiên ghi 2 số F, K theo thứ tự là chi phí cần trả và số lượng phòng cần đi qua.  
K dòng tiếp theo, mỗi dòng ghi số tầng và số phòng của một phòng theo thứ tự cần đi qua.  
(Các số trên 1 dòng của input/output file cách nhau ít nhất 1 dấu trống)

**Ví dụ:**

SIGN.INP				SIGN.OUT	
3	4	4		9	6
10	10	1	10	1	3
2	2	2	10	2	3
1	10	10	1	2	2
				2	1
				3	1
				3	4

### 013. LẮC NẠM KIM CƯƠNG

Lắc là một đồ trang sức rất được các cô gái ưa chuộng. Chính vì vậy mà chúng phải được chế tạo thật đẹp và đa dạng. Xét việc chế tạo lắc có  $m$  mắt xích, mỗi mắt được nạp một viên kim cương. Có  $n$  loại viên kim cương khác nhau,  $n \leq 7$ ;  $2 \leq m \leq 2^{7-n} + 19$ .

Hai lắc được gọi là khác nhau nếu ta không thể tìm cách đặt sao cho các mắt tương ứng có kim cương cùng loại. Lưu ý rằng lắc có hình vòng.

Với  $m$  và  $n$  cho trước, hãy xác định xem có thể tồn tại bao nhiêu loại lắc khác nhau.

Các loại kim cương được ký hiệu là A, B, C, ... Một cấu hình lắc được xác định bởi một xâu  $m$  ký tự A, B, C, ... và bắt đầu bằng ký tự nhỏ nhất.

Cho số thứ tự  $l$ , hãy xác định cấu hình tương ứng (Các cấu hình được sắp xếp theo thứ tự từ điển).

**Dữ liệu:** Vào từ file **BRASLET.INP** có dạng

$m$   $n$

$l_1$

$l_2$

...

**Kết quả:** Đưa ra file **BRASLET.OUT**

K - Số lượng lắc khác nhau

$s_1$

$s_2$

... (sẽ xác định cấu hình lắc tương ứng với  $l_i$ )

**Ví dụ:**

BRASLET . INP	
4	3
2	
21	

BRASLET . OUT
21
AAAB
CCCC



## 014. RẢI SỎI

Xét trò chơi rải sỏi với một người chơi như sau: Cho cây  $T$  và một đồng sỏi gồm  $K$  viên ở mỗi bước người ta lấy 1 viên sỏi từ đồng sỏi và đặt vào một nút lá tùy chọn. Nếu nút  $p$  có  $r$  nút lá và tất cả và tất cả các nút lá đều có sỏi thì người ta gom tất cả các viên sỏi ở lá lại, đặt 1 viên ở nút  $p$ , xoá các nút lá của nó và hoàn trả  $r - 1$  viên sỏi còn lại vào đồng sỏi. Trò chơi kết thúc khi đã đặt được 1 viên sỏi vào nút gốc. Nhiệm vụ đặt ra là theo cấu trúc của cây  $T$ , xác định số viên sỏi tối thiểu ban đầu để trò chơi có thể kết thúc bình thường. Cây có  $n$  nút ( $N \leq 400$ ), nút gốc được đánh số là 1.

**Dữ liệu:** vào từ file văn bản **STONE.INP**

- Dòng đầu: số  $n$
- Dòng thứ  $i$  trong số  $n$  dòng tiếp theo có dạng:  $i \ m \ i_1 \ i_2 \ \dots \ i_m$ . Trong đó  $m$  là số nút con của nút  $i$ ;  $i_1, i_2, \dots, i_m$ : Các nút con của nút  $i$ .

**Kết quả:** đưa ra file **STONE.OUT** số lượng viên sỏi tối thiểu cần thiết

**Ví dụ**

STONE . INP
7
1 2 2 3
2 2 5 4
3 2 6 7

STONE . OUT
3

cuu duong than cong . com

cuu duong than cong . com

## 015. ĐIỆP VIÊN

Địa bàn hoạt động của một điệp viên là một khu phố mà ở đó chỉ có các đường phố ngang, dọc tạo thành một lưới ô vuông. Với mục đích bảo mật, thay vì tên đường phố, điệp viên đánh số các phố ngang từ 0 đến  $m$  và các phố dọc từ 0 đến  $n$ . ở một số ngã ba hoặc ngã tư có các trạm kiểm soát. Anh ta đang đứng ở nút giao của hai đường  $(i_1, j_1)$  ( $j_1$  - đường ngang;  $i_1$  - đường dọc) và cần tới điểm hẹn ở giao của hai đường  $(i_2, j_2)$ . Để tránh bị theo dõi, đường đi phải không qua các trạm kiểm soát và cứ tới chỗ rẽ thì nhất thiết phải đổi hướng đi, thậm chí có thể sang đường và đi ngược trở lại. Việc đổi hướng chỉ được thực hiện ở ngã ba hoặc ngã tư. Hãy xác định đường đi ngắn nhất tới điểm hẹn hoặc cho biết không có đường đi đáp ứng được yêu cầu đã nêu.

**Dữ liệu: vào từ file SPY.INP**

Dòng đầu:  $m \ n \ i_1 \ j_1 \ i_2 \ j_2$  ( $0 \leq m, n \leq 100$ )

Các dòng sau: mỗi dòng 2 số  $i, j$  (tọa độ trạm kiểm soát).

**Kết quả: đưa ra file SPY.OUT**

Dòng đầu: độ dài đường đi ngắn nhất hoặc thông báo NO nếu không có đường đi.

Các dòng sau: mỗi dòng 2 số  $i, j$  chỉ nút tiếp theo cần tới theo đường đi tìm được, bắt đầu là  $i_1 \ j_1$  và kết thúc là  $i_2 \ j_2$ .

**Ví dụ:**

SPY . INP					
4	5	0	0	5	4
0	1				
0	4				
2	2				
2	3				
4	0				
5	2				
5	3				
-1					

SPY . OUT	
13	
0	0
1	0
1	1
1	0
2	0
2	1
3	1
3	2
4	2
4	3
3	3
4	3
4	4
5	4

## 016. KHOẢNG CÁCH GIỮA HAI XÂU

Cho hai chuỗi ký tự  $S_1$  và  $S_2$ , mỗi chuỗi có độ dài không quá 255 ký tự. Cho phép thực hiện các phép biến đổi sau đây đối với chuỗi ký tự:

1. Thay thế một ký tự nào đó bởi ký tự khác
2. Đổi chỗ hai ký tự liền nhau
3. Chèn một ký tự vào sau vị trí nào đó
4. Xóa bớt 1 ký tự

Ta gọi khoảng cách giữa hai chuỗi  $S_1$  và  $S_2$  là số ít nhất các phép biến đổi nêu trên cần áp dụng đối với chuỗi  $S_1$  để biến nó thành chuỗi  $S_2$ .

**Yêu cầu:** Tính khoảng cách giữa 2 chuỗi  $S_1, S_2$  cho trước và chỉ ra thứ tự các phép biến đổi.

**Ví dụ:** Giả sử  $S_1 = \text{'Barney'}$ ;  $S_2 = \text{'brawny'}$ . Khoảng cách giữa 2 chuỗi là 4. Dãy các phép biến đổi cần thực hiện là:

1. Thay ký tự 1 của  $S_1$  (B) bởi b
2. Đổi chỗ ký tự thứ 2 (a) và thứ 3 (r) của  $S_1$ .
3. Chèn ký tự w vào  $S_1$  sau ký tự thứ 3.
4. Xóa ký tự thứ 5 của  $S_1$ .

Dãy các phép biến đổi có thể mô tả như sau:

$\text{'Barney'} \rightarrow \text{'barney'} \rightarrow \text{'braney'} \rightarrow \text{'brawney'} \rightarrow \text{'brawny'}$

**Dữ liệu:** vào từ file văn bản **STREDIT.INP** có cấu trúc như sau:

- Dòng đầu tiên chứa chuỗi  $S_1$
- Dòng thứ hai chứa chuỗi  $S_2$

**Kết quả:** Ghi ra file văn bản **STREDIT.OUT**

- Dòng đầu tiên ghi số lượng các phép biến đổi cần sử dụng K
- Mỗi dòng i trong số K dòng tiếp theo mô tả phép biến đổi được sử dụng ở lần thứ i gồm các tham số sau: các tham số ghi trên 1 dòng ghi cách nhau 1 dấu cách.
  - ♦ 1, P, C (nếu là phép thay ký tự tại vị trí P bằng ký tự C)
  - ♦ 2, I, I + 1 (nếu là phép đổi chỗ 2 ký tự thứ I và thứ I + 1)
  - ♦ 3, P, C (nếu là phép chèn ký tự C vào sau vị trí P)
  - ♦ 4, P (nếu là phép xóa ký tự thứ P)

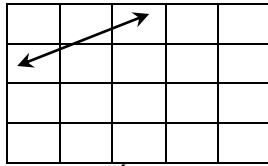
**Ví dụ:**

STREDIT.INP
Barney
brawny

STREDIT.OUT
4
1 1 b
2 2 3
3 3 w
4 5

## 017. XẾP LẠI BẢNG SỐ

Cho một bảng ô vuông gồm  $m$  hàng và  $n$  cột. Các ô được đánh chỉ số theo (hàng, cột) từ  $(0, 0)$  đến  $(m - 1, n - 1)$ . Trên  $m \times n$  ô người ta viết các số tự nhiên từ  $0$  đến  $m \times n - 1$  theo một thứ tự tùy ý. Cho phép đổi chỗ hai số đặt trong hai ô ở thế mã giao chân. Cần tìm cách đổi chỗ các số sao cho thu được bảng có tính chất: Số ở ô  $(i, j)$  là  $n \times i + j$ .



**Dữ liệu vào từ file văn bản BOARD.INP:** các số ghi trên 1 dòng cách nhau ít nhất 1 dấu trống.

- Dòng đầu ghi 2 số  $m, n$  ( $5 \leq m, n \leq 80$ )
- $m$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số tự nhiên theo đúng thứ tự các số ghi trên hàng  $i$  của bảng.

**Kết quả đưa ra file BOARD.OUT**

- Dòng thứ  $i$  chứa 4 số  $X_1, Y_1, X_2, Y_2$  cho biết tại bước thứ  $i$  cần đổi chỗ 2 số tại hai ô  $(X_1, Y_1)$  và  $(X_2, Y_2)$

**Ví dụ: ( $n = m = 8$ )**

Bảng ban đầu

	0	1	2	3	4	5	6	7
0	10	1	2	3	4	5	6	7
1	8	9	0	16	12	13	14	15
2	11	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31
4	32	33	34	35	36	37	38	39
5	40	41	55	43	44	45	46	47
6	48	49	50	51	52	53	54	42
7	56	57	58	59	60	61	62	63

Bảng cần tạo

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31
4	32	33	34	35	36	37	38	39
5	40	41	42	43	44	45	46	47
6	48	49	50	51	52	53	54	55
7	56	57	58	59	60	61	62	63

**Input/Output File:**

BOARD . INP								
8 8								
10	1	2	3	4	5	6	7	
8	9	0	16	12	13	14	15	
11	17	18	19	20	21	22	23	
24	25	26	27	28	29	30	31	
32	33	34	35	36	37	38	39	
40	41	55	43	44	45	46	47	
48	49	50	51	52	53	54	42	
56	57	58	59	60	61	62	63	

BOARD . OUT				
1	2	0	0	
2	0	3	2	
3	2	1	3	
3	2	2	0	
6	7	7	5	
7	5	6	3	
6	3	7	1	
7	1	5	2	
7	1	6	3	
6	3	7	5	
7	5	6	7	

## 018. THĂM KHU TRIỂN LÃM

Một khu triển lãm nghệ thuật có  $m \times n$  phòng được bố trí trong một hình chữ nhật kích thước  $m \times n$  ( $2 \leq m, n \leq 20$ ). Mỗi phòng biểu diễn bởi một ô và đều có cửa thông với các phòng chung cạnh với nó. Với mỗi một phòng, ta đánh chỉ số theo tọa độ  $(x, y)$  của ô ( $1 \leq \text{hàng } x \leq m; 1 \leq \text{cột } y \leq n$ ) và gán cho nó một chữ cái in hoa ('A'..'Z') thể hiện loại nghệ thuật trưng bày tại phòng đó. Có thể vào khu triển lãm ở các phòng có tọa độ  $(x \text{ bất kỳ}, y = 1)$  và có thể đi ra ở các phòng có tọa độ  $(x \text{ bất kỳ}, y = n)$

**Ví dụ với  $m=10$  và  $n=11$ :**

	1	2	3	4	5	6	7	8	9	10	11
1	B	B	B	B	B	B	F	F	F	F	F
2	A	A	A	A	A	B	D	C	C	F	F
3	A	F	F	F	A	B	A	A	C	F	C
4	B	F	E	F	A	B	B	B	B	B	D
5	F	F	D	E	A	B	A	A	A	B	A
6	E	E	D	E	E	E	E	E	A	B	B
7	D	D	D	E	E	E	E	E	A	A	B
8	D	C	C	F	F	F	C	C	A	B	A
9	D	C	C	F	F	F	C	C	A	A	A
10	C	C	C	C	C	C	C	C	C	C	C

Một vị thủ tướng đi thăm triển lãm có sở thích đặc biệt với một loại nghệ thuật. Yêu cầu của ông ta "rất đơn giản" là không nhất thiết phải đi thăm tất cả các phòng chứa loại nghệ thuật mà ông ta thích nhưng không được đi qua các phòng chứa loại nghệ thuật khác.

**Ví dụ: Để đi thăm loại nghệ thuật B, Thủ tướng có thể đi:**

$(1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (2,6), (3,6), (4,6), (4,7), (4,8), (4,9), (4,10), (5,10), (6,10), (6,11)$ .

**Nhưng không phải luôn tồn tại đường đi như vậy, ví dụ : nếu Thủ tướng muốn đi thăm loại nghệ thuật A thì không thể tìm được một đường đi (Bởi cột 6 của bảng không có một chữ A nào).**

Để có đường đi của vị thủ tướng đi thăm loại nghệ thuật A thì những người quản lý triển lãm phải tìm cách đổi loại nghệ thuật tại hai phòng nào đó. Trong ví dụ này thì để có đường đi chúng ta có thể đổi loại nghệ thuật B ở phòng  $(5,6)$  cho loại nghệ thuật A ở phòng  $(3,1)$  hoặc phòng  $(3,7), (3,8)$ ,

...

Trong những cách đổi đó, người ta thường quan tâm đến việc phải đổi sao cho tổng số phòng phải đổi là ít nhất có thể được. Trong những cách đổi với số cặp phòng phải đổi ít nhất hãy chỉ ra cách đổi mà con đường thủ tướng phải đi là ngắn nhất có thể được. Có thể có nhiều nghiệm thì chỉ cần chỉ ra một nghiệm.

**Dữ liệu vào từ file văn bản TL.INP bao gồm:**

- Dòng đầu tiên ghi số  $m, n$
- Dòng thứ hai ghi một chữ cái in hoa thể hiện loại nghệ thuật thủ tướng muốn thăm.
- $m$  dòng tiếp theo, dòng thứ  $i$  là một xâu ký tự độ dài  $n$  biểu diễn các loại nghệ thuật trong các phòng trên hàng  $i$  theo đúng thứ tự từ cột 1 đến cột  $n$ .

**Kết quả cho ra file văn bản TL.OUT bao gồm:**

- Dòng đầu tiên là số cặp phòng cần đổi  $(p)$ .
- $p$  dòng tiếp theo mỗi dòng gồm 4 số  $a, b, c, d$  có nghĩa là ta cần đổi loại nghệ thuật tại phòng  $(a,b)$  cho phòng  $(c,d)$ .
- Dòng tiếp theo ghi số phòng trên con đường đi ngắn nhất tìm được  $(q)$ .
- $q$  dòng tiếp theo, mỗi dòng ghi tọa độ  $x,y$  thể hiện cho con đường ngắn nhất đó theo đúng thứ tự phòng đi qua.
- Nếu không tồn tại phương án đổi phòng để có đường đi thì ghi vào file TL.OUT một dòng:

NO SOLUTION

**Ví dụ: Với khu triển lãm như trên:**

TL.INP	TL.OUT
10 11	0
B	16
BBBBBBBFFFF	1 1
AAAAABDCCFF	1 2
AFFFABAACFC	1 3
BFEFABBBBBD	1 4
FFDEABAAABA	1 5
EEDEEEEEABB	1 6
DDDEEEEEAAB	2 6
DCCFFFCCABA	3 6
DCCFFFCCAAA	4 6
CCCCCCCCCCC	4 7
	4 8
	4 9
	4 10
	5 10
	6 10
	6 11

TL.INP	TL.OUT
10 11	1
A	5 6 3 1
BBBBBBBFFFF	18
AAAAABDCCFF	2 1
AFFFABAACFC	2 2
BFEFABBBBBD	2 3
FFDEABAAABA	2 4
EEDEEEEEABB	2 5
DDDEEEEEAAB	3 5
DCCFFFCCABA	4 5
DCCFFFCCAAA	5 5
CCCCCCCCCCC	5 6
	5 7
	5 8
	5 9
	6 9
	7 9
	8 9
	9 9
	9 10
	9 11

cuu duong than cong . com

cuu duong than cong . com

## 019. DÒ MÌN

Cho một bãi mìn kích thước  $m \times n$  ô vuông, trên một ô có thể có chứa một quả mìn hoặc không, để biểu diễn bản đồ mìn đó, người ta có hai cách:

- Cách 1: dùng bản đồ đánh dấu: sử dụng một lưới ô vuông kích thước  $m \times n$ , trên đó tại ô  $(i, j)$  ghi số 1 nếu ô đó có mìn, ghi số 0 nếu ô đó không có mìn
- Cách 2: dùng bản đồ mật độ: sử dụng một lưới ô vuông kích thước  $m \times n$ , trên đó tại ô  $(i, j)$  ghi một số trong khoảng từ 0 đến 8 cho biết tổng số mìn trong các ô lân cận với ô  $(i, j)$  (ô lân cận với ô  $(i, j)$  là ô có chung với ô  $(i, j)$  ít nhất 1 đỉnh).

Giả thiết rằng hai bản đồ được ghi chính xác theo tình trạng mìn trên hiện trường.

**Ví dụ: Bản đồ đánh dấu và bản đồ mật độ tương ứng: ( $m = n = 10$ )**

Bản đồ đánh dấu									
1	0	1	0	1	0	1	0	0	0
0	1	0	0	0	1	0	0	1	1
0	0	1	0	1	0	0	0	0	1
0	1	1	1	1	0	0	1	1	0
0	1	1	1	0	0	0	1	0	1
0	0	0	1	0	1	0	1	0	0
1	1	1	0	0	1	1	0	1	1
1	0	0	1	0	1	0	1	0	1
0	0	1	0	1	1	1	1	1	0
1	0	0	0	0	1	0	0	0	0

Bản đồ mật độ									
1	3	1	2	1	3	1	2	2	2
2	3	3	4	3	3	2	2	2	2
2	4	4	5	3	3	2	3	5	3
2	4	6	6	3	2	2	2	4	3
2	3	6	5	5	2	4	3	5	1
3	5	6	3	4	2	5	3	5	3
2	3	3	3	5	3	5	4	4	2
2	5	4	3	5	5	7	5	6	3
2	3	1	3	4	4	5	3	3	2
0	2	1	2	3	3	4	3	2	1

Về nguyên tắc, lúc cài bãi mìn phải vẽ cả bản đồ đánh dấu và bản đồ mật độ, tuy nhiên sau một thời gian dài, khi người ta muốn gỡ mìn ra khỏi bãi thì vấn đề hết sức khó khăn bởi bản đồ đánh dấu đã bị thất lạc !! **Công việc của các lập trình viên là: Từ bản đồ mật độ, hãy tái tạo lại bản đồ đánh dấu của bãi mìn.**

cuuduongthancong.com

**Dữ liệu:** Vào từ file văn bản MINE.INP, các số trên 1 dòng cách nhau ít nhất 1 dấu cách

- Dòng 1: Ghi 2 số nguyên dương  $m, n$  ( $2 \leq m, n \leq 80$ )
- $m$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số trên hàng  $i$  của bản đồ mật độ theo đúng thứ tự từ trái qua phải.

**Kết quả:** Ghi ra file văn bản MINE.OUT, các số trên 1 dòng ghi cách nhau ít nhất 1 dấu cách

- Dòng 1: Ghi tổng số lượng mìn trong bãi
- $m$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số trên hàng  $i$  của bản đồ đánh dấu theo đúng thứ tự từ trái qua phải.

**Ví dụ:**

MINE . INP									
10	15								
0	3	2	3	3	3	5	3	4	4
1	4	3	5	5	4	5	4	7	7
1	4	3	5	4	3	5	4	4	4
1	4	2	4	4	5	4	2	4	4
1	3	2	5	4	4	2	2	3	2
2	3	2	3	3	5	3	2	4	4
2	3	2	4	3	3	2	4	6	6
2	6	4	5	2	4	1	3	3	5
4	6	5	7	3	5	3	5	5	6
2	4	4	4	2	3	1	2	2	2

MINE . OUT									
80									
1	0	1	1	1	1	0	1	1	1
0	0	1	0	0	1	1	1	0	1
0	0	1	0	0	1	0	0	1	1
1	0	1	1	1	0	0	1	0	0
1	0	0	0	1	1	1	0	0	1
0	0	0	0	1	0	0	0	1	0
0	1	1	0	0	1	0	0	1	0
1	0	1	0	1	0	1	0	1	1
0	1	1	0	1	0	0	0	0	1
1	1	1	1	1	0	1	1	1	0

## 020. XẾP LẠI DÃY SỐ

Cho dãy  $A = (a_1, a_2, \dots, a_n)$  là dãy các số nguyên dương đôi một khác nhau.

Hãy liệt kê tất cả các cách hoán vị phần tử của dãy  $A$  thoả mãn: giữa hai giá trị  $M$  và  $N$  bất kỳ trong hoán vị đó, không tồn tại giá trị  $P$  nào để:  $2P = M + N$ .

*Ví dụ: Với dãy  $A$  là  $(11, 22, 33, 44)$  thì*

*Hoán vị  $(11, 44, 33, 22)$  là thoả mãn điều kiện trên*

*Hoán vị  $(11, 44, 22, 33)$  không thoả mãn vì có giá trị  $P = 22$  nằm giữa hai giá trị  $M = 11$  và  $N = 33$  mà:  $22 * 2 = 11 + 33$ .*

**Dữ liệu:** Vào từ file văn bản **SORT.INP**. Các số trên 1 dòng cách nhau ít nhất 1 dấu trống

- Dòng 1: Ghi số  $n$  ( $2 \leq n \leq 11$ )
- Dòng 2: Ghi đủ giá trị  $n$  phần tử của dãy  $A$  ( $1 \leq a_i \leq 100$ ).

**Kết quả:** Ghi ra file văn bản **SORT.OUT**. Các số trên 1 dòng cách nhau ít nhất 1 dấu trống

- Dòng cuối cùng ghi số lượng hoán vị tìm được ( $K$ )
- $K$  dòng trước dòng cuối cùng, mỗi dòng ghi 1 hoán vị tìm được

**Ví dụ:**

SORT . INP
4
11 22 33 44

SORT . OUT
11 33 22 44
11 33 44 22
22 11 44 33
22 44 11 33
22 44 33 11
33 11 22 44
33 11 44 22
33 44 11 22
44 22 11 33
44 22 33 11
10



## 021. CO DÃY BẤT PHÂN

Cho một bảng A kích thước 8x8; Các dòng và các cột được đánh số từ 0 đến 7. Trên mỗi ô của bảng chứa một số nguyên trong khoảng từ 0 đến 7.

Cho dãy  $X = (x_1, x_2, \dots, x_n)$ , có các phần tử  $x_i \in \mathbb{N}$ ;  $0 \leq x_i \leq 7$ . ( $2 \leq n \leq 200$ ).

Với  $\forall i: 1 \leq i < n$ . **Phép co  $R(i)$  thực hiện trên dãy X: Xóa hai phần tử  $x_i$  và  $x_{i+1}$  và thay vào đó giá trị nằm trên hàng  $x_i$ , cột  $x_{i+1}$  của bảng A, sau đó dãy X được đánh chỉ số lại từ trái qua phải bắt đầu từ 1.**

Ví dụ:

A	0	1	2	3	4	5	6	7
0	0	1	2	3	0	0	0	0
1	3	2	3	0	0	0	0	0
2	5	3	0	1	0	0	0	0
3	7	0	1	2	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Ví dụ: Với bảng A như trên và dãy  $X = (0, 1, 2, 3, 1, 2)$  nếu ta thực hiện phép co  $R(3)$  thì ta sẽ được dãy  $(0, 1, 1, 1, 2)$ . Nếu thực hiện tiếp  $R(4)$  thì ta sẽ được dãy  $(0, 1, 1, 3)$ . Thực hiện tiếp  $R(2)$  thì sẽ được dãy  $(0, 2, 3)$ . Thực hiện tiếp  $R(1)$  thì sẽ còn  $(2, 3)$  và thực hiện  $R(1)$  một lần nữa sẽ được  $(1)$ .

**Yêu cầu:** cho trước một giá trị  $V$  ( $0 \leq V \leq 7$ ), hãy tìm một thứ tự thực hiện  $n - 1$  phép co trên dãy  $X$  để giá trị còn lại cuối cùng là  $V$ . Nếu có nhiều phương án thì chỉ cần cho biết một.

**Dữ liệu** vào từ file văn bản OCT.INP

- 8 dòng đầu tiên, dòng thứ  $i$  ghi 8 số trên hàng thứ  $i - 1$  của bảng A theo đúng thứ tự từ trái qua phải
- Dòng thứ 9 ghi số  $n$
- Dòng thứ 10 ghi đủ  $n$  số:  $x_1, x_2, \dots, x_n$  theo đúng thứ tự.
- Dòng thứ 11 ghi giá trị  $V$ .

**Kết quả ghi ra file văn bản OCT.OUT**, chỉ gồm 1 dòng, trên đó:

- Ghi số 0 nếu không tồn tại phương án sử dụng  $n - 1$  phép co để cho giá trị  $V$ . Hoặc ghi (theo đúng thứ tự thực hiện) đủ  $n - 1$  vị trí của các phép co trên dãy  $X$  để cho giá trị  $V$ .

Chú ý: Các số trên 1 dòng của Input/Output File ghi cách nhau ít nhất 1 dấu cách.

Ví dụ:

OCT . INP
5 7 2 1 7 1 4 0
0 6 0 0 1 3 1 6
0 4 5 1 3 6 6 1
2 5 6 5 5 3 2 5
2 7 1 3 7 3 5 1
2 5 2 4 6 0 4 5
6 3 5 6 7 6 0 2
0 6 0 1 3 3 4 4
15
5 2 3 0 1 6 1 0 4 2 4 3 2 4 4
6

OCT . OUT
13 13 10 10 10 9 7 7 6 5 3 3 2 1

## 022. TUYẾN BAY

Có  $N$  thành phố và  $M$  đường hàng không hai chiều giữa một số cặp thành phố nào đó, các đường bay được quản lý bởi 16 hãng hàng không. Các thành phố được đánh số từ 1 tới  $N$  ( $N \leq 100$ ) và các hãng được đánh số từ 1 tới 16.

Được biết chi phí bay trực tiếp giữa hai thành phố  $i, j$  bất kỳ (nếu như có đường bay) là  $C$ . Nếu đang đi máy bay của một hãng đến sân bay nào đó rồi chuyển sang máy bay của hãng khác thì sẽ phải mất thêm một khoản phụ phí  $A$ .

**Yêu cầu:** Cho trước hai thành phố  $S$  và  $F$ , hãy tìm hành trình bay từ thành phố  $S$  đến thành phố  $F$  với chi phí ít nhất. Với giả thiết rằng luôn luôn tồn tại cách bay từ  $S$  tới  $F$ .

**Dữ liệu:** Vào từ file văn bản AIRLINES.INP. Trong đó:

- Dòng 1 ghi sáu số nguyên dương  $N, M, C, A, S, F$ . ( $1 \leq A, C \leq 100$ )
- $M$  dòng tiếp theo, mỗi dòng có dạng  $u \ v \ k_1 \ k_2 \dots$  cho biết rằng giữa thành phố  $u$  và thành phố  $v$  có đường bay và  $k_1, k_2, \dots$  là số hiệu các hãng sở hữu đường bay đó

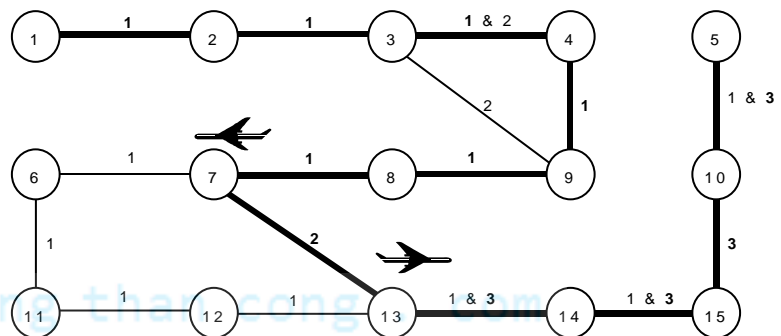
**Kết quả:** Ghi ra file văn bản AIRLINES.OUT. Trong đó:

- Dòng 1: Ghi chi phí tối thiểu phải trả
- Các dòng tiếp theo, mỗi dòng ghi một bộ ba  $i, j, k$ . Thể hiện tại bước đó sẽ bay từ thành phố  $i$  đến thành phố  $j$  bởi máy bay của hãng  $k$ . Thứ tự các dòng phải theo đúng thứ tự bay trong hành trình.

**Các số trên một dòng của Input/Output file ghi cách nhau ít nhất một dấu cách.**

**Ví dụ:** Với mạng lưới đường không như dưới đây: cần đi từ thành phố 1 đến thành phố 5. Chi phí đường bay trực tiếp giữa hai thành phố bất kỳ  $C = 3$ , phụ phí chuyển tuyến  $A = 2$ . Các số ghi bên cạnh các đường bay trực tiếp là tên các hãng sở hữu đường bay đó.

AIRLINES . INP	AIRLINES . OUT
15 16 3 2 1 5	37
1 2 1	1 2 1
2 3 1	2 3 1
3 4 1 2	3 4 1
3 9 2	4 9 1
4 9 1	9 8 1
5 10 1 3	8 7 1
6 7 1	7 13 2
6 11 1	13 14 3
7 8 1	14 15 3
7 13 2	15 10 3
8 9 1	10 5 3
10 15 3	
11 12 1	
12 13 1	
13 14 1 3	
14 15 1 3	



## 023. MÔ PHỎNG CÁC PHÉP TOÁN

Cho hai số nguyên dương  $a$  và  $b$  ( $1 \leq b \leq a < 10^{1000}$ ), hãy tính  $a + b$ ,  $a - b$ ,  $a * b$ ,  $a \text{ div } b$ ,  $a \text{ mod } b$ .

**Dữ liệu:** Vào từ file văn bản OPT.INP

- Dòng 1: Chứa số  $a$
- Dòng 2: Chứa số  $b$

**Kết quả:** Ghi ra file văn bản OPT.OUT

- Dòng 1: Ghi giá trị  $a + b$
- Dòng 2: Ghi giá trị  $a - b$
- Dòng 3: Ghi giá trị  $a * b$
- Dòng 4: Ghi giá trị  $a \text{ div } b$
- Dòng 5: Ghi giá trị  $a \text{ mod } b$

**Ví dụ:**

OPT . INP	OPT . OUT
56	106
50	6
	2800
	1
	6

OPT . INP	OPT . OUT
987111	1055001
67890	919221
	67014965790
	14
	36651

cuu duong than cong . com

cuu duong than cong . com

## 024. DÃY CON CỦA DÃY NHỊ PHÂN

Xét dãy  $B_0, B_1, B_2, \dots, B_n$  là các dãy các xâu nhị phân, được xây dựng như sau:

$$B_0 = '1'$$

Với  $\forall i: (i \geq 1)$  thì  $B_i$  là ghép của  $B_{i-1}$  với  $\neg(B_{i-1})$ . Trong đó  $\neg(S)$  là xâu được tạo thành từ xâu  $S$  bằng cách đảo tất cả các số 1 thành 0 và số 0 thành 1

$$B_0 = 1$$

$$B_1 = 10$$

$$B_2 = 1001$$

$$B_3 = 10010110$$

$$B_4 = 1001011001101001$$

$$B_5 = 10010110011010010110100110010110$$

$$B_6 = 1001011001101001011010011001011001101001100101101001011001101001$$

...

Yêu cầu: Cho trước số nguyên dương  $n \leq 30$  và một số  $k \leq 2^n$ . hãy cho biết ký tự thứ  $k$  của  $B_n$  là ký tự 0 hay 1.

cuu duong than cong . com

cuu duong than cong . com

## 025. TỔNG CÁC CHỮ SỐ

Cho trước hai số nguyên dương  $n$  và  $k$  ( $n \leq 20$ ,  $k \leq 30$ ).

Yêu cầu 1: Hãy cho biết có bao nhiêu số có  $\leq n$  chữ số mà tổng các chữ số đúng bằng  $k$

Yêu cầu 2: Cho số nguyên dương  $p$ , hỏi nếu đem các số tìm được sắp xếp theo thứ tự tăng dần thì số thứ  $p$  là số nào. ( $p$  không lớn hơn số lượng các số tìm được)

**Dữ liệu:** Vào từ file văn bản DIGITSUM.INP gồm 1 dòng chứa ba số  $n$ ,  $k$ ,  $p$  theo đúng thứ tự cách nhau 1 dấu cách.

**Kết quả:** Ghi ra file văn bản DIGITSUM.OUT gồm 2 dòng

- Dòng 1: Ghi số lượng các số tìm được trong yêu cầu 1
- Dòng 2: Ghi số thứ  $p$  trong yêu cầu 2 tìm được

**Ví dụ:**

DIGITSUM.INP
3 8 10

DIGITSUM.OUT
45
107

cuu duong than cong . com

cuu duong than cong . com

## 026. ĐƯỜNG ĐI NHIỀU ĐIỂM NHẤT

Cho một bảng A kích thước  $m \times n$  ( $1 \leq m, n \leq 100$ ), trên đó ghi các số nguyên  $a_{ij}$  ( $|a_{ij}| \leq 100$ ). Một người xuất phát tại ô nào đó của cột 1, cần sang cột  $n$  (tại ô nào cũng được).

Quy tắc đi: Từ ô  $(i, j)$  chỉ được quyền sang một trong 3 ô  $(i, j + 1)$ ;  $(i - 1, j + 1)$ ;  $(i + 1, j + 1)$ . Xem hình vẽ:

1	2	6	7	9
7	6	5	6	7
1	2	3	4	2
4	7	8	7	6

**Yêu cầu:** Hãy tìm vị trí ô xuất phát và một hành trình đi từ cột 1 sang cột  $n$  sao cho tổng các số ghi trên đường đi là lớn nhất.

**Dữ liệu:** Vào từ file văn bản MAX.INP. Trong đó:

- Dòng 1: Ghi hai số  $m, n$  là số hàng và số cột của bảng.
- $m$  dòng tiếp theo, dòng thứ  $i$  ghi đủ  $n$  số trên hàng  $i$  của bảng theo đúng thứ tự từ trái qua phải.

**Kết quả:** Ghi ra file văn bản MAX.OUT. Trong đó:

- Dòng 1: Ghi số điểm tối đa có được
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi chỉ số hàng của ô thứ  $i$  trong hành trình.

**Các số trên 1 dòng trong Input/ Output file cách nhau ít nhất 1 dấu cách**

**Ví dụ:**

	1	2	3	4	5	6	7
1	9	-2	6	2	1	3	4
2	0	-1	6	7	1	3	3
3	8	-2	8	2	5	3	2
4	1	-1	6	2	1	6	1
5	7	-2	6	2	1	3	7

MAX . INP	MAX . OUT
5 7	41
9 -2 6 2 1 3 4	1
0 -1 6 7 1 3 3	2
8 -2 8 2 5 3 2	3
1 -1 6 2 1 6 1	2
7 -2 6 2 1 3 7	3
	4
	5

## 027. KẾ HOẠCH THUÊ NHÂN CÔNG

Giám đốc điều hành của một Công ty tin học cần xác định số lượng nhân công cần sử dụng trong mỗi tháng để thực hiện một dự án phát triển tin học. Ông giám đốc nắm được số lượng nhân công tối thiểu cần cho mỗi tháng. Mỗi lần thuê hoặc sa thải một nhân công luôn mất thêm một khoản chi phí. Mỗi khi một thợ nào đó được thuê, anh ta luôn nhận được tiền lương ngay cả khi không làm việc. Giám đốc nắm được chi phí để thuê một nhân công mới, chi phí sa thải một nhân công, lương tháng của một nhân công. Vấn đề đặt ra cho giám đốc là phải xác định số lượng nhân công cần thuê hay sa thải trong mỗi tháng để cho chi phí thực hiện dự án là tối thiểu.

**Dữ liệu: Vào từ file văn bản PROJECT.INP.**

- Dòng đầu tiên ghi thời gian thực hiện dự án  $n$  (đơn vị tính: tháng,  $n \leq 12$ )
- Dòng thứ hai chứa ba số nguyên dương theo thứ tự là chi phí thuê một nhân công mới, lương tháng của một nhân công, chi phí sa thải một nhân công.
- Dòng cuối cùng ghi  $n$  số nguyên dương  $d_1, d_2, \dots, d_n$ , trong đó  $d_i$  là số lượng nhân công cần sử dụng trong tháng  $i$ .

**Kết quả: Ghi ra file văn bản PROJECT.OUT**

- Dòng đầu tiên ghi chi phí tối thiểu tìm được
- Mỗi dòng thứ  $i$  trong số  $n$  dòng tiếp theo ghi số  $s_i$ . Được hiểu là:
  - ♦ Nếu  $s_i > 0$  thì nó là số lượng nhân công cần thuê thêm ở tháng  $i$ .
  - ♦ Nếu  $s_i < 0$  thì  $|s_i|$  là số lượng nhân công cần sa thải ở tháng  $i$
  - ♦ Nếu  $s_i = 0$  thì không có biến động nhân sự trong tháng  $i$  của dự án

**Ví dụ:**

PROJECT . INP			
3			
4	5	6	
10	9	11	

PROJECT . OUT	
199	
10	
0	
1	

cuu duong than cong . com

## 028. DÂY CÁC HÌNH CHỮ NHẬT

Giả sử ABCD là một hình chữ nhật trên mặt phẳng tọa độ có các đỉnh:

A (0, 0); B(0, 1); C(K, 1) và D(K, 0).

Ta xem hình này là hình có số hiệu 1.

Hình có số hiệu 2 xây dựng trên cạnh Bắc của hình 1 và cạnh kia gấp K lần. Hình có số hiệu 3 xây dựng trên cạnh tây của hình chữ nhật hợp các hình 1 và 2 và cạnh kia gấp K lần. Hình có số hiệu 4 xây dựng trên cạnh nam của hợp các hình 1,2,3 và cạnh kia gấp K lần. Hình có số hiệu 5 xây dựng trên cạnh đông của hợp các hình 1,2,3,4 và cạnh kia gấp K lần. Tương tự quy luật đó với các hình mang thứ tự 6,7...

Bài toán đặt ra là cho trước 3 số thực K,X,Y, hãy cho biết số hiệu nhỏ nhất của hình chữ nhật chứa điểm có tọa độ (X,Y)

**Dữ liệu:** Vào từ bởi file văn bản REC.INP gồm 1 số dòng.

Mỗi dòng gồm 3 số K,X,Y với ý nghĩa nêu trên.

**Kết quả:** Ghi ra file văn bản REC.OUT như sau:

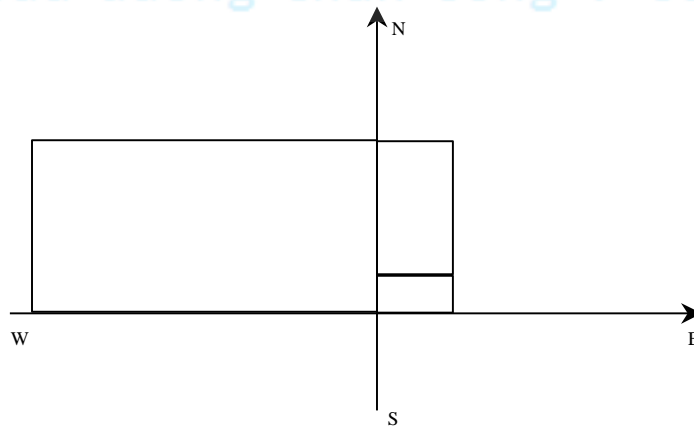
Với mỗi dòng của file dữ liệu ghi trên 1 dòng số hiệu của điểm đã cho:

Chú ý: K, X, Y có thể có tới 100 chữ số.

**Ví dụ:**

REC . INP
3 0 1
2 7 -2
4 1 17

REC . OUT
1
5
2





## 029. SƠN CỘT

Trên một nền phẳng đã được chia thành các lưới ô vuông đơn vị gồm  $m \times n$  ô ( $m, n \leq 100$ ), người ta đặt chồng khít lên nhau các khối lập phương đơn vị thành những cột. Khối dưới cùng của cột chiếm trọn một ô của lưới. Chiều cao của mỗi cột được tính bằng số khối lập phương đơn vị tạo thành cột đó. Sau khi xếp xong toàn bộ các cột, người ta tiến hành **sơn các mặt nhìn thấy được** của các cột.

***Yêu cầu: Biết chiều cao của mỗi cột, hãy tính số đơn vị diện tích cần sơn.***

**Dữ liệu** vào đặt trong file văn bản PAINT.INP. Trong đó:

Dòng đầu tiên ghi hai số nguyên dương  $m, n$  là kích thước của lưới nền ( $m$  hàng,  $n$  cột)  
 $m$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số nguyên không âm, số nguyên thứ  $j$  biểu thị chiều cao của cột dựng tại ô  $(i, j)$  của lưới. Các số cách nhau ít nhất một dấu cách.

**Kết quả** ra đặt trong file văn bản PAINT.OUT, ghi số diện tích cần sơn.

**Ví dụ:**

Với hình vẽ bên, các cột được xây trên nền kích thước  $2 \times 3$ . Các file dữ liệu vào và kết quả ra sẽ là:

PAINT . INP		
2	3	
4	3	4
1	2	1

PAINT . OUT
42

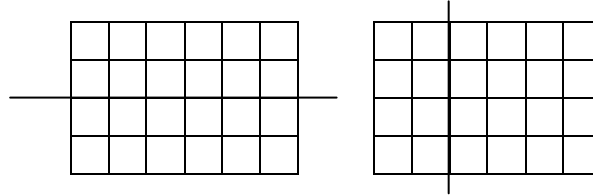
cuu duong than cong . com

cuu duong than cong . com

### 030. CẮT VẢI

Một cơ sở may mặc chuyên sản xuất khăn vuông đủ mọi kích cỡ, nguyên liệu là các tấm vải. Với một tấm vải hình chữ nhật chiều dài  $m$  đơn vị và chiều rộng  $n$  đơn vị ( $m, n$  nguyên dương không quá 100), người ta có hai cách cắt, cắt ngang và cắt dọc.

Đặc điểm của mỗi thao tác cắt là: mỗi lần cắt bắt buộc phải cắt rời một mảnh vải hình chữ nhật thành hai mảnh khác cũng hình chữ nhật và kích thước hai mảnh cắt rời đó cũng phải là số nguyên.



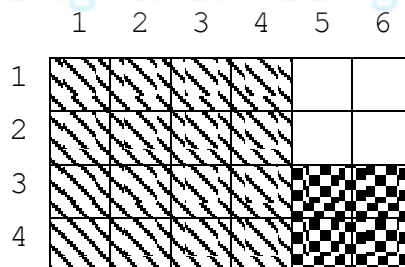
**Yêu cầu:** Cho trước tấm vải kích thước  $m \times n$ . Hãy tìm cách cắt tấm vải đó thành những mảnh vuông (không được để lại một mảnh nào không vuông) sao cho số mảnh vuông cắt ra là ít nhất.

**Dữ liệu:** Vào từ file văn bản CUT.INP gồm 1 dòng chứa hai số  $m, n$  cách nhau 1 dấu cách

**Kết quả:** Ghi ra file văn bản CUT.OUT. Trong đó:

- Dòng 1: Ghi số  $K$  là số mảnh vuông tối thiểu có thể cắt ra được
- $K$  dòng tiếp theo, mỗi dòng ghi 3 số  $X, Y, d$ . ở đây  $(X, Y)$  là tọa độ ô vuông ở góc trái trên của một hình vuông cắt ra được và  $d$  là độ dài cạnh hình vuông đó. Quy ước tọa độ của ô ở góc trái trên hình chữ nhật ban đầu là  $(1, 1)$ . Tọa độ của ô ở góc phải dưới hình chữ nhật ban đầu là  $(m, n)$ . Ba số  $X, Y, d$  ghi cách nhau ít nhất 1 dấu cách.

**Ví dụ:**



CUT . INP
4 6

CUT . OUT
3
1 1 4
1 5 2
3 5 2

### 031. CHIA KẸO

Cho  $n$  gói kẹo đánh số từ 1 đến  $n$ , gói kẹo thứ  $i$  có  $A_i$  viên kẹo.

Giả thiết  $2 \leq n \leq 200$  và  $1 \leq A_i \leq 200$  với  $\forall i: 1 \leq i \leq n$ .

Yêu cầu: Chia  $n$  gói kẹo đã cho làm hai nhóm sao cho hiệu số kẹo của hai nhóm chênh lệch nhau ít nhất, nếu có nhiều cách chia thì chỉ cần chỉ ra một cách.

**Dữ liệu:** Vào từ file văn bản **CANDY.INP**. Trong đó:

- Dòng đầu tiên ghi số  $n$
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi số  $A_i$

**Kết quả:** Ghi ra file văn bản **CANDY.OUT**. Trong đó:

- Dòng đầu tiên ghi hai số  $m_1$  và  $c_1$  cách nhau ít nhất một dấu cách,  $m_1$  là số gói nhóm I,  $c_1$  là số kẹo nhóm I.
- $m_1$  dòng tiếp theo, mỗi dòng ghi chỉ số một gói kẹo được chọn vào nhóm I
- Dòng  $m_1+2$  ghi hai số  $m_2$  và  $c_2$  cách nhau ít nhất một dấu cách,  $m_2$  là số gói nhóm II,  $c_2$  là số kẹo nhóm II.
- $m_2$  dòng tiếp theo, mỗi dòng ghi chỉ số một gói kẹo được chọn vào nhóm II

**Ví dụ:**

CANDY . INP	CANDY . OUT
6	3 111
100	1
4	4
9	5
5	3 111
6	2
98	3
	6

CANDY . INP	CANDY . OUT
10	6 27
1	2
2	3
3	4
4	5
5	6
6	7
7	4 28
8	1
9	8
10	9
	10

## 032. BẢNG QUAN HỆ

Cho bảng vuông  $A$ , kích thước  $n \times n$ , các phần tử là số nguyên  $\in \{-2, -1, 0, 1, 2, 3\}$ .

Giả thiết  $2 \leq n \leq 200$ .

Bảng  $A$  gọi là tương thích với dãy  $T = (t_1, t_2, \dots, t_n)$ , hay dãy  $T$  tương thích với bảng  $A$  nếu:

- $A_{ij} = 0 \Rightarrow t_i = t_j$
- $A_{ij} = 1 \Rightarrow t_i < t_j$
- $A_{ij} = -1 \Rightarrow t_i > t_j$
- $A_{ij} = 2 \Rightarrow t_i \leq t_j$
- $A_{ij} = -2 \Rightarrow t_i \geq t_j$
- $A_{ij} = 3 \Rightarrow t_i \neq t_j$

(Với mọi  $i, j: 1 \leq i, j \leq n$ )

Ví dụ: Dãy  $T = (1, 4, 5, 4, 5, 9)$  tương thích với bảng:

A	1	2	3	4	5	6
1	0	1	1	1	2	2
2	-2	0	1	0	2	2
3	-2	-1	0	3	0	1
4	-2	-2	3	0	1	1
5	-1	-2	0	-1	0	1
6	-1	-2	-1	-1	-1	0

Dãy  $T = (10, 20, 30, 20, 30, 40)$  cũng tương thích với bảng

***Yêu cầu, cho trước bảng quan hệ  $A$ , hãy tìm dãy số nguyên dương  $T = (t_1, t_2, \dots, t_n)$  tương thích với bảng  $A$  mà  $\max(T)$  là bé nhất có thể. Biết rằng luôn tồn tại một dãy như vậy***

**Dữ liệu:** Vào từ file văn bản REL.INP:

- Dòng 1: Chứa số  $n$
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số trên dòng  $i$  của bảng  $A$  theo đúng thứ tự từ  $A_{i1}$  đến  $A_{in}$

**Kết quả:** Ghi ra file văn bản REL.OUT:

Chỉ gồm 1 dòng ghi  $n$  số của dãy  $T$  tìm được theo đúng thứ tự từ  $t_1$  đến  $t_n$ .

***Các số trên một dòng của Input/ Output File cách nhau ít nhất 1 dấu cách***

**Ví dụ:**

REL . INP						
6						
0	1	1	1	2	2	
-2	0	1	0	2	2	
-2	-1	0	3	0	1	
-2	-2	3	0	1	1	
-1	-2	0	-1	0	1	
-1	-2	-1	-1	-1	0	

REL . OUT						
1	2	3	2	3	4	

### 033. ĐONG NƯỚC

Nền phẳng của một công trường xây dựng đã được chia thành lưới ô vuông đơn vị kích thước  $m \times n$  ô. Trên mỗi ô  $(i, j)$  của lưới, người ta dựng một cột bê tông hình hộp có đáy là ô  $(i, j)$  và chiều cao là  $H_{ij}$  đơn vị. Sau khi dựng xong, thì trời đổ mưa to và đủ lâu. Giả thiết rằng nước không thấm thấu qua các cột bê tông cũng như không rò rỉ qua các đường ghép giữa chúng.

**Yêu cầu:** Xác định lượng nước đọng giữa các cột

Chú ý kỹ thuật:  $m, n, H_{ij}$  là các số nguyên dương.  $1 \leq m, n \leq 100$ .  $1 \leq H_{ij} \leq 1000$

**Dữ liệu:** Vào từ file văn bản WATER.INP được ghi dưới khuôn dạng sau:

Dòng 1:  $m \quad n$   
 Dòng 2:  $H_{11} \quad H_{12} \quad \dots \quad H_{1n}$   
 Dòng 3:  $H_{21} \quad H_{22} \quad \dots \quad H_{2n}$   
 ...  
 Dòng  $m + 1$ :  $H_{m1} \quad H_{m2} \quad \dots \quad H_{mn}$

Các số trên 1 dòng các nhau ít nhất 1 dấu cách

**Kết quả:** Ghi ra file văn bản WATER.OUT chứa số đơn vị khối nước đọng

**Ví dụ:**

WATER . INP	WATER . OUT
5 5 9 9 9 9 9 9 2 2 2 9 9 2 1 2 9 9 2 2 2 9 9 9 9 9 9	64

WATER . INP	WATER . OUT
5 7 3 3 3 3 3 3 3 3 1 1 1 1 1 3 3 1 2 2 2 1 3 3 1 1 1 1 1 3 3 3 3 3 3 3 3	27

WATER . INP	WATER . OUT
10 10 9 9 9 9 9 9 9 9 9 9 9 1 1 1 1 9 1 1 1 9 9 1 1 1 1 1 1 1 1 9 9 1 1 1 1 9 1 1 1 9 9 9 9 9 9 9 9 1 9 9 9 1 1 1 1 9 1 1 1 9 9 1 1 1 1 9 1 1 1 9 9 1 1 1 1 9 1 1 1 9 9 1 1 1 1 9 1 1 1 9 9 9 9 9 9 9 9 1 9 9	128

### 034. TRẢ TIỀN

Nước Silverland sử dụng hệ thống 20 loại tiền xu, trong đó các xu có mệnh giá là một số chính phương từ  $1^2$  đến  $20^2$ :

1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400.

Với hệ thống này, để trả 10 xu ta có 4 cách:

1. Trả 10 đồng 1 xu
2. Trả 6 đồng 1 xu và 1 đồng 4 xu
3. Trả 2 đồng 1 xu và 2 đồng 4 xu
4. Trả 1 đồng 1 xu và 1 đồng 9 xu

*Nhiệm vụ của bạn là xác định xem có bao nhiêu cách trả một số tiền cho trước ở Silverland và cho biết một cách trả phải dùng ít đồng xu nhất.*

**Dữ liệu vào từ file văn bản COIN.INP**

Ghi số tiền nguyên dương không lớn hơn 666 xu.

**Kết quả: Đưa ra file văn bản COIN.OUT**

- Dòng 1: Ghi số cách trả số tiền ghi trong file dữ liệu
- Dòng 2: Ghi số đồng xu tối thiểu phải trả
- Các dòng tiếp theo, mỗi dòng ghi hai số a, b cách nhau ít nhất một dấu cách: cho biết sẽ có a đồng xu loại mệnh giá  $b^2$  trong phương án tối ưu (dùng ít đồng xu nhất)

**Ví dụ:**

COIN . INP	COIN . OUT
10	4 2 1 3 1 1

COIN . INP	COIN . OUT
19	10 3 1 1 2 3

COIN . INP	COIN . OUT
499	9508585 3 2 15 1 7

cuu duong than cong . com

### 035. HOÁN VỊ CHỮ CÁI

Cho một chuỗi S chỉ gồm các chữ cái in hoa,  $1 \leq \text{độ dài} \leq 9$ .

**Hãy lập chương trình trả lời hai câu hỏi sau:**

- Có bao nhiêu cách hoán vị các chữ cái của chuỗi S
- Liệt kê các hoán vị đó theo thứ tự từ điển.

**Dữ liệu:** Vào từ file văn bản PERMUTE.INP gồm 1 dòng chứa chuỗi S

**Kết quả:** Ghi ra file văn bản PERMUTE.OUT.

- Dòng 1: Ghi số lượng hoán vị tìm được (K)
- K dòng tiếp theo, mỗi dòng ghi một chuỗi hoán vị của chuỗi S (phải liệt kê theo đúng thứ tự từ điển)

PERMUTE . INP	PERMUTE . OUT
ABAB	6
	AABB
	ABAB
	ABBA
	BAAB
	BABA
	BBAA

cuu duong than cong . com

cuu duong than cong . com

### 036. DỰ TIỆC BÀN TRÒN

Có  $n$  nhà khoa học đánh số  $1, 2, \dots, n$  và 26 lĩnh vực khoa học ký hiệu  $A, B, C, \dots, Z$ . Thông tin về người thứ  $i$  được cho bởi một chuỗi ký tự  $S_i$  gồm các chữ cái in hoa thể hiện những lĩnh vực khoa học mà người đó biết.

Ví dụ:  $S_2 = 'ABCXYZ'$  cho biết nhà khoa học thứ 2 có hiểu biết về các lĩnh vực  $A, B, C, X, Y, Z$ .

Một lần cả  $n$  nhà khoa học đến dự một bữa tiệc. Chủ nhân của bữa tiệc định xếp  $n$  nhà khoa học ngồi quanh một bàn tròn, nhưng một vấn đề khiến chủ nhân rất khó xử là các nhà khoa học của chúng ta có hiểu biết xã hội tương đối kém, nên nếu như phải ngồi cạnh một ai đó không hiểu biết gì về các lĩnh vực của mình thì rất khó nói chuyện.

Vậy hãy giúp chủ nhân xếp  $n$  nhà khoa học ngồi quanh bàn tròn sao cho hai người bất kỳ ngồi cạnh nhau phải có ít nhất một lĩnh vực hiểu biết chung, để các nhà khoa học của chúng ta không những ăn ngon mà còn có thể trò chuyện rôm rả.

**Dữ liệu:** Vào từ file văn bản PARTY.INP. Trong đó:

- Dòng 1: Ghi số  $n$
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi chuỗi ký tự  $S_i$

**Kết quả:** Ghi ra file văn bản PARTY.OUT gồm  $n$  dòng.

- Dòng thứ  $i$  ghi nhà khoa học ngồi tại vị trí  $i$  của bàn (Các vị trí trên bàn tròn được đánh số từ 1 đến  $n$  theo chiều kim đồng hồ)

**Lưu ý:**

- $n \leq 20$
- Nếu có nhiều cách xếp thì chỉ cần chỉ ra một cách
- Nếu không có cách xếp thì ghi vào file PARTY.OUT một dòng: NO SOLUTION

**Ví dụ:**

PARTY . INP	PARTY . OUT
6	1
AV	3
DIQR	6
DV	2
CQ	4
AC	5
DR	

PARTY . INP	PARTY . OUT
10	1
AX	3
BI	2
ABTX	5
AS	6
IK	4
KS	8
BE	7
AB	9
EK	10
AK	

PARTY . INP	PARTY . OUT
6	NO SOLUTION
AB	
BC	
CD	
DE	
EF	
FG	

cuu duong than cong . com



### 037. TRÁO BÀI

Có  $2n$  lá bài, trên đó ghi lần lượt các số từ 1 đến  $2n$  (mỗi lá bài ghi một số và không có hai lá bài nào trùng số). Ban đầu các lá bài được xếp chồng nhau theo thứ tự từ lá bài ghi số 1 đến lá bài ghi số  $2n$  từ dưới lên trên.

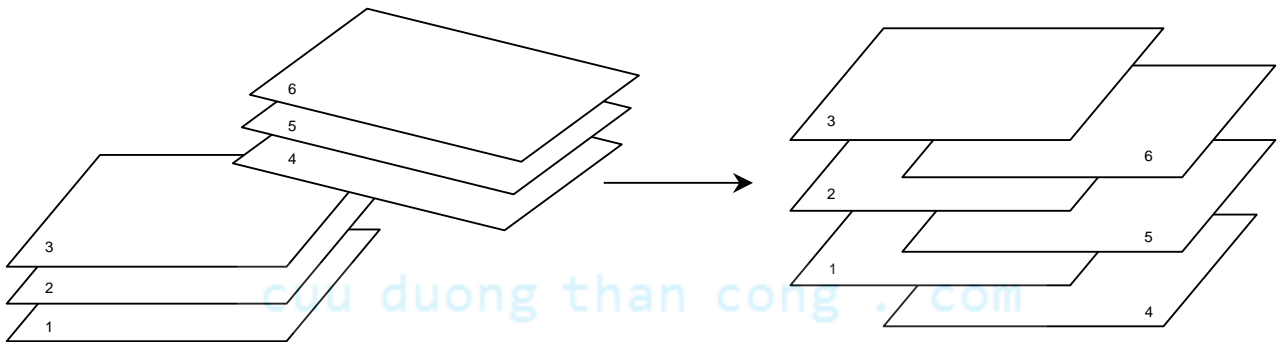
**Sau đó người ta tiến hành tráo các lá bài theo cách:**

- Nếu thứ tự các lá bài từ dưới lên đang là:  
(1, 2, 3 ..., n, n + 1, n + 2, n + 3, ..., 2n)
- Sẽ tráo thành thứ tự mới:  
(n + 1, 1, n + 2, 2, n + 3, 3, ..., 2n, n).

Bằng cách đổi vai trò các lá bài cho nhau, ta có thể hình dung ra được cách tráo trong các lần tiếp theo.

Ví dụ:  $n = 3$

Trạng thái ban đầu: (1, 2, 3, 4, 5, 6)  
 Sau lần tráo thứ nhất: (4, 1, 5, 2, 6, 3) (Xem hình vẽ)  
 Sau lần tráo thứ hai: (2, 4, 6, 1, 3, 5)  
 Sau lần tráo thứ ba: (1, 2, 3, 4, 5, 6)



Cách tráo bài này rất hay được sử dụng, tưởng rằng nó sẽ tạo ra một hoán vị hoàn toàn "vô tư" đối với các quân bài nhưng thực ra không phải như vậy, **sau một số hữu hạn lần tráo, tập bài lại trở về trạng thái ban đầu như chưa tráo.**

Ví dụ như bộ bài có 52 quân ( $n = 26$ ) thì chỉ qua 52 lần tráo là đâu vẫn hoàn đấy, hay bộ bài có 104 quân ( $n = 52$ ) thì chỉ qua có 12 lần tráo là sẽ trở về trạng thái ban đầu.

**Nhiệm vụ của bạn là khi biết được số  $n$  là một nửa số quân bài, hãy tính xem sau ít nhất bao nhiêu lần tráo thì tập bài sẽ trở về trạng thái ban đầu.**

**Dữ liệu:** Vào từ file văn bản CARD.INP chỉ gồm 1 dòng ghi số nguyên dương  $n$  ( $n \leq 10000$ )

**Kết quả:** Ghi ra file văn bản CARD.OUT cũng chỉ gồm 1 dòng ghi một số nguyên dương, là số lần tráo tối thiểu để tập bài trở lại trạng thái ban đầu.

**Ví dụ:**

CARD . INP	CARD . OUT
999	333

CARD . INP	CARD . OUT
26	52

CARD . INP	CARD . OUT
9875	9875

## 038. ĐỐI XỨNG HOÁ

### Định nghĩa:

- Một **xâu ký tự X** gọi là **chứa xâu ký tự Y** nếu như có thể xoá bớt một số ký tự trong xâu X để được xâu Y: Ví dụ: Xâu '1a2b3c45d' chứa xâu '12345'.
- Một xâu ký tự gọi là **đối xứng** nếu nó không thay đổi khi ta viết các ký tự trong xâu theo thứ tự ngược lại: Ví dụ: 'abcABADABAcba', 'MADAM' là các xâu đối xứng

Cho trước một xâu ký tự S có độ dài không quá 128.

Hãy tìm xâu ký tự T thoả mãn cả 3 điều kiện:

1. **Đối xứng**
2. **Chứa xâu S**
3. **Có ít ký tự nhất (có độ dài ngắn nhất)**

Lưu ý rằng với một xâu S, nếu có nhiều xâu T thoả mãn đồng thời 3 điều kiện trên thì chỉ cần cho biết một. Chẳng hạn với S = 'a\_101\_b' thì chọn T = 'ab\_101\_ba' hay T = 'ba\_101\_ab' đều đúng.

**Dữ liệu:** Vào từ file văn bản STR.INP chỉ gồm 1 dòng chứa xâu ký tự S

**Kết quả:** Ghi ra file văn bản STR.OUT cũng chỉ gồm 1 dòng ghi xâu ký tự T

**Ví dụ:** Một vài file dữ liệu vào và file kết quả tương ứng:

STR.INP	STR.OUT	STR.INP	STR.OUT
MADAM	MADAM	edbabcd	edcbabcde

STR.INP	STR.OUT
00_11_22_33_222_1_000	000_11_222_33_222_11_000

STR.INP	STR.OUT
abcdefg_hh_gfe_1_d_2_c_3_ba	ab_3_c_2_d_1_efg_hh_gfe_1_d_2_c_3_ba

cuu duong than cong . com

### 039. MẠNG MÁY TÍNH

Trên một nền phẳng với hệ tọa độ Decartes vuông góc đặt  $n$  máy tính và  $m$  cáp mạng nối chúng. Các máy tính được đánh số  $1, 2, \dots, n$  và các cáp mạng được đánh số  $1, 2, \dots, m$ . Vị trí của máy tính thứ  $i$  được cho bởi tọa độ  $(X_i, Y_i)$ , cáp mạng thứ  $j$  được cho nối giữa hai máy tính  $(p_j, q_j)$ . Hai máy tính bất kỳ có thể chuyển thông tin cho nhau bằng một trong hai cách: Truyền trực tiếp qua cáp nối chúng (nếu có) hoặc truyền qua một số máy trung gian.

**Yêu cầu:** Người ta muốn nối thêm các dây cáp mạng sao cho hai máy bất kỳ trong cả hệ thống  $n$  máy tính đều có thể chuyển thông tin cho nhau. Hãy chỉ ra cách nối thêm các dây cáp mạng sao cho tổng độ dài các dây cáp nối thêm là ít nhất, giả thiết rằng các dây cáp mạng được nối theo đường thẳng giữa hai máy.

**Dữ liệu:** Vào từ file văn bản NET.INP theo khuôn dạng sau:

Dòng	Nội dung
1	$n \ m$
2	$x_1 \ y_1$
3	$x_2 \ y_2$
...	...
$n + 1$	$x_n \ y_n$
$n + 2$	$p_1 \ q_1$
$n + 3$	$p_2 \ q_2$
...	...
$n + m + 1$	$p_m \ q_m$

**Kết quả:** Ghi ra file văn bản NET.OUT. Trong đó:

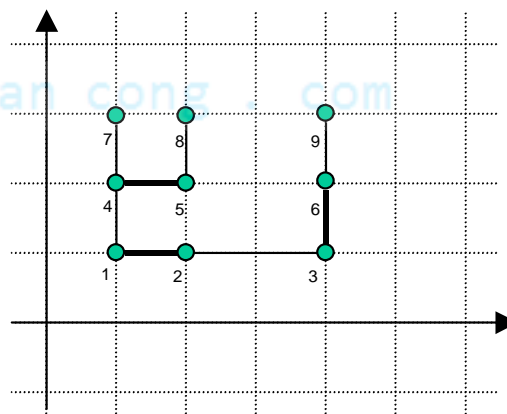
- Dòng 1: Ghi số nguyên dương  $K$  và số thực  $L$ .  $K$  là số dây cáp mạng phải nối thêm và  $L$  là tổng độ dài các dây cáp mạng nối thêm ( $L$  lấy chính xác tới 6 chữ số sau dấu chấm thập phân).
- $K$  dòng tiếp theo, mỗi dòng ghi số hiệu hai máy tính, cho biết sẽ đặt thêm dây cáp mạng nối hai máy tính đó

**Lưu ý:**

- Các số trên một dòng của Input/ Output file cách nhau ít nhất một dấu cách
- $1 \leq n \leq 1000$ ;  $0 \leq m \leq 10000$  và tọa độ của các máy tính là số nguyên có giá trị tuyệt đối không quá 1000.

**Ví dụ:**

NET . INP	NET . OUT
9 5	3 3.000000
1.0 1.0	1 2
2.0 1.0	4 5
4.0 1.0	3 6
1.0 2.0	
2.0 2.0	
4.0 2.0	
1.0 3.0	
2.0 3.0	
4.0 3.0	
1 4	
2 3	
4 7	
5 8	
6 9	



## 040. LẬT ĐÔ MI NÔ

Cho  $n$  quân đô-mi-nô xếp dựng đứng theo hàng ngang và được đánh số từ 1 đến  $n$ . Quân đô-mi-nô thứ  $i$  có số ghi ở ô trên là  $a_i$  và số ghi ở ô dưới là  $b_i$ . Xem hình vẽ:

1	2	3	4	5	6
1	1	4	4	0	6
6	3	1	1	6	1

Biết rằng  $1 \leq n \leq 100$  và  $0 \leq a_i, b_i \leq 6$  với  $\forall i: 1 \leq i \leq n$ .

Cho phép lật ngược các quân đô-mi-nô. Khi một quân đô-mi-nô thứ  $i$  bị lật, nó sẽ có số ghi ở ô trên là  $b_i$  và số ghi ở ô dưới là  $a_i$ .

**Vấn đề đặt ra là hãy tìm cách lật các quân đô-mi-nô sao cho chênh lệch giữa tổng các số ghi ở hàng ô trên và tổng các số ghi ở hàng ô dưới là tối thiểu. Nếu có nhiều phương án lật tốt như nhau, thì chỉ ra phương án phải lật ít quân nhất.**

**Dữ liệu:** Vào từ file văn bản DOMINO.INP. Trong đó:

- Dòng 1 ghi số  $n$
- Dòng 2 ghi  $n$  số  $a_1, a_2, \dots, a_n$  theo đúng thứ tự.
- Dòng 3 ghi  $n$  số  $b_1, b_2, \dots, b_n$  theo đúng thứ tự.

**Kết quả:** Ghi ra file văn bản DOMINO.OUT. Trong đó:

- Dòng 1: Ghi số quân Đô-mi-nô bị lật ( $C$ )
- Dòng 2: Ghi chỉ số của  $C$  quân Đô-mi-nô bị lật
- Dòng 3: Ghi độ chênh lệch giữa tổng các số hàng trên và tổng các số hàng dưới sau khi lật.

**Các số trên một hàng của Input/ Output File cách nhau ít nhất một dấu cách.**

**Ví dụ:**

DOMINO . INP	DOMINO . OUT
6	2
1 1 4 4 0 6	6 5
6 3 1 1 6 1	0

## 041. SỐ NHỊ PHÂN LỚN NHẤT

Xâu nhị phân là xâu ký tự chỉ gồm các chữ số 0 và 1. Người ta nói xâu nhị phân X là **xâu con** của xâu nhị phân Y nếu có thể xóa bớt một số ký tự trong xâu Y để được xâu X.

Ví dụ: Xâu '0101' là xâu con của xâu '000111000111'.

Lưu ý rằng nếu như xâu X = xâu Y thì xâu X cũng được coi là xâu con của xâu Y.

Nếu coi xâu nhị phân là biểu diễn nhị phân của một số nguyên thì số nguyên đó gọi là **trị số** của xâu nhị phân.

**Yêu cầu:** Cho trước hai xâu nhị phân A và B, hãy tìm một xâu nhị phân C là xâu con của cả A và B mà trị số của C là lớn nhất có thể được.

**Dữ liệu:** Nhập từ file văn bản BSTR.INP gồm 2 dòng:

- Dòng 1: Ghi xâu nhị phân A
- Dòng 2: Ghi xâu nhị phân B

**Kết quả:** Tạo file văn bản BSTR.OUT gồm 1 dòng ghi xâu nhị phân C tìm được.

**Ví dụ:**

BSTR.INP	BSTR.OUT
00000000101000101010 10000000000000010101	1000010101

BSTR.INP	BSTR.OUT
110011001100 001100110011	1100110011

cuu duong than cong . com

cuu duong than cong . com

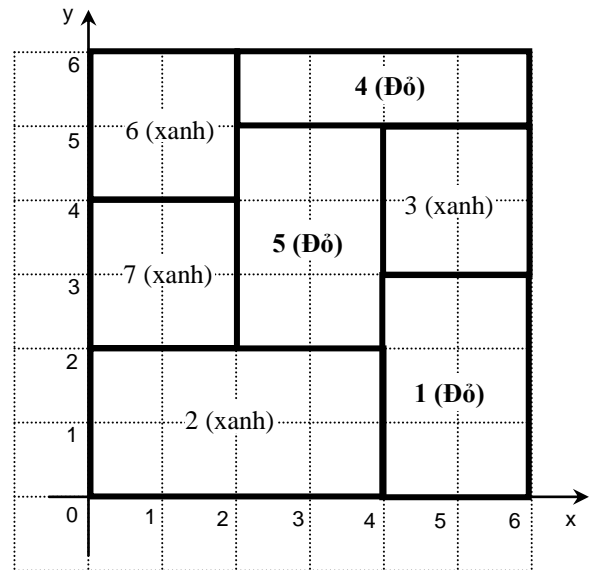
## 042. SƠN CÁC HÌNH CHỮ NHẬT

Một bảng hình chữ nhật phẳng đã được chia thành các miền hình chữ nhật **không giao nhau** và có cạnh song song với cạnh của bảng. Người ta muốn sơn các miền chữ nhật này, mỗi miền sẽ được sơn bằng một màu định sẵn.

Vì khi sơn có hiện tượng sơn chảy xuống phía dưới nên một **miền chữ nhật phía dưới chỉ được phép sơn khi mà các miền trên, có ảnh hưởng tới nó đã được sơn**.

Theo hình bên thì miền 2 chỉ được sơn sau khi miền 5 và miền 7 đã sơn xong. Nói một cách chính xác: Miền A bắt buộc phải sơn sau miền B nếu cả hai điều kiện sau thỏa mãn:

1. Hình chiếu của miền A và miền B trên trục hoành có ít nhất hai điểm chung
2. Tung độ tâm miền B lớn hơn tung độ tâm miền A



Để sơn tất cả các miền, người ta sử dụng một hệ thống chổi sơn đủ màu sắc, hai chổi sơn khác nhau có màu khác nhau. **Hãy tìm thứ tự sơn các miền chữ nhật sao cho số lần phải thay chổi là ít nhất.**

**Dữ liệu:** Vào từ file văn bản PAINT.INP. Trong đó:

- Dòng đầu tiên ghi số miền chữ nhật trong bảng (n)
- n dòng tiếp theo, Dòng thứ i ghi thông tin về miền thứ i gồm 5 số nguyên  $X_1$   $Y_1$   $X_2$   $Y_2$  C theo đúng thứ tự đó. ( $X_1$ ,  $Y_1$ ) là tọa độ đỉnh trái dưới, ( $X_2$ ,  $Y_2$ ) là tọa độ đỉnh phải trên, C là mã màu cần tô cho miền.

**Kết quả:** Ghi ra file văn bản PAINT.OUT. Trong đó

- Dòng 1: Ghi số lần thay chổi ít nhất (tính cả lần đầu tiên khi bắt đầu sơn)
- Dòng 2: Ghi số hiệu các miền chữ nhật theo đúng thứ tự sẽ tô.

Các số trên một dòng của Input/ Output file ghi cách nhau ít nhất một dấu cách.

Giới hạn:  $1 \leq n \leq 20$ ;  $1 \leq \text{mã màu} \leq 15$ ;  $0 \leq \text{các tọa độ} \leq 100$ ;

**Ví dụ:** Với hình vẽ trong bài, số 2 là mã màu đỏ và số 1 là mã màu xanh.

PAINT . INP					
7					
4	0	6	3	2	
0	0	4	2	1	
4	3	6	5	1	
2	5	6	6	2	
2	2	4	5	2	
0	4	2	6	1	
0	2	2	4	1	

PAINT . OUT					
3					
4	5	3	6	7	2 1

### 043. PHÂN HOẠCH TAM GIÁC

Xét một đa giác lồi với  $n$  cạnh, các đỉnh được đánh số theo thứ tự từ 1 tới  $n$ . Một bộ  $n - 3$  đường chéo **đôi một không cắt nhau** sẽ chia đa giác đã cho thành  $n - 2$  tam giác. Ta gọi bộ gồm  $n - 3$  đường chéo đó là một phép **tam giác phân** của đa giác lồi ban đầu.

**Trọng số** của một phép tam giác phân là **tổng độ dài các đường chéo** được sử dụng trong phép phân hoạch.

**Yêu cầu:**

*Cho trước một đa giác lồi, hãy tìm một phép tam giác phân nhỏ nhất (có trọng số nhỏ nhất)*

**Dữ liệu:** Vào từ file văn bản POLYGON.INP. Trong đó:

- Dòng 1: Ghi số đỉnh  $n$  của đa giác đã cho
- $n$  dòng tiếp theo, dòng thứ  $i$  gồm 2 số thực  $X_i, Y_i$  theo thứ tự là hoành độ và tung độ của đỉnh thứ  $i$ . (Các đỉnh được liệt kê theo đúng thứ tự gọi tên đa giác)

**Kết quả:** Ghi ra file văn bản POLYGON.OUT. Trong đó:

- Dòng 1: Ghi trọng số của phép tam giác phân nhỏ nhất
- $n - 3$  dòng tiếp theo, mỗi dòng ghi hai số nguyên dương  $i, j$  cho biết có sử dụng đường chéo nối đỉnh  $i$  với đỉnh  $j$  trong phép phân hoạch tìm được

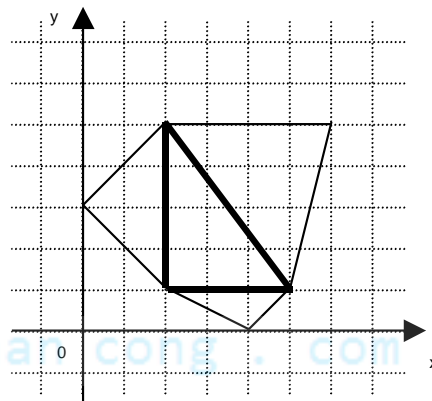
Các số trên một dòng của Input/Output file được ghi cách nhau ít nhất một dấu cách.

**Giới hạn:**

1.  $n$  nguyên dương,  $4 \leq n \leq 100$
2. Các tọa độ đỉnh là số thực:  $|X_i|, |Y_i| \leq 10^6$
3. Trọng số của phép tam giác phân nhỏ nhất được ghi dưới dạng số thực làm tròn lấy 6 chữ số sau dấu chấm thập phân.

**Ví dụ:**

POLYGON . INP	POLYGON . OUT
6	12.000000
4 0	2 6
5 1	2 4
6 4	4 6
2 4	
0 3	
2 1	



## 044. CÁC THÀNH PHẦN LIÊN THÔNG MẠNH

Cho đồ thị có hướng  $G = (V, E)$  gồm  $n$  đỉnh và  $m$  cung.

Một đồ thị con  $G'$  của  $G$  được gọi là một thành phần liên thông mạnh nếu hai điều kiện sau thoả mãn:

1. Hoặc  $G'$  chỉ gồm 1 đỉnh, hoặc với hai đỉnh  $i, j$  bất kỳ của  $G'$  luôn tồn tại đường đi từ đỉnh  $i$  tới đỉnh  $j$ .
2. Việc thêm vào  $G'$  một đỉnh bất kỳ sẽ làm hỏng tính chất 1

**Yêu cầu:** Cho biết số thành phần liên thông mạnh của đồ thị đã cho và liệt kê tất cả các thành phần liên thông mạnh.

**Dữ liệu:** Vào từ file văn bản GRAPH.INP, trong đó:

- Dòng 1: Ghi hai số  $n, m$
- $m$  dòng tiếp theo, mỗi dòng ghi hai số nguyên dương  $x, y$  thể hiện có cung nối từ đỉnh  $x$  tới đỉnh  $y$

**Kết quả:** Ghi ra file văn bản GRAPH.OUT, trong đó:

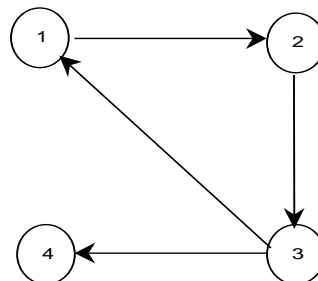
- Dòng 1: Ghi số thành phần liên thông mạnh ( $K$ )
- $K$  dòng tiếp theo, dòng thứ  $i$ , ghi các đỉnh thuộc thành phần liên thông mạnh thứ  $i$  tìm được

**Các số trên một dòng của Input/ Output file được ghi cách nhau ít nhất một dấu cách**

Giới hạn:  $1 \leq n \leq 1000$ ;  $1 \leq m \leq 3000$

**Ví dụ:**

GRAPH . INP	GRAPH . OUT
4 4	2
1 2	1 2 3
2 3	4
3 1	
3 4	



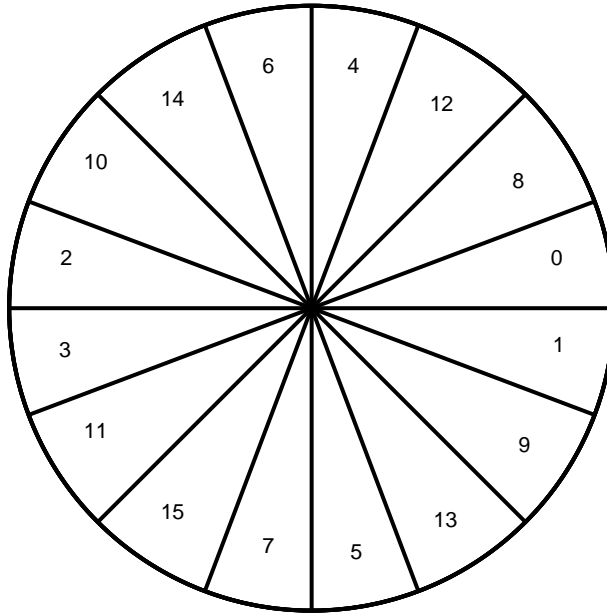


## 045. MÃ GRAY

Một hình tròn được chia làm  $2^n$  hình quạt đồng tâm, các hình quạt được đánh số từ 1 tới  $2^n$  theo chiều kim đồng hồ. Hãy chỉ ra **một** cách xếp tất cả số từ 0 tới  $2^n - 1$  vào các hình quạt, mỗi số vào một hình quạt sao cho bất cứ hai số nào ở hai hình quạt cạnh nhau đều chỉ khác nhau đúng 1 bit trong biểu diễn nhị phân của nó.

Ví dụ: Với  $n = 4$ :

0 = 0000  
1 = 0001  
2 = 0010  
3 = 0011  
4 = 0100  
5 = 0101  
6 = 0110  
7 = 0111  
8 = 1000  
9 = 1001  
10 = 1010  
11 = 1011  
12 = 1100  
13 = 1101  
14 = 1110  
15 = 1111



**Dữ liệu:** Nhập từ bàn phím số nguyên dương  $n$ . Giới hạn ( $1 \leq n \leq 20$ ).

**Kết quả:** Ghi ra File (of LongInt) GRAYCODE.OUT gồm  $2^n$  số nguyên kiểu LongInt theo đúng thứ tự từ số ghi trên hình quạt 1 tới số ghi trên hình quạt  $2^n$ .

## 046. DỰ ÁN XÂY CẦU

Trong một khu công viên nước có  $n$  hòn đảo nhỏ và một số cầu nối giữa chúng. Giả thiết rằng các cầu được nối theo đường thẳng.

Hai câu hỏi đặt ra là:

1. Có tồn tại một đường đi qua tất cả các đảo mỗi đảo đúng một lần hay không ?
2. Nếu không tồn tại đường đi như vậy, hãy chỉ ra các xây thêm các cây cầu để thực hiện được điều đó sao cho tổng độ dài những cây cầu xây thêm là ít nhất.

**Dữ liệu:** Vào từ file văn bản WPARK.INP

- Dòng 1: Ghi số đảo  $n$  ( $\leq 16$ ) và số cầu đã có  $m$
- $n$  dòng tiếp theo, dòng thứ  $i$  gồm 2 số thực  $x[i]$   $y[i]$  là toạ độ của hòn đảo  $i$ .
- $m$  dòng tiếp theo, dòng thứ  $j$  ghi số hiệu hai đảo tương ứng với chiếc cầu thứ  $j$ .

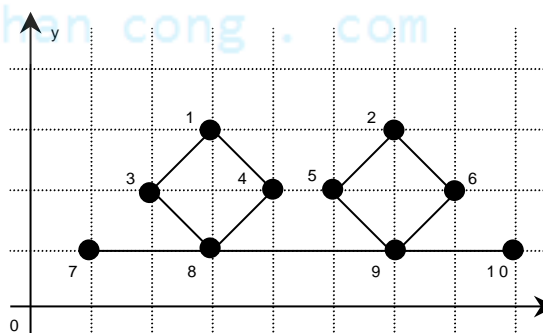
**Kết quả:** Ghi ra file văn bản WPARK.OUT

- Dòng 1: ghi số  $k$  là số cầu cần xây thêm và số thực  $T$  (lấy tới 6 chữ số sau dấu chấm thập phân) là tổng độ dài các cây cầu xây thêm
- $k$  dòng tiếp theo, mỗi dòng ghi số hiệu hai đảo tương ứng với một cây cầu xây thêm
- Dòng  $k + 2$  ghi số hiệu các đảo trên đường đi tìm được (sau khi đã xây thêm cầu)

Các số trên một dòng của Input/ Output file được ghi cách nhau ít nhất một dấu cách.

**Ví dụ:**

WPARK . INP	WPARK . OUT
10 11	1 1.000000
3.0 3.0	4 5
6.0 3.0	7 8 3 1 4 5 2 6 9 10
2.0 2.0	
4.0 2.0	
5.0 2.0	
7.0 2.0	
1.0 1.0	
3.0 1.0	
6.0 1.0	
8.0 1.0	
1 3	
1 4	
2 5	
2 6	
3 8	
4 8	
5 9	
6 9	
7 8	
8 9	
9 10	



## 047. BẢO TỒN ĐỘNG VẬT HOANG DÃ

Một khu bảo tồn động vật có  $n$  địa điểm và các đường đi hai chiều nối các địa điểm đó, địa điểm thứ  $i$  có nhiệt độ là  $t_i$ , giữa hai địa điểm bất kỳ có nhiều nhất là một đường đi nối chúng.

Người ta muốn di chuyển một loài động vật quý hiếm từ địa điểm A tới địa điểm B, tuy nhiên nếu chênh lệch về nhiệt độ giữa hai địa điểm liên tiếp trên đường đi là quá cao thì loài động vật này rất có thể bị chết.

**Yêu cầu:** *Hãy chỉ ra một hành trình mà độ lệch nhiệt độ lớn nhất giữa hai địa điểm liên tiếp bất kỳ trên đường đi là cực tiểu.*

**Dữ liệu:** Vào từ file văn bản MOVE.INP

- Dòng 1: Chứa ba số  $n, A, B$  ( $2 \leq n \leq 200; A \neq B$ )
- Dòng 2: Chứa  $n$  số tự nhiên  $t_1, t_2, \dots, t_n$  ( $\forall i: 0 \leq t_i \leq 20000$ )
- Các dòng tiếp theo, mỗi dòng chứa hai số nguyên dương  $u, v$  cho biết giữa hai địa điểm  $u$  và  $v$  có đường đi nối chúng.

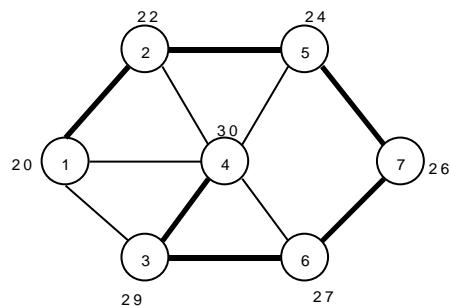
**Kết quả:** Ghi ra file văn bản MOVE.OUT

- Dòng 1: Ghi độ lệch nhiệt độ lớn nhất giữa hai địa điểm liên tiếp bất kỳ trên đường đi tìm được, nếu không tồn tại đường đi thì dòng này ghi số -1.
- Trong trường hợp tìm được đường đi thì dòng 2 ghi hành trình tìm được, bắt đầu từ địa điểm A, tiếp theo là những địa điểm đi qua, kết thúc là địa điểm B. Các địa điểm phải được liệt kê theo đúng thứ tự đi qua trên hành trình

**Các số trên một dòng của Input/ Output file được ghi cách nhau ít nhất một dấu cách.**

Ví dụ:

MOVE . INP	MOVE . OUT
7 1 4	2
20 22 29 30 24 27 26	1 2 5 7 6 3 4
1 2	
1 3	
1 4	
2 4	
2 5	
3 4	
3 6	
4 5	
4 6	
5 7	
6 7	



## 048. PHÁ TƯỜNG

Có một toà lâu đài hình chữ nhật với hai cạnh là  $m, n$  nguyên dương không lớn hơn 50. Lâu đài được chia thành các ô vuông đơn vị. Các dòng ô vuông được đánh số từ 1 tới  $m$  từ trên xuống dưới, trên mỗi dòng, các ô được đánh số theo thứ tự từ 1 tới  $n$  từ trái qua phải. Quanh mỗi ô có thể có từ 0 tới 4 bức tường, tuy nhiên tình trạng có tường tại các ô kề cạnh là không mâu thuẫn nhau.

Để thể hiện tình trạng tường quanh một ô, ta gán cho mỗi ô một số nguyên, mà trong biểu diễn nhị phân của số nguyên đó:

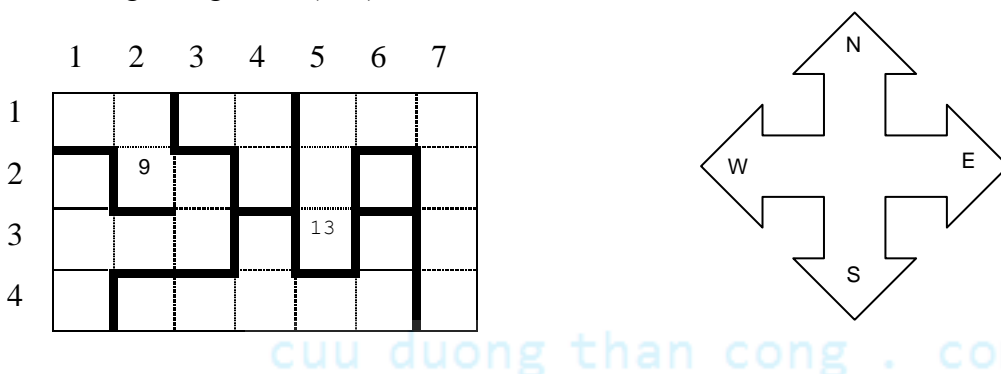
- Bít 0 (Bít đơn vị) bằng 1 hay 0 tùy theo ô đó có tường hay không có tường hướng Tây
- Bít 1 bằng 1 hay 0 tùy theo ô đó có tường hay không có tường hướng Bắc
- Bít 2 bằng 1 hay 0 tùy theo ô đó có tường hay không có tường hướng Đông
- Bít 3 bằng 1 hay 0 tùy theo ô đó có tường hay không có tường hướng Nam

Quanh lâu đài có tường bao bọc.

Ví dụ trong hình vẽ dưới, ta có một lâu đài  $4 \times 7$ .

Tình trạng tường của ô (2, 2) được thể hiện bởi số 9 = 1001

Tình trạng tường của ô (3, 5) được thể hiện bởi số 13 = 1101



Lâu đài được chia thành các phòng, các phòng phân cách nhau bởi các bức tường. Hãy lập chương trình trả lời các câu hỏi sau:

1. Cho biết lâu đài có bao nhiêu phòng
2. Cho biết số ô của phòng rộng nhất
3. Hãy tìm cách phá đi một và chỉ một bức tường để được một phòng rộng nhất có thể

**Dữ liệu:** Vào từ file văn bản DWALL.INP

- Dòng 1: Ghi hai số  $m, n$
- $m$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số nguyên, số thứ  $j$  thể hiện tình trạng tường quanh ô  $(i, j)$

**Kết quả:** Ghi ra file văn bản DWALL.OUT

- Dòng 1: Ghi số phòng
- Dòng 2: Ghi số ô của phòng rộng nhất
- Dòng 3: Ghi hai số  $P, Q$  và ký tự  $c \in \{W, N, E, S\}$  với ý nghĩa phá tường ở hướng  $c$  của ô  $(P, Q)$
- Dòng 4: Ghi số ô của phòng rộng nhất thu được sau khi phá tường

Các số trên một dòng của Input/Output file được ghi cách nhau ít nhất một dấu cách

**Ví dụ:**

DWALL . INP							
4	7						
11	06	11	06	03	10	06	
07	09	06	13	05	14	05	
01	10	12	07	13	07	05	
13	11	10	08	10	12	13	

DWALL . OUT			
5			
9			
3	2	S	
16			

## 049. TRUYỀN TIN TRÊN MẠNG

Trong một mạng gồm  $N$  máy tính đánh số từ 1 đến  $N$ . Sơ đồ nối mạng được cho bởi  $m$  kênh nối trực tiếp giữa một số cặp máy trong mạng. Biết chi phí truyền một đơn vị thông tin theo mỗi kênh nối của mạng.

Người ta cần chuyển một bức thông điệp từ máy  $S$  đến máy  $D$  ( $S \neq D$ ). Để đảm bảo an toàn, người ta muốn chuyển bức thông điệp này theo hai đường truyền tin khác nhau (tức là không có kênh nào của mạng được sử dụng trong cả hai đường truyền tin). Chi phí của một đường truyền tin được hiểu là tổng chi phí trên các kênh của nó. Chi phí truyền thông điệp bằng tổng chi phí của hai đường truyền.

**Yêu cầu:** Giả sử bức thông điệp có độ dài là 1 đơn vị thông tin, hãy tìm cách truyền thông điệp từ  $s$  đến  $t$  sao cho chi phí truyền thông điệp là nhỏ nhất

**Dữ liệu:** Nhập từ file văn bản MESSAGE.INP với cấu trúc như sau:

- Dòng đầu tiên ghi bốn số  $n, m, S, D$  ( $n \leq 100$ );
- Mỗi dòng thứ  $i$  trong số  $m$  dòng tiếp theo ghi thông tin về kênh nối thứ  $i$  của mạng gồm ba số  $a_i, b_i, c_i$ , trong đó  $a_i, b_i$  là chỉ số của hai máy tương ứng với kênh này và  $c_i$  (nguyên dương  $\leq 200$ ) là chi phí để truyền một đơn vị thông tin từ máy  $a_i$  đến máy  $b_i$  (và ngược lại) theo kênh này ( $i=1,2,\dots,m$ ).

**Kết quả:** Ghi ra file văn bản MESSAGE.OUT theo cấu trúc sau:

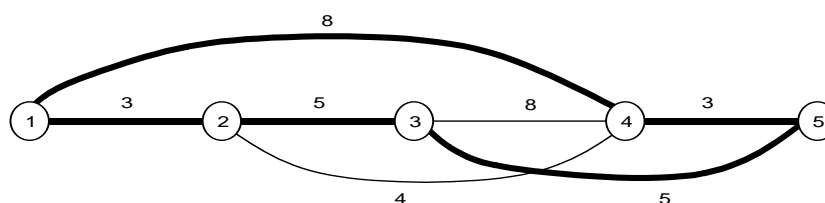
- Dòng đầu tiên ghi chi phí truyền thông điệp theo cách truyền tin tìm được.
- Dòng thứ hai ghi đường truyền tin thứ nhất dưới dạng dãy có thứ tự các máy, bắt đầu từ máy  $S$  và kết thúc ở máy  $D$ .
- Dòng thứ ba ghi đường truyền tin thứ hai dưới dạng dãy có thứ tự các máy bắt đầu từ máy  $S$  và kết thúc ở máy  $D$ .

Nếu không tồn tại cách truyền thì chỉ cần ghi vào file MESSAGE.OUT một dòng:

NO SOLUTION

Các số trên một dòng của Input/ Output file ghi cách nhau ít nhất một dấu cách.

Ví dụ:



MESSAGE . INP			
5	7	1	5
1	2	3	
1	4	8	
2	3	5	
2	4	4	
3	5	5	
4	3	8	
4	5	3	

MESSAGE . OUT			
24			
1	2	3	5
1	4	5	

## 050. HÌNH VUÔNG CỰC ĐẠI

Cho một bảng kích thước  $m \times n$ , được chia thành lưới ô vuông đơn vị  $m$  dòng  $n$  cột. Trên các ô của bảng ghi số 0 hoặc 1. Các dòng của bảng được đánh số 1, 2, ...,  $m$  theo thứ tự từ trên xuống dưới và các cột của bảng được đánh số 1, 2, ...,  $n$  theo thứ tự từ trái qua phải.

**Hãy tìm một hình vuông gồm các ô của bảng thoả mãn các điều kiện sau:**

1. **Hình vuông là đồng nhất:** tức là các ô thuộc hình vuông đó phải ghi các số giống nhau (0 hoặc 1)
2. **Cạnh hình vuông song song với cạnh bảng.**
3. **Kích thước hình vuông là lớn nhất có thể.**

**Dữ liệu:** Vào từ file văn bản SQUARE.INP

- Dòng 1: Ghi hai số  $m, n$
- $m$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số mà số thứ  $j$  là số ghi trên ô  $(i, j)$  của bảng

**Kết quả:** Ghi ra file văn bản SQUARE.OUT

- Dòng 1: Ghi kích thước cạnh hình vuông tìm được
- Dòng 2: Ghi 4 số nguyên  $r_1, c_1, r_2, c_2$ . ở đây  $(r_1, c_1)$  là chỉ số hàng và chỉ số cột của ô thuộc góc trên bên trái,  $(r_2, c_2)$  là chỉ số hàng và chỉ số cột của ô thuộc góc dưới bên phải hình vuông tìm được.

**Các số trên một dòng của Input/ Output file ghi cách nhau ít nhất một dấu cách.**

Ví dụ:

SQUARE . INP	SQUARE . OUT
11 13	7
0 0 0 0 0 1 0 0 0 0 0 0 0	3 3 9 9
0 0 0 0 1 1 1 0 0 0 0 0 0	
0 0 1 1 1 1 1 1 1 0 0 0 0	
0 0 1 1 1 1 1 1 1 0 0 0 0	
0 1 1 1 1 1 1 1 1 1 0 0 0	
1 1 1 1 1 1 1 1 1 1 1 0 0	
0 1 1 1 1 1 1 1 1 1 0 0 0	
0 0 1 1 1 1 1 1 1 0 0 0 0	
0 0 1 1 1 1 1 1 1 0 0 0 0	
0 0 0 0 1 1 1 0 0 0 0 1 1	
0 0 0 0 0 1 0 0 0 0 0 1 1	

## 051. ĐOÀN XE QUA CẦU

Cho một đoàn xe gồm  $n$  chiếc đi trên một đường một chiều và đoàn xe đã được bố trí theo thứ tự từ 1 đến  $n$ . Mỗi một xe trong đoàn có vận tốc là  $v_i$  và trọng lượng  $w_i$ .

Khi đi qua một chiếc cầu có trọng tải giới hạn là  $P$  thì đoàn xe phải chia thành các nhóm sao cho tổng trọng lượng của mỗi nhóm không quá  $P$  (Lưu ý rằng không được đảo thứ tự đoàn xe). Các nhóm phải đi tuần tự có nghĩa là nhóm thứ  $i$  chỉ được khởi hành khi mà toàn bộ xe của nhóm thứ  $i - 1$  đã qua cầu. Giả thiết rằng  $P > w_i$  với  $\forall i: 1 \leq i \leq n$ .

Rõ ràng khi đó thời gian để một nhóm xe qua cầu phụ thuộc vào xe chậm nhất trong nhóm đó nếu coi như chiều dài cũng như khoảng cách của các xe là không đáng kể.

**Hãy tìm cách chia đoàn xe thành các nhóm sao cho thời gian mà đoàn xe sang được cầu là nhỏ nhất có thể được.**

**Dữ liệu:** Vào từ file văn bản CARGROUP.INP

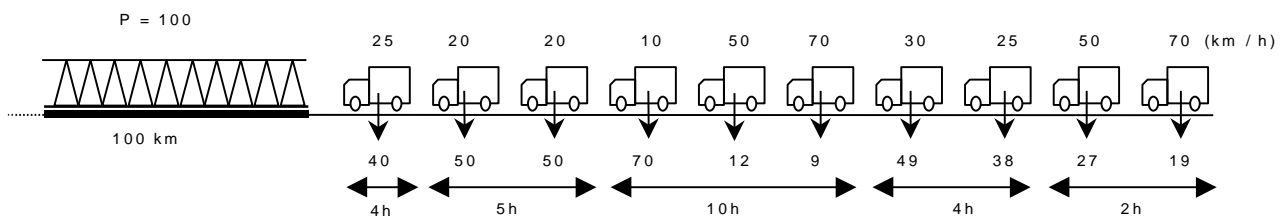
- Dòng đầu là 3 số nguyên dương  $n$ ,  $P$  và  $L$  ( $n, P, L \leq 1000$ ) thể hiện cho số xe, trọng lượng giới hạn của cầu và độ dài của cầu.
- Dòng thứ  $i$  trong  $n$  dòng kế tiếp gồm 2 số nguyên dương  $w_i$  và  $v_i$  ( $w_i, v_i \leq 100$ )

**Kết quả:** Ghi ra file văn bản CARGROUP.OUT

- Dòng đầu ghi một số thực là tổng thời gian nhỏ nhất để xe qua cầu, cho phép làm tròn lấy 2 chữ số sau dấu chấm thập phân.
- Dòng kế tiếp gồm các số  $x_1, x_2, \dots, x_k$  thể hiện: nhóm 1 gồm các xe từ 1 đến xe thứ  $x_1$ , nhóm 2 gồm các xe thứ  $x_1+1$  đến xe thứ  $x_2$ , ..., nhóm  $k$  từ xe thứ  $x_{k-1}+1$  tới  $x_k$

**Các số trên một dòng của Input / Output file ghi cách nhau ít nhất một dấu cách.**

**Ví dụ:**



CARGROUP . INP	CARGROUP . OUT
10 100 100	25.00
40 25	1 3 6 8 10
50 20	
50 20	
70 10	
12 50	
09 70	
49 30	
38 25	
27 50	
19 70	

## 052. SỐ LƯỢNG

Cho số nguyên dương  $n$  ( $n \leq 2\,000\,000\,000$ ). Hãy xác định xem trong phạm vi từ 1 tới  $n$  có bao nhiêu số mà trong dạng biểu diễn nhị phân của nó có đúng  $K$  chữ số 0 có nghĩa.

Ví dụ:  $n = 18, k = 3$  có 3 số:

1.  $8 = 1000$
2.  $17 = 10001$
3.  $18 = 10010$

**Dữ liệu:** Vào từ file văn bản NUMBER.INP, gồm một dòng chứa hai số nguyên  $N$  và  $K$  cách nhau một dấu cách.

**Kết quả:** Đưa ra file NUMBER.OUT, ghi số lượng các số tìm được

Ví dụ:

NUMBER . INP	NUMBER . OUT
18 3	3

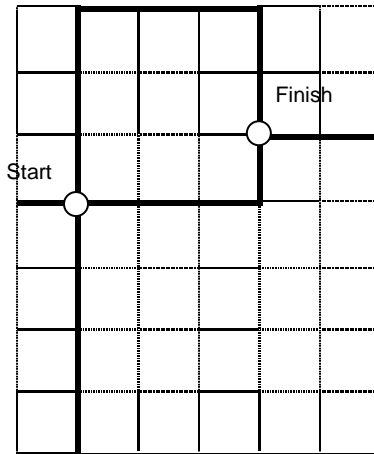
cuu duong than cong . com

cuu duong than cong . com



### 053. THẨM HIỂM LÒNG ĐẤT

Một nhà khảo cổ nghiên cứu những di sản văn hoá cổ đại ở một thành phố bị chôn vùi dưới lòng đất. Để thám hiểm thành phố đó, nhà khảo cổ của chúng ta buộc phải đào các đường ngầm. Bắt đầu tại vị trí xuất phát, ông ta đào theo một trong 4 hướng Đông (E), Tây (W), Nam (S), Bắc (N), mỗi lần đào một đơn vị độ dài. Sau đó có thể đào tiếp theo hướng đó hoặc đổi hướng theo một trong 4 hướng trên. Giả sử rằng đường kính của đường ngầm đào được là không đáng kể. Để tránh bị lạc, ông ta ghi lại vào file văn bản MAP.INP trong máy tính xách tay của mình một trong 4 ký tự E, W, S, N tương ứng với một trong bốn hướng mà ông ta sẽ đào tới mỗi lần. Ví dụ với điểm xuất phát và quy trình đào hầm dưới đây, sơ đồ các đường ngầm sẽ là:



Sau khi đã khảo sát xong, nhà khảo cổ muốn quay trở lại điểm xuất phát bằng đường hầm đã đào. Hãy dựa vào thông tin trong máy tính xách tay của nhà khảo cổ để chỉ cho ông ta đường đi ngắn nhất quay trở lại.

**Dữ liệu:** Vào từ file văn bản MAP.INP của nhà khảo cổ gồm 1 dòng không quá 5000 ký tự  $\in \{E, W, N, S\}$

**Kết quả:** Ghi ra file văn bản MAP.OUT gồm 1 dòng chứa các ký tự chỉ hướng đi dẫn về nơi xuất phát.

Ví dụ:

MAP . INP	MAP . OUT
EEEENNNWWWSSSSSSSEEEEENNNNNWW	SWWWW

cuu duong than cong . com

## 054. THỨ TỰ TỪ ĐIỂN

*Một bảng danh mục gồm các từ đã được sắp xếp theo một trật tự từ điển nào đấy (không nhất thiết là từ điển thông thường). Yêu cầu từ bảng danh mục, hãy khôi phục lại trật tự từ điển đã dùng.*

- **Dữ liệu** vào được cho bởi file văn bản NOTE.INP. Dòng đầu là số lượng từ, các dòng tiếp, (theo thứ tự) mỗi dòng là một từ trong bảng danh mục. Giả thiết rằng mỗi từ đều không quá 20 ký tự được lấy trong bảng chữ cái nhỏ tiếng Anh (từ 'a' đến 'z'). Số lượng từ trong bảng danh mục không quá 10000.
- **Kết quả** đưa ra file văn bản NOTE.OUT gồm một dòng là xâu gồm các chữ cái đã xuất hiện trong bảng danh mục. Các chữ cái trong xâu viết liền nhau và theo thứ tự phù hợp với trật tự từ điển đã dùng.

**Ví dụ:**

NOTE . INP	NOTE . OUT
10	gsvqnx
svxngqqnsnvqv	
snnng	
qsqsqvgsqq	
qqns	
qnvq	
nsxnxnvsqsvvs	
nqq	
nn	
xsgvsgggqvsgqsgv	
xxgxxggsvnxsnxsnqq	

## 055. DÃY LỆCH

Cho hai dãy số nguyên:

- $A = (a_1, a_2, \dots, a_n)$
- $B = (b_1, b_2, \dots, b_n)$

( $n \leq 100$ ;  $-10000 \leq a_i, b_j \leq 10000$  với  $\forall i, j : 1 \leq i, j \leq n$ )

**Hãy tìm một hoán vị  $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$  của dãy số  $(1, 2, \dots, n)$**

**Để cực tiểu hoá biểu thức:**

$$F(\sigma) := |1 - a_{\sigma_1}| + |b_{\sigma_1} - a_{\sigma_2}| + |b_{\sigma_2} - a_{\sigma_3}| + \dots + |b_{\sigma_{n-1}} - a_{\sigma_n}| + |b_{\sigma_n} - 1|$$

**Dữ liệu:** Vào từ file văn bản SLANTING.INP

- Dòng 1: Ghi số  $n$
- $n$  dòng tiếp theo, Dòng thứ  $i$  ghi 2 số nguyên  $a_i$  và  $b_i$  cách nhau ít nhất 1 dấu cách

**Kết quả:** Ghi ra file văn bản SLANTING.OUT

- Dòng 1: Ghi giá trị cực tiểu  $F(\sigma)$  tìm được
- $n$  Dòng tiếp theo, dòng thứ  $i$  ghi giá trị  $\sigma_i$

**Ví dụ:**

cuu duong than cong . com

cuu duong than cong . com

## 056. RÚT GỌN DÃY SỐ

Cho dãy gồm  $n$  số nguyên dương  $a = (a_1, a_2, \dots, a_n)$ . Trên dãy số này ta có thể thực hiện phép rút gọn tại vị trí  $i$ :

***R(i): thay hai số hạng liên tiếp  $a_i$  và  $a_{i+1}$  bằng hiệu của chúng  $a_i - a_{i+1}$ .***

Sau  $n - 1$  lần rút gọn, với dãy  $a$ , ta thu được duy nhất một số nguyên.

**Ví dụ:** Thực hiện lần lượt các phép rút gọn 2, 3, 2 và 1 đối với dãy số (12, 10, 4, 3, 5) ta sẽ thu được kết quả như sau:

1. Ban đầu: (12, 10, 4, 3, 5)
2. Rút gọn R(2): (12, 6, 3, 5)
3. Rút gọn R(3): (12, 6, -2)
4. Rút gọn R(2): (12, 8)
5. Rút gọn R(1): (4)

Yêu cầu cho dãy số  $a = (a_1, a_2, \dots, a_n)$  và số  $T$ , hãy tìm thứ tự thực hiện  $N - 1$  phép rút gọn đối với dãy đã cho để thu được  $T$ .

**Dữ liệu:** Vào từ file văn bản SUBTRACT.INP

- Dòng đầu tiên chứa hai số  $n$  và  $T$  cách nhau một dấu cách ( $1 \leq n \leq 100$ ;  $-10000 \leq T \leq 10000$ )
- Dòng thứ  $i$  trong số  $n$  dòng tiếp theo ghi số  $a_i$ . ( $1 \leq a_i \leq 100$ ).

**Kết quả:** Ghi ra file văn bản SUBTRACT.OUT

- Gồm  $n - 1$  dòng, dòng thứ  $i$  ghi vị trí thực hiện phép rút gọn thứ  $i$ .

Giả thiết rằng các dữ liệu đều có ít nhất một lời giải

**Ví dụ:**

SUBTRACT . INP	
4	5
10	
2	
5	
2	

SUBTRACT . OUT	
3	
1	
1	

## 057. BUÔN TIỀN

Một người làm việc ở một ngân hàng ngoại tệ theo dõi tỉ giá hối đoái phát hiện ra là: Nếu khôn khéo, thì từ một lượng ngoại tệ ban đầu, nhờ chuyển đổi sang các loại ngoại tệ khác, anh ta có thể thu được lợi nhuận đáng kể.

Ví dụ: Nếu anh ta có 1 USD và tỉ giá hối đoái giữa các ngoại tệ như sau:

- 1 USD = 0.7 bảng Anh
- 1 bảng Anh = 9.5 Franc Pháp
- 1 Franc Pháp = 0.16 USD

Khi đó với 1 USD anh ta có thể mua được  $0.7 * 9.5 * 0.16 = 1.064$  USD nhờ việc chuyển đổi tiền qua bảng Anh, rồi từ bảng Anh sang Franc Pháp, và cuối cùng lại quay về USD. Nhờ đó mỗi USD đã đem lại cho anh ta lợi nhuận là 0.064USD.

Giả sử trong nhà băng quản lý n loại ngoại tệ đánh số 1, 2, ..., n. Biết bảng tỉ giá hối đoái  $R[i, j]$  ( $1 \leq i, j \leq n$ ). (Tức là 1 đơn vị ngoại hối i mua được  $R[i, j]$  đơn vị ngoại hối j). Cần xác định xem có cách đổi tiền đem lại lợi nhuận hay không ?

**Dữ liệu:** Vào từ file văn bản MONEY.INP

- Dòng đầu tiên chứa số n ( $n \leq 100$ )
- Dòng thứ i trong số n dòng tiếp theo chứa n số thực dương  $R[i, 1], R[i, 2], \dots, R[i, n]$ .

**Kết quả:** Ghi ra file văn bản MONEY.OUT

Dòng đầu tiên ghi YES hoặc NO tương ứng với việc có hoặc không có cách đổi tiền sinh lợi nhuận. Nếu dòng đầu tiên là YES thì dòng thứ hai ghi hai số u và s. Trong đó u là loại tiền xuất phát, còn s là lợi nhuận thu được nhờ cách đổi 1 đơn vị tiền u. Dòng thứ ba ghi trình tự cần tiến hành đổi tiền để thu lại được lợi nhuận bắt đầu từ loại tiền xuất phát.

**Các số trên một dòng của Input/Output File được ghi cách nhau ít nhất một dấu cách. Lợi nhuận (nếu có) trong Output File có thể chỉ cần làm tròn giữ lại 6 chữ số sau dấu chấm thập phân.**

**Ví dụ:**

MONEY . INP					
5					
1.00	1.10	0.83	0.81	0.85	
0.83	1.00	0.86	1.09	0.81	
0.89	0.84	1.00	0.83	1.02	
0.84	0.83	1.01	1.00	0.84	
1.09	0.84	0.87	0.90	1.00	

MONEY . OUT	
YES	
1	0.007160
1 2 4	

## 058. DẪY NGOẶC

Một dãy dấu ngoặc hợp lệ là một dãy các ký tự "(" và ")" được định nghĩa như sau:

- Dãy rỗng là một dãy dấu ngoặc hợp lệ độ sâu 0
- Nếu A là dãy dấu ngoặc hợp lệ độ sâu k thì (A) là dãy dấu ngoặc hợp lệ độ sâu k + 1
- Nếu A và B là hai dãy dấu ngoặc hợp lệ với độ sâu lần lượt là p và q thì AB là dãy dấu ngoặc hợp lệ độ sâu là max(p, q)

Độ dài của một dãy ngoặc là tổng số ký tự "(" và ")"

**Ví dụ:** Có 5 dãy dấu ngoặc hợp lệ độ dài 8 và độ sâu 3:

(( ( ) ( ) ) )  
 (( ( ) ) ( ) )  
 (( ( ) ) ) ( )  
 ( ( ) ( ( ) ) )  
 ( ) ( ( ( ) ) )

**Bài toán đặt ra là khi cho biết trước hai số nguyên dương n và k. Hãy cho biết có bao nhiêu dãy ngoặc hợp lệ có độ dài là n và độ sâu là k. Nếu có không quá 100 dãy thì hãy liệt kê hết các dãy, nếu có nhiều hơn 100 dãy thì hãy chỉ ra 100 dãy ngoặc phân biệt.**

**Dữ liệu:** Vào từ file văn bản NGOAC.INP gồm 1 dòng ghi hai số nguyên dương n và k cách nhau một dấu cách ( $n \leq 64$ ,  $k \leq 32$ ).

**Kết quả:** Ghi ra file văn bản NGOAC.OUT

- Dòng 1: Ghi số C là số lượng dãy ngoặc hợp lệ có độ dài là n và độ sâu là k.
- Nếu  $C \leq 100$ , thì C dòng tiếp theo mỗi dòng ghi một dãy ngoặc tìm được. Nếu  $C > 100$ , thì 100 dòng tiếp theo mỗi dòng ghi một dãy ngoặc. Các dãy ngoặc được liệt kê đôi một khác nhau.

**Ví dụ:**

NGOAC . INP	NGOAC . OUT	NGOAC . INP	NGOAC . OUT
8 3	5 (( ( ) ( ) ) ) (( ( ) ) ( ) ) (( ( ) ) ) ( ) ( ( ) ( ( ) ) ) ( ) ( ( ( ) ) )	10 2	15 ( ( ) ( ) ( ) ( ) ) ( ( ) ( ) ( ) ) ( ) ( ( ) ( ) ) ( ( ) ) ( ( ) ( ) ) ( ) ( ) ( ( ) ( ) ) ( ) ( ) ( ( ) ( ) ( ) ( ) ) ( ( ) ( ) ( ) ) ( ) ( ( ) ( ) ( ) ) ( ) ( ( ) ( ) ( ) ) ( ) ( ( ) ( ) ( ) ) ( ) ( ( ) ( ) ( ) ) ( ) ( ( ) ( ) ( ) ) ( ) ( ( ) ( ) ( ) ) ( ) ( ( ) ( ) ( ) ) ( ) ( ( ) ( ) ( ) ) ( ) ( ( ) ( ) ( ) ) ( )

cuu duong than cong . com

## 059. THĂNG BỜM VÀ PHÚ ÔNG

Bờm thắng phú ông trong một cuộc đánh cược và buộc phú ông phải đãi rượu. Phú ông bèn bày ra một dãy  $n$  chai chứa đầy rượu, và nói với Bờm rằng có thể uống bao nhiêu tùy ý, nhưng đã **chọn** **chai nào thì phải uống hết** và **không được uống ở ba chai liên nhau** bởi đó là điều xui xẻo.

*Bạn hãy chỉ cho Bờm cách uống được nhiều rượu nhất.*

**Dữ liệu:** Vào từ file văn bản BOTTLES.INP

- Dòng 1: Ghi số nguyên dương  $n$  ( $n \leq 10000$ )
- Các dòng tiếp ghi các số nguyên dương ( $\leq 10000$ ) là dung tích của các chai rượu phú ông bày ra, theo thứ tự liệt kê từ chai thứ nhất tới chai thứ  $n$ , các số được ghi cách nhau bởi dấu cách hoặc dấu xuống dòng.

**Kết quả:** Ghi ra file văn bản BOTTLES.OUT

- Dòng 1: Ghi số chai được chọn và lượng rượu tối đa có thể uống cách nhau một dấu cách.
- Các dòng tiếp theo, mỗi dòng ghi chỉ số của một chai chọn ra được

**Ví dụ:**

BOTTLES . INP
6
6 10 10 13
10 10

BOTTLES . OUT
4 40
2
3
5
6

cuu duong than cong . com

cuu duong than cong . com

## 060. SỐ THẬP PHẦN

Kết quả của phép chia:  $a/b$  với  $a$  và  $b$  là hai số nguyên ( $b \neq 0$ ) có thể biểu diễn dưới dạng một số thập phân hữu hạn hoặc số thập phân vô hạn tuần hoàn.

Ví dụ:

$$\begin{aligned}6/25 &= 0.24 \\1/3 &= 0.(3) \\-17/140 &= -0.12(142857)\end{aligned}$$

**Vấn đề đặt ra là khi biết hai số nguyên  $a, b$  ( $-10^9 \leq a \leq 10^9$ ;  $-10^7 \leq b \leq 10^7$ ;  $b \neq 0$ ). Hãy tìm biểu diễn thập phân của phép chia  $a/b$ .**

**Dữ liệu:** Vào từ file văn bản DECIMAL.INP

Input file gồm nhiều dòng, mỗi dòng ghi một bộ dữ liệu là cặp số nguyên  $a, b$  cách nhau một dấu cách.

**Kết quả:** Ghi ra file văn bản DECIMAL.OUT

Output file có số dòng bằng số dòng của input file, chương trình phải ghi kết quả tương ứng với bộ dữ liệu thứ  $i$  trong input file vào dòng thứ  $i$  của output file.

**Chú ý:**

- Trong trường hợp  $a/b$  là số nguyên thì chỉ ghi kết quả phần nguyên, không có phần thập phân và dấu chấm thập phân.
- Trường hợp  $a/b$  là số thập phân hữu hạn, không được ghi thừa số 0 ở cuối.
- Trường hợp  $a/b$  là số thập phân vô hạn tuần hoàn, phần thập phân đứng trước chu kỳ phải là ngắn tối thiểu.

**Ví dụ:**

DECIMAL.INP	DECIMAL.OUT	DECIMAL.OUT dưới đây tuy giá trị đúng nhưng là sai khuôn dạng
100 10	10	10.00
6 25	0.24	0.240
1 3	0.(3)	0.33(3)
99 101	0.(9801)	0.98(0198)
431 3500	0.123(142857)	0.123142(857142)



## 061. DANH SÁCH VÒNG

Để làm việc với một danh sách gồm  $N$  số nguyên cần phải có hai thao tác.

- Thao tác Top chuyển phần tử đầu tiên của danh sách xuống vị trí cuối cùng của danh sách.
- Thao tác Bottom chuyển phần tử cuối cùng của danh sách lên vị trí đầu tiên của danh sách.

Một phép biến đổi danh sách đã cho là việc thực hiện  $K$  lần thao tác Top, rồi sau đó đến  $L$  lần thao tác Bottom.

Do số lần thực hiện phép biến đổi trên là rất lớn nên đòi hỏi phải có những thủ tục thực hiện hiệu quả để thực hiện liên tiếp  $X$  phép biến đổi đưa danh sách về trạng thái cuối cùng.

**Yêu cầu:** Viết chương trình cho phép với một danh sách và ba số  $K$ ,  $L$ ,  $X$  cho trước, xác định trạng thái của danh sách sau  $X$  lần thực hiện phép biến đổi.

**Dữ liệu:** Vào từ file văn bản CLIST.INP

- Dòng đầu tiên chứa ba số nguyên dương  $N$ ,  $K$ ,  $L$  ( $1 \leq N, K, L \leq 10000$ ).
- Dòng thứ hai chứa  $N$  số nguyên, mỗi số có giá trị tuyệt đối không quá 10000, được sắp xếp theo thứ tự tương ứng với trạng thái khởi đầu của danh sách.
- Dòng thứ ba chứa số nguyên  $X$  ( $0 \leq X \leq 2 \cdot 10^9$ ).

**Kết quả:** Ghi ra file văn bản CLIST.OUT

Ghi ra trên một dòng của file văn bản CLIST.OUT các phần tử của danh sách sau  $X$  phép biến đổi. Các phần tử phải được ghi đúng thứ tự từ phần tử đầu tiên đến phần tử cuối cùng.

Các số trên một dòng của Input/Output File ghi cách nhau ít nhất một dấu cách.

**Ví dụ:**

CLIST.INP
5 2 1
3 5 2 4 7
9

CLIST.OUT
7 3 5 2 4

## 062. TÍNH DIỆN TÍCH

Cho một lưới ô vuông kích thước  $M \times N$ . Mỗi ô chứa một số 0 hoặc 1. Các số 1 trên lưới tạo thành một đường kín (tức là dãy các ô mà hai ô liên tiếp có chung cạnh hoặc đỉnh và ô cuối cùng của dãy có chung cạnh hoặc đỉnh với ô đầu tiên) bọc được một vùng của lưới mà ta sẽ gọi là một hình. Diện tích của hình là số ô chứa số 0 nằm trong đó.

**Yêu cầu:** *Viết chương trình tính diện tích của hình trong một lưới ô vuông cho trước. Giả thiết là diện tích của một hình khác 0.*

**Dữ liệu:** Vào từ file văn bản SZERO.INP:

Dòng đầu tiên chứa hai số nguyên dương  $M, N$  ( $5 \leq M, N \leq 100$ )

$M$  dòng tiếp theo mô tả bảng cho trước, mỗi dòng chứa dãy gồm  $N$  số 0 hoặc 1 được ghi liền nhau

**Kết quả:** Ghi ra trên một dòng của file văn bản SZERO.OUT diện tích của hình trên lưới đã cho.

**Ví dụ:**

SZERO . INP	SZERO . OUT
6 8 01000000 10100000 10010000 10001000 01010000 00100000	7

SZERO . INP	SZERO . OUT
5 5 00000 01111 10010 01010 00100	3

## 063. THANG MÁY

Trong toà nhà của một trung tâm thương mại gồm 101 tầng (các tầng được đánh số từ 0 đến 100) khách hàng có thể sử dụng hai loại thang máy:

- Thang máy loại I: cho phép di chuyển đến bất kỳ tầng nào với thời gian di chuyển qua một tầng là  $E_1$  giây.
- Thang máy loại II (siêu tốc) chỉ dừng lại ở các tầng có chỉ số chia hết cho 10, thực hiện việc di chuyển qua 10 tầng với thời gian là  $E_2$  giây.

Bất kể thang máy đang ở đâu, thời gian chờ đợi thang máy I và II (để chuyển thang máy hoặc vào thang máy) là  $W_1$  và  $W_2$  giây tương ứng. Ngoài ra tại mỗi tầng, khách hàng còn có thể di chuyển từ tầng này lên tầng trên hoặc xuống tầng dưới theo cầu thang cố định với thời gian là  $S$  giây.

**Yêu cầu:** Xác định thời gian nhỏ nhất  $T$  cần thiết để một khách hàng có thể di chuyển từ tầng  $X$  đến tầng  $Y$ . Giả thiết là  $1 \leq E_1, E_2, W_1, W_2, S \leq 1000$ .

**Dữ liệu:** Vào từ file văn bản LMOVE.INP

- Dòng đầu tiên chứa hai số  $E_1, W_1$ .
- Dòng thứ hai chứa hai số  $E_2, W_2$ .
- Dòng thứ ba chứa số  $S$
- Dòng thứ tư chứa hai số  $X, Y$ .

**Kết quả:** Ghi ra file văn bản LMOVE.OUT thời gian  $T$  tìm được

Các số trên một dòng của input file được ghi cách nhau ít nhất một dấu cách.

**Ví dụ:**

LMOVE . INP	LMOVE . OUT
2 25	96
4 15	
10	
85 43	

Cách di chuyển tối ưu với dữ liệu trên như sau:

Đang ở tầng 85, chờ thang loại I: 25 giây  
Tụt xuống tầng 80: 2giây x 5 = 10 giây  
Chờ thang loại II: 15 giây  
Tụt xuống tầng 40: 4giây x 4 = 16 giây  
Di chuyển theo cầu thang lên tầng 43: 10giây x 3 = 30 giây

Tổng cộng: 96 giây

## 064. TRỌNG SỐ XÂU

Xét tập chữ cái  $A = \{I, W, N\}$ . Một từ là một dãy liên tiếp không quá 6 ký tự của  $A$ . Cho một danh sách  $L$  gồm  $m$  từ phân biệt.

- Mỗi từ trong danh sách được gán một trọng số dương  $\leq 60000$ .
- Những từ không có trong danh sách mang trọng số 0.

Xét một xâu  $S$  chỉ gồm các ký tự trong  $A$ . Trọng số của xâu  $S$  được tính bằng tổng trọng số các từ trong  $S$ . (Các từ trong  $S$  được liệt kê dưới dạng các đoạn ký tự liên tiếp của  $S$  tính cả việc giao nhau và chứa nhau)

**Yêu cầu:** Cho trước danh sách  $L$  và độ dài  $n \leq 100$ . Hãy tìm xâu  $S = S_1S_2...S_n$  có trọng số nhỏ nhất. Nếu có nhiều xâu  $S$  đều có trọng số nhỏ nhất thì chỉ cần chỉ ra một xâu.

**Dữ liệu:** Vào từ file văn bản STR.INP

- Dòng 1: Ghi hai số  $n, m$  cách nhau một dấu cách.
- $m$  cặp dòng tiếp theo, cặp dòng thứ  $i$  gồm 2 dòng:
  - Dòng thứ nhất ghi từ thứ  $i$  trong danh sách  $L$
  - Dòng thứ hai ghi trọng số của từ đó

**Kết quả:** Ghi ra file văn bản STR.OUT gồm 2 dòng:

- Dòng 1: Ghi trọng số của từ  $S$  tìm được
- Dòng 2: Ghi xâu ký tự  $S$

**Ví dụ:**

STR. INP	STR. OUT	STR. INP	STR. OUT
8 10	62	8 8	98
I	WWIWWIWW	W	IWIWIWIW
13		10	
W		I	
6		10	
N		N	
12		30	
II		WI	
6		1	
NI		WW	
6		10	
IIN		II	
13		11	
WWW		WIW	
7		2	
WNN		IWI	
23		3	
NWW			
18			
NWN			
0			

## 065. PHỐ MAY MẮN

Người dân thành phố Byteland có rất nhiều điều kiêng kỵ trong cuộc sống. Theo quan điểm của họ, các số 2, 6, 13 và nhiều số khác không mang lại điều may mắn. Trong khi đó, các số 3, 5, 7 lại rất được ưa chuộng. Những ngôi nhà có số mà khi phân tích ra thừa số nguyên tố chỉ chứa các thừa số 3, 5, 7 được coi là may mắn và được mua rất nhanh.

Sau một thời gian dài thảo luận, Hội đồng thành phố quyết định đánh số tất cả các ngôi nhà trên một đường phố mới mở bằng các số may mắn liên tiếp nhau, biến phố đó thành một phố may mắn. Ký hiệu dãy các số may mắn là  $X_1, X_2, X_3, X_4, \dots$ . Khi đó các nhà bên trái sẽ mang số  $X_1, X_3, X_5$ . Còn dãy nhà bên phải sẽ mang số  $X_2, X_4, X_6, \dots$ . Toàn bộ đường phố có không quá 4000 nhà.

**Hãy xác định xem một số cho trước có phải là một số nhà ở phố may mắn không. Nếu đúng thì cho biết nhà đó nằm ở bên phải hay bên trái của phố.**

**Dữ liệu:** Vào từ file văn bản STREET.INP gồm không quá 100000 dòng, mỗi dòng chứa một số nguyên dương không quá 18 chữ số.

**Kết quả:** Ghi ra file văn bản STREET.OUT, gồm nhiều dòng, mỗi dòng tương ứng với một số ở file dữ liệu vào và chứa một trong ba chữ cái L, R, N tương ứng với nhà bên trái, bên phải hay không phải số nhà ở phố may mắn.

Lưu ý: Dãy số may mắn được tính bắt đầu từ  $X_1=3$ .

**Ví dụ:**

STREET . INP
5
3
4
98415
12814453125

STREET . OUT
R
L
N
R
L

## 066. TÍN HIỆU GIAO THÔNG

Trong một thành phố có:

- $m$  đường phố (hai chiều) song song chạy thẳng dọc theo hướng Tây↔Đông, để tiện, ta gọi các đường phố đó là  $H_1, H_2, \dots, H_m$  theo thứ tự từ Bắc xuống Nam.
- $n$  đường phố (hai chiều) song song chạy thẳng theo hướng Bắc↔Nam, ta gọi các đường phố đó là  $V_1, V_2, \dots, V_n$  theo thứ tự từ Tây sang Đông

Hai đường phố vuông góc bất kỳ cắt nhau tạo thành một nút giao thông. Ngoại trừ hai nút giao thông nằm ở vị trí góc Đông-Nam và góc Tây-Bắc những nút giao thông khác có thể gắn đèn tín hiệu giao thông hai trạng thái:

**0.** Trạng thái EW: Xanh hướng Đông và Tây, Đỏ hướng Bắc và Nam.

**1.** Trạng thái NS: Xanh hướng Bắc và Nam, Đỏ hướng Đông và Tây.

Mỗi đèn tín hiệu có một chu kỳ thời gian riêng, cứ sau mỗi chu kỳ thời gian đó, đèn đổi trạng thái một lần. Tại thời điểm 0, các đèn tín hiệu đều ở trạng thái 0 (EW).

Để giữ an toàn, luật giao thông quy định: Khi xe tới một nút giao thông từ một hướng nào đó dừng vào thời điểm đèn tín hiệu theo hướng đó đang Đỏ hay chuyển sang Đỏ thì buộc phải dừng lại, đứng vào thời điểm đèn tín hiệu theo hướng đó đang Xanh hay chuyển sang Xanh thì có thể đi thẳng, rẽ phải hay rẽ trái tùy ý.

Trên một đường phố, thời gian xe đi giữa hai nút giao thông liên tiếp cố định là 1 đơn vị thời gian.

**Yêu cầu:** Cho biết sơ đồ giao thông và các đèn tín hiệu. Cho một xe xuất phát tại thời điểm 0 từ nút giao thông ở góc Tây-Bắc. Tìm hành trình và thời điểm sớm nhất để xe tới nút giao thông ở góc Đông-Nam.

**Dữ liệu:** Vào từ file văn bản TRAFFIC.INP

- Dòng 1: Ghi hai số nguyên dương  $m, n$  ( $m, n \leq 100$ )
- Dòng 2: Ghi số  $k$  là số đèn hiệu giao thông
- $k$  dòng tiếp theo, dòng thứ  $i$  gồm 3 số nguyên dương  $x, y, t$  cho biết đèn hiệu thứ  $i$  nằm ở giao điểm của đường  $H_x$  và  $V_y$  có chu kỳ là  $t$  ( $t \leq 10000$ ).

**Kết quả:** Ghi ra file văn bản TRAFFIC.OUT

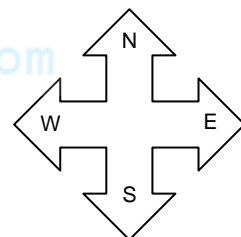
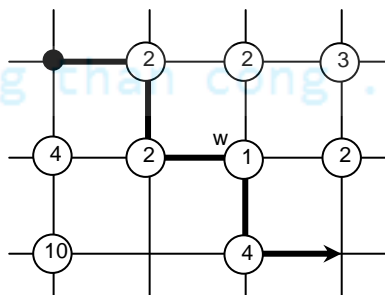
- Dòng 1: Ghi thời điểm sớm nhất để xe chạy từ góc Tây-Bắc tới góc Đông-Nam
- Dòng 2: Ghi một dãy ký tự, ký tự thứ  $p \in \{w, E, W, S, N\}$  cho biết trong khoảng thời gian từ  $p-1$  tới  $p$ , xe trong trạng thái đứng đợi hay chạy theo hướng Đông, Tây, Nam hay Bắc (theo thứ tự  $w, E, W, S, N$  đó).

Các số trên một dòng của Input File được ghi cách nhau ít nhất một dấu cách.

**Ví dụ:**

TRAFFIC.INP		
3	4	
9		
1	2	2
1	3	2
1	4	3
2	1	4
2	2	2
2	3	1
2	4	2
3	1	10
3	3	4

TRAFFIC.OUT		
6		
ESEwSE		



## 067. PHÂN NHÓM

Cho  $n$  học sinh và  $m$  đặc điểm ( $n \leq 100$ ), ( $m \leq 10$ ).

*Cần phân các học sinh này thành một số ít các nhóm nhất để đảm bảo rằng ta chỉ cần quan tâm tới một số ít nhất các đặc điểm là có thể phân biệt được các học sinh trong nội bộ một nhóm.*

**Chú ý:**

- Trước tiên phải thỏa mãn yêu cầu ít nhóm nhất, trong các cách chia ít nhóm nhất mà vẫn có thể phân biệt được các học sinh trong một nhóm thì chỉ ra một cách chia phải dùng ít đặc điểm nhất.*
- Tập các đặc điểm được chọn phải sử dụng được trên tất cả các nhóm để phân biệt học sinh.*

**Dữ liệu:** Vào từ file văn bản GROUP.INP

- Dòng 1 ghi hai số  $n, m$
- $n$  dòng tiếp theo, dòng thứ  $i$  mô tả đặc điểm của học sinh thứ  $i$ : Gồm có  $m$  số nguyên mà số thứ  $j$  là 1 hay 0 tùy theo học sinh thứ  $i$  có hay không có đặc điểm  $j$ .

**Kết quả:** Ghi ra file văn bản GROUP.OUT

Dòng 1: Ghi số  $k$  là số nhóm chia ra được

Dòng 2: Ghi các đặc điểm được chọn để phân biệt các học sinh trong nội bộ các nhóm  $k$  dòng tiếp theo, dòng thứ  $p$  ghi các học sinh trong nhóm  $p$

*Các số trên một dòng của Input/Output File được ghi cách nhau ít nhất một dấu cách.*

**Ví dụ:**

GROUP . INP	GROUP . OUT	GROUP . INP	GROUP . OUT (Không tối ưu)
10 4	2	10 4	2
0 0 0 1	1 2 4	0 0 0 1	1 2 3 4
0 0 1 0	2 5 10 1 6	0 0 1 0	1 2 5 6 7 10
0 1 1 0	4 3 9 7 8	0 1 1 0	3 4 8 9
1 0 0 0		1 0 0 0	
1 0 0 0		1 0 0 0	
1 0 0 1		1 0 0 1	
1 0 1 1		1 0 1 1	
1 1 0 1		1 1 0 1	
1 1 1 0		1 1 1 0	
1 1 1 0		1 1 1 0	

cuu duong than cong . com

## 068. TUA DU LỊCH RẺ NHẤT

Một khu thắng cảnh gồm  $n$  điểm đánh số từ 1 tới  $n$  ( $n \leq 100$ ) và  $m$  đường đi hai chiều giữa các cặp địa điểm đó, chi phí đi trên các đường đi là biết trước ( $\leq 10000$ ).

Một Tour du lịch là một hành trình xuất phát từ một địa điểm đi thăm  $\geq 2$  địa điểm khác và quay trở về điểm xuất phát, ngoại trừ địa điểm xuất phát, không địa điểm nào bị thăm tới hai lần. Chi phí của một Tour du lịch là tổng chi phí các quãng đường đi qua.

**Yêu cầu:** Hãy tìm Tour du lịch có chi phí rẻ nhất.

**Dữ liệu:** Vào từ file văn bản TOUR.INP

- Dòng 1: Ghi hai số nguyên dương  $n, m$
- $m$  dòng tiếp theo mỗi dòng có dạng  $x \ y \ c$ . Cho biết có đường đi trực tiếp nối địa điểm  $x$  với địa điểm  $y$  và chi phí đi quãng đường đó là  $c$ .

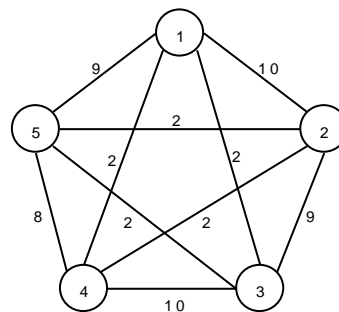
**Kết quả:** Ghi ra file văn bản TOUR.OUT

- Dòng 1: Ghi số 1 nếu như tồn tại hành trình theo yêu cầu, ghi số 0 nếu không tồn tại hành trình.
- Nếu dòng đầu tiên ghi số 1:
  - ♦ Dòng thứ 2 ghi chi phí của tour tìm được
  - ♦ Dòng thứ 3 ghi số  $k$  là số địa điểm tới thăm
  - ♦ Dòng thứ 4 gồm  $k$  số, số thứ  $i$  là địa điểm tới thăm thứ  $i$  trong tour, quy ước địa điểm thăm đầu tiên là địa điểm xuất phát, địa điểm thăm thứ  $k$  (địa điểm cuối cùng) là địa điểm mà từ đó quay trở lại điểm xuất phát để kết thúc hành trình.

Các số trên một dòng của Input/Output File được ghi cách nhau ít nhất một dấu cách.

Ví dụ:

TOUR . INP	TOUR . OUT
5 10	1
1 3 2	10
2 4 2	5
3 5 2	3 5 2 4 1
4 1 2	
5 2 2	
1 2 10	
2 3 9	
3 4 10	
4 5 8	
5 1 9	





## 069. DU LỊCH NHIỀU TUA NHẤT

Một khu thắng cảnh gồm  $n$  điểm đánh số từ 1 tới  $n$  ( $n \leq 200$ ) và  $m$  đường đi hai chiều giữa các cặp địa điểm đó.

Một Tour du lịch là một hành trình xuất phát từ một địa điểm đi thăm  $\geq 2$  địa điểm khác và quay trở về điểm xuất phát, ngoại trừ địa điểm xuất phát, không địa điểm nào bị thăm tới hai lần.

**Yêu cầu:** Hãy tìm một số tour du lịch nhiều nhất sao cho mỗi tour du lịch tìm được đều có một đoạn đường riêng hoàn toàn không có mặt trong các tua du lịch còn lại.

**Dữ liệu:** Vào từ file văn bản TOURS.INP

- Dòng 1: Ghi hai số  $n, m$
- $m$  dòng tiếp theo mỗi dòng có dạng  $x \ y$  cho biết giữa hai địa điểm  $x$  và  $y$  có đường đi trực tiếp.

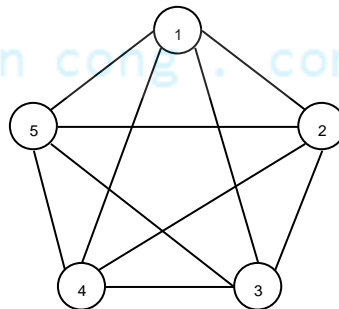
**Kết quả:** Ghi ra file văn bản TOURS.OUT

- Dòng 1: Ghi số  $k$  là số tour du lịch tìm được
- $k$  dòng tiếp theo, dòng thứ  $i$  mô tả tour du lịch thứ  $i$ : bắt đầu là số địa điểm thăm được trong tour, tiếp theo là danh sách các địa điểm theo thứ tự trong hành trình bắt đầu từ địa điểm xuất phát cho tới kết thúc là địa điểm mà từ đó quay lại điểm xuất phát để kết thúc hành trình

*Các số trên một dòng của Input/Output file được ghi cách nhau ít nhất một dấu cách*

**Ví dụ:**

TOURS . INP	TOURS . OUT
5 10	6
1 3	3 3 2 1
2 4	4 4 3 2 1
3 5	3 4 3 2
4 1	5 5 4 3 2 1
5 2	4 5 4 3 2
1 2	3 5 4 3
2 3	
3 4	
4 5	
5 1	



cuu duong than cong . com

## 070. PHÂN CÔNG

Có  $m$  thợ và  $n$  công việc, các thợ đánh số từ 1 tới  $m$  và các việc đánh số từ 1 tới  $n$ . Mỗi thợ có khả năng thực hiện một số công việc nào đó.

Khi giao việc cho các thợ thực hiện, đối với một người thợ thì họ sẽ thực hiện các công việc được giao một cách tuần tự và liên tục (sequence), làm mỗi việc mất một đơn vị thời gian. Nhưng đối với nhiều thợ thì các công việc của họ được thực hiện song song (parallel), việc của ai người đấy làm, không ảnh hưởng tới tiến độ của người khác.

**Hãy tìm các phân công công việc cho các thợ để tất cả các công việc được thực hiện, mỗi việc chỉ phân cho một thợ và thời gian hoàn thành tất cả các công việc là nhanh nhất.**

**Dữ liệu:** Vào từ file văn bản ASSIGN.INP

- Dòng 1: Chứa hai số nguyên dương  $m$  và  $n$  ( $1 \leq m \leq 100$ ;  $1 \leq n \leq 500$ )
- $m$  dòng tiếp theo, dòng  $i$  chứa danh sách các công việc mà thợ  $i$  có thể thực hiện, có thêm một ký hiệu kết thúc là số 0.

**Kết quả:** Ghi ra file văn bản ASSIGN.OUT

- Dòng 1: Ghi từ YES hay NO tùy theo có tồn tại cách phân công để thực hiện tất cả các công việc hay không.
- Nếu dòng 1 ghi từ YES:
  - ♦ Dòng 2: Ghi thời gian nhanh nhất có thể để hoàn thành các công việc
  - ♦  $m$  dòng tiếp theo, dòng  $i$  ghi danh sách các công việc được phân cho thợ  $i$ , ghi thêm một ký hiệu kết thúc là số 0.

**Các số trên một dòng của Input/Output File được ghi cách nhau ít nhất một dấu cách**

**Ví dụ:**

ASSIGN.INP	ASSIGN.OUT
4 10	YES
1 2 3 4 5 0	3
4 5 6 7 8 0	3 4 5 0
1 2 3 4 5 7 8 9 0	6 7 8 0
1 2 3 4 5 6 7 8 9 10 0	2 9 0
	1 10 0

cuu duong than cong . com

## 071. NHẮN TIN

Một khoá học có  $n$  học viên đánh số từ 1 tới  $n$ , mỗi học viên có thể biết số điện thoại của một vài học viên khác.

Học viên A có thể nhắn tin cho học viên B nếu như học viên A biết số điện thoại của học viên B. Lưu ý rằng việc biết số điện thoại ở đây không phải quan hệ đối xứng: Có thể học viên A biết số điện thoại của học viên B nhưng học viên B hoàn toàn không biết số điện thoại của học viên A.

*Thầy giáo nắm được tất cả số điện thoại của các học viên trong hồ sơ của trường, hỏi khi thầy giáo muốn nhắn tin tới tất cả các học viên trong khoá, thầy giáo sẽ phải nhắn trực tiếp tới một số ít nhất các học viên nào để thông điệp đó đến được tất cả các học viên khác.*

**Dữ liệu:** Vào từ file văn bản MESSAGE.INP

- Dòng 1 chứa số  $n$  ( $n \leq 700$ )
- Các dòng tiếp theo, mỗi dòng chứa hai số nguyên dương  $x, y$  ( $x \neq y: 1 \leq x, y \leq n$ ) cho ta thông tin: học viên  $x$  biết số điện thoại của học viên  $y$

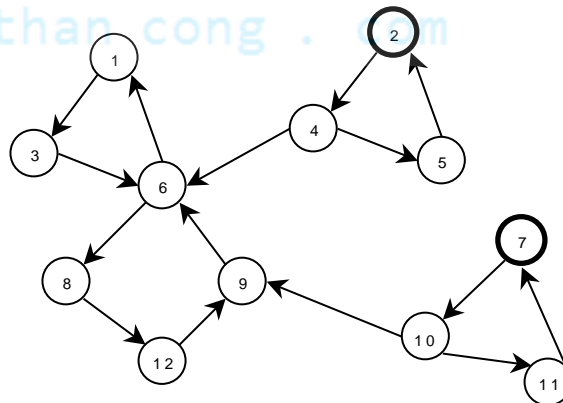
**Kết quả:** Ghi ra file văn bản MESSAGE.OUT

- Dòng 1: Ghi số  $k$  là số học sinh được thầy giáo nhắn tin trực tiếp khi cần
- Dòng 2: Ghi  $k$  số hiệu của các học sinh được thầy giáo nhắn tin trực tiếp

*Các số trên một dòng của Input/Output File được ghi cách nhau ít nhất một dấu cách.*

Ví dụ:

MESSAGE . INP	MESSAGE . OUT
12	2
1 3	7 2
3 6	
6 1	
6 8	
8 12	
12 9	
9 6	
2 4	
4 5	
5 2	
4 6	
7 10	
10 11	
11 7	
10 9	



Giới hạn không gian và thời gian: 512KB - 1 giây

## 072. CÁC SỐ ĐIỆN THOẠI

Ngày nay bạn phải nhớ quá nhiều số điện thoại mà chúng lại ngày càng dài hơn. Một trong những cách để dễ ghi nhớ các con số như vậy là thay thế các chữ số bằng chữ cái theo một qui ước nào đó. Ví dụ như ta có thể thay:

1 = ij      2 = abc      3 = def      4 = gh      5 = kl      6 = mn      7 = prs  
8 = tuv      9 = wxy      0 = oqz

Bằng cách này, mỗi từ hoặc một nhóm từ có thể gán cho một số duy nhất, và vì thế bạn có thể nhớ các từ thay vì các con số. Ví dụ số điện thoại của người bạn chơi cờ 941837296 thì có thể nhớ bởi từ WHITE PAWN còn số điện thoại của một thầy giáo 8322437 thì có thể nhớ bằng từ TEACHER thì dễ nhớ hơn nhiều so với các con số dài dòng đó.

**Cho biết các phép thay thế số bằng chữ cái, và một từ điển. Hãy tìm một dãy gồm ít nhất các từ để gán cho con số cần ghi nhớ cho trước. Mỗi từ có thể dùng nhiều lần.**

**Dữ liệu:** Vào từ file văn bản PHONE.INP

- 10 dòng đầu tiên, dòng thứ i ghi danh sách các chữ cái có thể dùng để thay cho số i - 1.
- Dòng 11 ghi con số cần ghi nhớ (không quá 100 chữ số)
- Các dòng tiếp theo, mỗi dòng ghi một từ trong từ điển, mỗi từ gồm không quá 50 chữ cái tiếng Anh in thường. Ký hiệu kết thúc từ điển là dòng cuối cùng ghi dấu #. Số từ trong từ điển không quá 50000.

Trong Input File hoàn toàn không chứa dấu cách.

**Kết quả:** Ghi ra file văn bản PHONE.OUT

- Dòng thứ nhất: Ghi từ YES hay NO tùy theo có phép gán dãy từ cho số đã cho hay không ?
- Nếu dòng thứ nhất ghi từ YES, dòng thứ hai, ghi danh sách các từ để ghép lại theo đúng thứ tự đó sẽ được số đã cho, các từ ghi cách nhau ít nhất một dấu trống.

**Ví dụ:**

PHONE . INP	PHONE . OUT
oqz	YES
ij	reality our
abc	
def	
gh	
kl	
mn	
prs	
tuv	
wxy	
7325189087	
it	
your	
reality	
real	
our	
#	

### 073. GIÁ TRỊ LỚN NHẤT

Một số nguyên dương  $x$  gọi là con của số nguyên dương  $y$  nếu ta có thể xoá bớt một số chữ số của  $y$  để được  $x$ .

**Cho hai số  $a$  và  $b$  hãy tìm số  $c$  là con của cả  $a$  và  $b$  sao cho giá trị của  $c$  là lớn nhất có thể.**

Ràng buộc:  $1 \leq a, b \leq 10^{100}$ , Dữ liệu vào luôn có nghiệm.

**Dữ liệu:** Vào từ file văn bản NUMBER.INP

- Dòng thứ nhất chứa số  $a$
- Dòng thứ hai chứa số  $b$

**Kết quả:** Ghi ra file văn bản NUMBER.OUT

- Ghi ra trên một dòng số  $c$ .

**Ví dụ:**

NUMBER . INP	NUMBER . OUT
123456781234 567812345678	56781234

NUMBER . INP	NUMBER . OUT
2468013579 1234567890	24689

cuu duong than cong . com

cuu duong than cong . com

## 074. NÚT GIAO THÔNG TRỌNG ĐIỂM

Trong một đường phố có  $n$  nút giao thông và  $m$  đường hai chiều nối trực tiếp các cặp nút giao thông đó, giữa hai nút giao thông bất kỳ có không quá một đường đi trực tiếp.

Một nút giao thông  $c$  được gọi là trọng điểm nếu tồn tại hai nút giao thông  $a$  và  $b$  ( $a, b, c$  đôi một khác nhau) sao cho:

- Giữa  $a$  và  $b$  có ít nhất một đường đi theo các đường phố đã cho
- Nếu nút  $c$  bị tắc thì không có cách nào đi từ  $a$  sang  $b$ . Hay nói cách khác, mọi đường đi từ  $a$  tới  $b$  chắc chắn phải qua  $c$ .

**Cho biết sơ đồ giao thông của thành phố, hãy xác định các nút giao thông trọng điểm.**

**Dữ liệu:** Vào từ file văn bản CNODE.INP

- Dòng 1: Ghi hai số nguyên dương  $n, m$  ( $n \leq 1000; m \leq 10000$ )
- $m$  dòng tiếp theo, mỗi dòng ghi hai số nguyên dương  $u, v$ , cho ta thông tin: Giữa hai nút giao thông  $u$  và  $v$  có một đường đi trực tiếp.

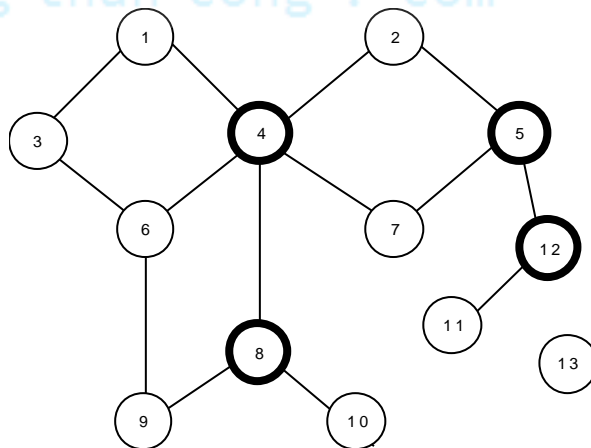
**Kết quả:** Ghi ra file văn bản CNODE.OUT

- Dòng 1: Ghi số nút giao thông trọng điểm
- Dòng 2: Ghi chỉ số của các nút giao thông trọng điểm, các chỉ số này phải liệt kê đôi một khác nhau.

**Các số trên một dòng của Input/Output File được ghi cách nhau ít nhất một dấu cách**

**Ví dụ:**

CNODE . INP	CNODE . OUT
13 14	4
1 3	4 5 8 12
3 6	
6 4	
4 1	
4 2	
2 5	
5 7	
7 4	
6 9	
9 8	
8 4	
8 10	
11 12	
5 12	



## 075. TẬP KẾT

Một bàn cờ kích thước  $n \times n$  ( $2 \leq n \leq 100$ ) trong đó đánh dấu một số ô cấm. Trên bàn cờ có  $k$  quân mã đang đứng ở những vị trí nào đó ( $1 \leq k \leq 100$ ). Cần đi những quân mã này đến  $k$  vị trí tập kết (mỗi quân mã một vị trí). Trong quá trình di chuyển, mã không được nhảy đến các ô cấm nhưng có thể nhảy đến ô đã có những quân mã khác đang đứng. Vai trò của các quân mã và các vị trí tập kết là như nhau (một quân mã có thể cho đi tới bất kỳ vị trí tập kết nào nếu có đường nhảy). ở trạng thái ban đầu  $k$  vị trí xuất phát và  $k$  vị trí tập kết được cho hoàn toàn phân biệt

**Yêu cầu:** *Lập chương trình xác định cách đi các quân mã sao cho tổng số bước đi của các quân mã là nhỏ nhất.*

C		C			S
			S	D	
		C			
		S	C		
C	C		S	D	D
C					D

**Dữ liệu:** Vào từ file văn bản HORSES.INP

- Dòng 1: Ghi số  $n$  ( $n \leq 100$ )
- $n$  dòng tiếp theo, dòng  $i$ , ghi  $n$  ký tự thể hiện hàng  $i$  của bàn cờ. Ký tự thứ  $i$  là:
  - ♦ ". " : Thể hiện ô trống
  - ♦ "C" : Thể hiện ô cấm
  - ♦ "S" : Thể hiện ô có mã đang đứng
  - ♦ "D" : Thể hiện ô ở vị trí tập kết

**Kết quả:** Ghi ra file văn bản HORSES.OUT

- Dòng 1: Ghi số  $m$  là tổng bước di chuyển để đưa các quân mã về vị trí tập kết. Nếu không có cách tập kết thì ghi số -1.
- $m$  dòng tiếp theo, dòng thứ  $i$  ghi 4 số  $x_1 \ y_1 \ x_2 \ y_2$  cách nhau ít nhất một dấu cách, cho biết tại bước thứ  $i$  sẽ di chuyển một quân mã từ ô  $(x_1, y_1)$  đến ô  $(x_2, y_2)$

**Ví dụ:**

HORSES . INP	HORSES . OUT
6	7
C . C . . S	5 4 4 6
. . . S D .	4 6 2 5
. . C . . .	4 3 5 5
. . S C . .	1 6 3 5
C C . S D D	3 5 5 6
C . . . . D	2 4 4 5
	4 5 6 6

## 076. MỜI KHÁCH DỰ TIỆC

Công ty trách nhiệm hữu hạn "Vui vẻ" có  $n$  cán bộ đánh số từ 1 tới  $n$ . Cán bộ thứ  $i$  có đánh giá độ vui tính là  $h_i$ . Ngoài trừ giám đốc công ty, mỗi người đều có một thủ trưởng trực tiếp của mình.

*Bạn cần giúp công ty mời một nhóm cán bộ đến dự dạ tiệc "Những người thích đùa" sao cho tổng đánh giá độ vui tính của những người dự tiệc là lớn nhất, với yêu cầu: trong số những người được mời không đồng thời có mặt nhân viên cùng thủ trưởng trực tiếp của người đó.*

**Dữ liệu:** Vào từ file văn bản GUEST.INP

- Dòng đầu tiên ghi số cán bộ công ty:  $n$  ( $2 \leq n \leq 10000$ )
- $n$  dòng tiếp theo, dòng thứ  $i$  gồm hai số tự nhiên  $b_i, h_i$  cho ta thông tin, người thứ  $i$  có thủ trưởng trực tiếp là  $b_i$  và độ vui tính là  $h_i$ . Nếu như  $b_i = 0$  thì ta hiểu  $i$  là giám đốc công ty.

**Kết quả:** Ghi ra file văn bản GUEST.OUT

- Dòng 1: Ghi số người được mời ( $k$ ) và tổng độ vui tính của những người đó ( $m$ )
- $k$  dòng tiếp, mỗi dòng ghi số hiệu một người được mời tới dạ tiệc.
- *Các số trên một dòng của Input/Output File được ghi cách nhau ít nhất một dấu cách*
- *Dữ liệu vào được cho đúng đắn: không tồn tại một dãy  $x_1, x_2, \dots, x_p, x_{p+1} = x_1$  mà người  $i$  là thủ trưởng trực tiếp của người  $i + 1$  (ví:  $1 \leq i \leq p$ ).*
- *Không nhất thiết phải mời giám đốc công ty*

Ví dụ:

GUEST . INP	GUEST . OUT
10	4 36
2 9	1
3 7	4
4 8	6
0 10	10
4 2	
5 11	
6 6	
6 4	
4 6	
9 6	



## 077. KHÔI PHỤC NGOẶC

Một dãy dấu ngoặc hợp lệ là một dãy các ký tự "(" và ")" được định nghĩa như sau:

- Dãy rỗng (không có ký tự nào) là một dãy dấu ngoặc hợp lệ
- Nếu A là một dãy dấu ngoặc hợp lệ thì (A) là dãy dấu ngoặc hợp lệ. Dấu ngoặc mở và dấu ngoặc đóng hai bên dãy A được gọi là tương ứng với nhau
- Nếu A và B là hai dãy dấu ngoặc hợp lệ thì AB là dãy dấu ngoặc hợp lệ.

Ví dụ: ((( )))(( ))( ) là một dãy dấu ngoặc hợp lệ. các dấu mở ngoặc ở các vị trí: 1, 2, 3, 7, 8, 11, 13 tương ứng lần lượt với các dấu đóng ngoặc ở các vị trí: 6, 5, 4, 10, 9, 12, 14.

Ban đầu có một dãy dấu ngoặc hợp lệ, người ta viết vào dưới mỗi dấu ngoặc mở một số là số dấu ngoặc (cả đóng và mở) nằm giữa dấu ngoặc mở đó và dấu ngoặc đóng tương ứng:

(   (   (   )   )   )   (   (   )   )   (   )   (   )  
4   2   0                      2   0                      0                      0

Sau đó xoá đi dãy ngoặc.

**Yêu cầu:** Cho biết dãy số còn lại, hãy khôi phục lại dãy ngoặc ban đầu

**Dữ liệu:** Vào từ file văn bản BRACKETS.INP

- Dòng 1: Ghi số n là số phần tử của dãy số còn lại ( $n \leq 10000$ )
- Dòng 2: Ghi lần lượt các số trong dãy

**Kết quả:** Ghi ra file văn bản BRACKETS.OUT

Gồm 1 dòng ghi dãy dấu ngoặc khôi phục được

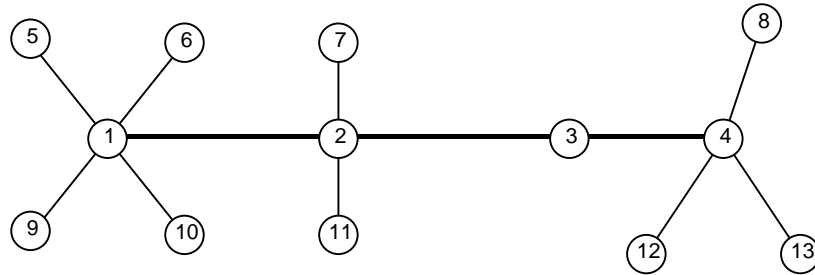
**Ví dụ:**

BRACKETS . INP	BRACKETS . OUT
7 4 2 0 2 0 0 0	(( ( )) ) ( ( ) ) ( ) ( )

BRACKETS . INP	BRACKETS . OUT
10 8 2 0 0 0 4 0 0 0 0	(( ( )) ( ) ( )) (( ( ) ) ) ( ) ( )

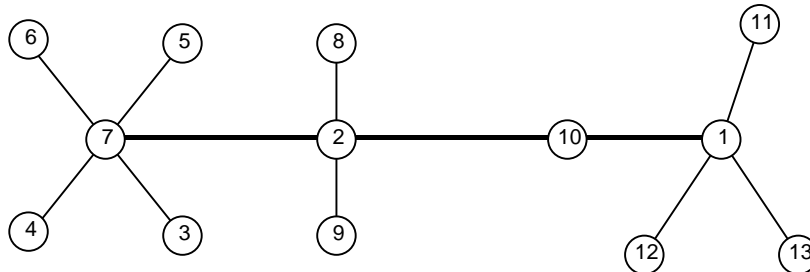
## 078. DÂY XÍCH

Một dây xích là một cây có tính chất: Tồn tại một đường đi sao cho mỗi đỉnh treo phải kề với đúng một đỉnh trên đường đi đó. Với mỗi dây xích, đường đi này không nhất thiết phải duy nhất.



Cho một dây xích với các nút được đánh số  $1..n$  ( $2 \leq n \leq 10000$ ). Hãy tìm cách gán cho mỗi đỉnh  $i$  một nhãn  $Lab(i)$ ;  $1 \leq Lab(i) \leq n$  sao cho các điều kiện sau được thỏa mãn:

- Hai đỉnh khác nhau có hai nhãn khác nhau
- Không có hai cạnh nào có cùng giá trị tuyệt đối của hiệu các nút ở hai đầu mút



**Dữ liệu:** Vào từ file văn bản CHAIN.INP

- Dòng 1: ghi số  $n$
- $n - 1$  dòng tiếp theo, mỗi dòng ghi hai đầu mút của một cạnh thuộc xích

**Kết quả:** Ghi ra file văn bản CHAIN.OUT (Nếu có nhiều lời giải thì chỉ cần chọn một)

- Một dòng  $n$  số, số thứ  $i$  là  $Lab(i)$

**Ví dụ:**

CHAIN . INP	CHAIN . OUT
13	7 2 10 1 6 5 8 11 4 3 9 12 13
1 2	
1 5	
1 6	
1 9	
1 10	
2 7	
2 11	
2 3	
3 4	
4 8	
4 12	
4 13	

## 079. PHÂN CÔNG

Có  $n$  thợ và  $n$  việc ( $n \leq 200$ ), các thợ được đánh số từ 1 tới  $n$  và các việc cũng được đánh số từ 1 tới  $n$ . Với thợ  $i$  và việc  $j$  nào đó thì có hai khả năng: Hoặc thợ  $i$  không làm được việc  $j$ , hoặc làm được với chi phí là  $c_{ij}$ . ( $c_{ij}$  là số tự nhiên  $\leq 10^9$ ).

**Hãy phân công cho mỗi thợ làm đúng một việc sao cho có thể thực hiện tất cả các công việc với tổng chi phí ít nhất có thể.**

**Dữ liệu:** Vào từ file văn bản ASSIGN.INP

- Dòng 1: Ghi số  $n$
- Các dòng tiếp, mỗi dòng ghi ba số  $i$   $j$   $c_{ij}$  cho ta thông tin: Thợ  $i$  làm được việc  $j$  với chi phí  $c_{ij}$ .

**Kết quả:** Ghi ra file văn bản ASSIGN.OUT

- Dòng 1: Ghi tổng chi phí thực hiện các công việc, nếu không tồn tại cách phân công thì dòng này ghi số -1.
- Nếu có phương án phân công,  $n$  dòng tiếp theo, dòng thứ  $i$  ghi số hiệu việc được phân cho thợ  $i$ .

**Các số trên một dòng của Input File được ghi cách nhau ít nhất một dấu cách**

**Ví dụ:**

ASSIGN.INP	ASSIGN.OUT
4	10
1 1 1	1
1 2 2	3
2 1 2	2
2 2 5	4
2 3 1	
3 2 1	
3 3 10	
4 3 10	
4 4 7	

ASSIGN.INP	ASSIGN.OUT
10	-1
2 2 6	
2 3 1	
2 6 5	
5 5 14	
7 3 10	
8 7 15	
8 9 10	

## 080. DÂY CUNG

Trên mặt phẳng với hệ trục tọa độ Decartes vuông góc, cho đường tròn có tâm  $O$  là gốc tọa độ, bán kính  $R$ . Trên đường tròn  $O$  xét  $n$  điểm xanh và  $n$  điểm đỏ đều có hoành độ nguyên, tung độ khác  $0$ . Các điểm được đánh số thứ tự từ  $1$  đến  $2n$  và nằm ở các vị trí hoàn toàn phân biệt.

Theo giả thiết ở trên, thông tin về điểm thứ  $i$  có thể cho bởi bộ ba  $(C_i, X_i, D_i)$  với:

- Ký tự  $C_i \in \{R, B\}$ ;  $C_i = R$  có nghĩa là điểm đỏ,  $C_i = B$  có nghĩa là điểm xanh
- Số nguyên  $X_i$  là hoành độ điểm đó.
- Số nguyên  $D_i \in \{-1, 1\}$ ;  $D_i = -1$  tức là tung độ âm (nằm dưới trục hoành),  $D_i = 1$  tức là tung độ dương (nằm trên trục hoành).

Dễ thấy cách xác định điểm nói trên là đúng đắn.

**Yêu cầu:** Hãy xác định  $n$  dây cung của đường tròn thỏa mãn:

- Mọi dây cung phải nối một điểm xanh với một điểm đỏ trong số các điểm kể trên
- Các dây cung đôi một không có điểm chung

**Dữ liệu:** Vào từ file văn bản CHORDS.INP

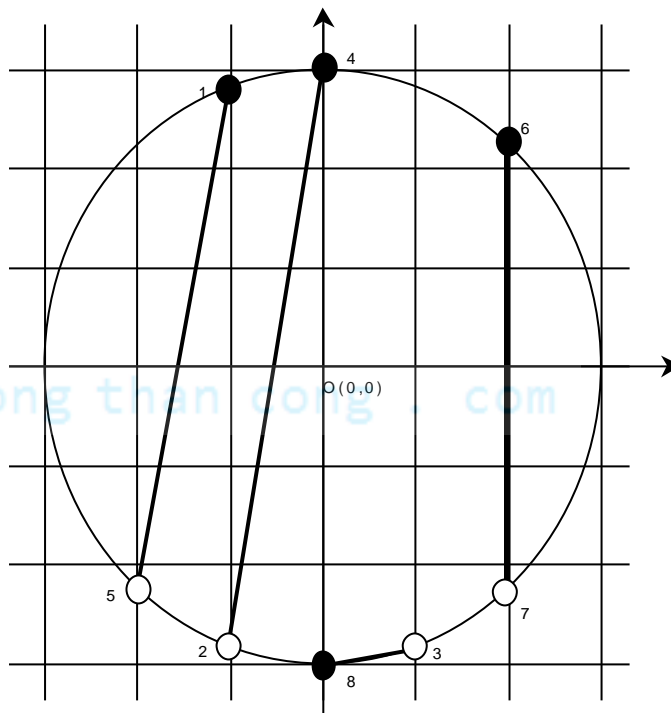
- Dòng 1: Ghi hai số nguyên dương  $n, R$  cách nhau một dấu cách ( $1 \leq n \leq 5000$ ;  $1 \leq R \leq 10001$ )
  - $2n$  dòng tiếp theo, dòng thứ  $i$  chứa thông tin về điểm thứ  $i$ :
    - ♦ Đầu dòng là ký tự  $C_i$ .
    - ♦ Tiếp theo là hoành độ  $X_i$  ( $-R < X_i < R$ )
    - ♦ Tiếp theo là số nguyên  $D_i$
- Ba thành phần này được ghi cách nhau đúng một dấu cách

**Kết quả:** Ghi ra file văn bản CHORDS.OUT

Gồm  $n$  dòng, mỗi dòng ghi chỉ số hai điểm tương ứng trên một dây cung.

**Ví dụ:**

CHORDS . INP	CHORDS . OUT
4 3	8 3
B -1 1	1 5
R -1 -1	4 2
R 1 -1	6 7
B 0 1	
R -2 -1	
B 2 1	
R 2 -1	
B 0 -1	



## 081. MÊ CUNG

Bản đồ mê cung có dạng hình chữ nhật kích thước  $m \times n$  được chia thành lưới ô vuông đơn vị bằng các đường song song với các cạnh ( $m$  hàng,  $n$  cột). Mỗi ô vuông của bản đồ được đánh dấu hoặc là ô cấm, hoặc là ô tự do. Từ một ô tự do có thể di chuyển sang các ô tự do có chung cạnh với nó. Không được phép di chuyển vượt khỏi biên của mê cung.

Mê cung được thiết kế khá đặc biệt, giữa hai ô tự do bất kỳ chỉ có **duy nhất** một cách di chuyển từ ô này đến ô kia mà trong quá trình di chuyển không đi tới bất kỳ ô nào quá một lần. Tại tâm của mỗi ô tự do đều có một cái móc. Trong mê cung có hai ô tự do đặc biệt, mà nếu bạn nối được hai cái móc ở hai ô đó bằng một sợi dây thừng (tất nhiên phải nối qua các móc của các ô trung gian) thì cánh cửa bí mật của mê cung sẽ tự mở ra.

**Vấn đề đặt ra là phải chuẩn bị một sợi dây thừng với độ dài ngắn nhất đảm bảo cho dù hai ô đặc biệt có nằm ở vị trí nào trong mê cung, bạn vẫn có thể nối được hai cái móc ở hai ô đó bằng sợi dây đã chuẩn bị.**

**Dữ liệu:** Vào từ file văn bản LABYR.INP

Dòng đầu tiên chứa hai số  $n, m$  ( $3 \leq m, n \leq 1000$ )

Các dòng tiếp theo mô tả mê cung, dòng thứ  $i$  trong số  $m$  dòng tiếp theo chứa  $n$  ký tự, mỗi ký tự chỉ là "#" hoặc ".". Trong đó ký tự "#" cho biết ô ở vị trí tương ứng là bị cấm, còn ký tự "." cho biết ô ở vị trí tương ứng là tự do ( $1 \leq i \leq m$ ).

**Kết quả:** Ghi ra trên một dòng của file văn bản LABYR.OUT độ dài của sợi dây thừng cần chuẩn bị.

**Ví dụ:**

LABYR . INP	LABYR . OUT	LABYR . INP	LABYR . OUT
### #.# ###	0	8 10 ##### .....# .#.#.#. .##### #....#. #...#. #...#. #...#. #...#. #...#. #...#.	29

## 082. DU LỊCH

Một khu thắng cảnh gồm  $n$  điểm đánh số từ 1 tới  $n$  ( $n \leq 200$ ) và  $m$  đường đi hai chiều nối giữa các cặp địa điểm đó. Giữa hai cặp địa điểm có nhiều nhất là một đường đi trực tiếp. Có hai địa điểm đặc biệt: A và B.

Một Tour du lịch là một hành trình của du khách: Trước hết là đáp máy bay xuống địa điểm A, sau đó đi bộ theo các đường hai chiều đã cho để tới địa điểm B, và lại đi bộ quay trở về địa điểm xuất phát A để rồi quay về bằng máy bay. Để tránh sự nhầm lẫn cho du khách, hành trình không được đi qua đoạn đường nào nhiều hơn một lần.

**Vấn đề đặt ra là một du khách có thể đến thăm khu thắng cảnh nhiều lần. Để phục vụ khách tham quan tốt hơn. Hãy tìm một số tour du lịch nhiều nhất sao cho hai tour du lịch bất kỳ tìm được đều không tồn tại một đoạn đường nào chung.**

**Dữ liệu:** Vào từ file văn bản TOURS.INP

- Dòng 1: Ghi bốn số  $n, m, A, B$
- $m$  dòng tiếp theo mỗi dòng có dạng  $x y$  cho biết giữa hai địa điểm  $x$  và  $y$  có đường đi trực tiếp.

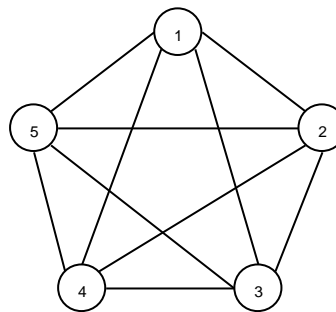
**Kết quả:** Ghi ra file văn bản TOURS.OUT

- Dòng 1: Ghi số  $k$  là số tour du lịch tìm được
- $k$  dòng tiếp theo, dòng thứ  $i$  mô tả tour du lịch thứ  $i$ : bắt đầu từ địa điểm A tiếp theo là danh sách các địa điểm theo thứ tự trong hành trình tới địa điểm B và tiếp theo là danh sách các địa điểm theo thứ tự trong hành trình quay trở lại địa điểm A. (Như vậy địa điểm A là địa điểm chắc chắn phải được liệt kê hai lần).

**Các số trên một dòng của Input/Output file được ghi cách nhau ít nhất một dấu cách**

**Ví dụ:**

TOURS . INP	TOURS . OUT
5 10 1 2	2
1 3	1 2 3 1
2 4	1 4 2 5 1
3 5	
4 1	
5 2	
1 2	
2 3	
3 4	
4 5	
5 1	



## 083. SỬA ĐƯỜNG

Trong một thành phố có  $n$  nút giao thông và  $m$  đường phố hai chiều. Giữa hai nút giao thông có nhiều nhất là một đường phố nối chúng. Hệ thống giao thông đảm bảo sự đi lại giữa hai nút bất kỳ.

Sau một thời gian dài, các đường phố xuống cấp nghiêm trọng đòi hỏi ban quản lý giao thông và công trình đô thị phải lên kế hoạch nâng cấp tất cả các đường phố. Khi một đường phố đang trong thời gian nâng cấp thì sự đi lại trên tuyến đường đó bị cấm. Xét về khả năng, với phương tiện kỹ thuật hiện đại và lực lượng nhân công dồi dào, người ta có thể tiến hành nâng cấp cùng lúc  $k$  đường phố, bất kể đường phố nào cũng chỉ cần sửa chữa trong một ngày. Tuy nhiên vì vẫn muốn đảm bảo sự đi lại giữa hai nút giao thông bất kỳ trong thời gian sửa chữa, người ta phải lên lịch thi công các tuyến đường một cách hợp lý.

**Yêu cầu:** Hãy xếp lịch thi công để thời gian nâng cấp toàn bộ các tuyến đường là ngắn nhất.

**Dữ liệu:** Vào từ file văn bản SCHEDULE.INP

- Dòng 1: Ghi ba số nguyên dương  $n$   $m$   $k$  ( $2 \leq n \leq 100$ ;  $1 \leq m \leq n * (n - 1) / 2$ ;  $1 \leq k \leq 10$ ).
- $m$  dòng tiếp theo, mỗi dòng có dạng  $u$   $v$  cho biết giữa hai nút giao thông  $u$  và  $v$  có một đường phố nối chúng.

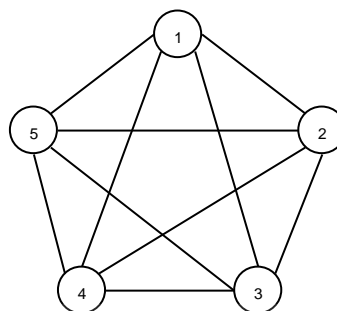
**Kết quả:** Ghi ra file văn bản SCHEDULE.OUT

- Dòng 1: Ghi số ngày tối thiểu cần để thực hiện dự án sửa đường. Nếu không có phương án thì chỉ cần ghi số -1.
- Nếu có phương án xếp lịch,  $m$  dòng tiếp theo, mỗi dòng có dạng  $u$   $v$   $p$  cho biết sẽ phải tiến hành sửa chữa đoạn đường nối giữa nút  $u$  và nút  $v$  trong ngày thứ  $p$  của dự án. (Ngày khởi công dự án là ngày thứ 1).

Các số trên một dòng của Input / Output file được ghi cách nhau ít nhất một dấu cách.

Ví dụ:

SCHEDULE . INP	SCHEDULE . OUT
5 10 5	2
1 2	1 2 1
1 3	1 3 2
1 4	1 4 2
1 5	1 5 2
2 3	2 3 1
2 4	2 4 2
2 5	2 5 1
3 4	3 4 1
3 5	3 5 2
4 5	4 5 1



## 084. ĐI THI

Hàng năm, sau khi công bố kết quả vòng I kỳ thi quốc gia, Bộ Giáo dục và Đào tạo lại tổ chức thi tiếp vòng II. Khác với vòng I, tất cả các thí sinh đều phải tập trung tại Hà Nội để tham dự kỳ thi diễn ra trong k ngày.

Bản đồ Hà Nội có n nút giao thông và m đường phố hai chiều. Giữa hai nút giao thông bất kỳ có nhiều nhất một đường phố nối chúng. Khách sạn (nơi ở của các thí sinh) nằm ở nút giao thông 1 và địa điểm thi nằm ở nút giao thông n.

Những học sinh ở xa tới Hà Nội muốn kết hợp đi thăm các đường phố của thủ đô. Với bản đồ Hà Nội trong tay và kỹ thuật lập trình siêu đẳng, các bạn thường vạch kế hoạch đi và về trong k ngày thi, mà ngoại trừ nút 1 và nút n, không đi qua nút giao thông nào khác quá một lần.

**Yêu cầu:** Có nhiều khả năng bạn sẽ là người tham dự kỳ thi vòng II, hãy cho biết kế hoạch đi thi và quay trở về khách sạn của bạn trong cả k ngày thi.

**Dữ liệu:** Vào từ file văn bản THI.INP

- Dòng 1: Ghi ba số nguyên dương n, m, k ( $2 \leq n \leq 100$ ;  $1 \leq k \leq 40$ )
- m dòng tiếp theo, mỗi dòng có dạng u v cho biết giữa hai nút giao thông u và v có một đường phố nối chúng.

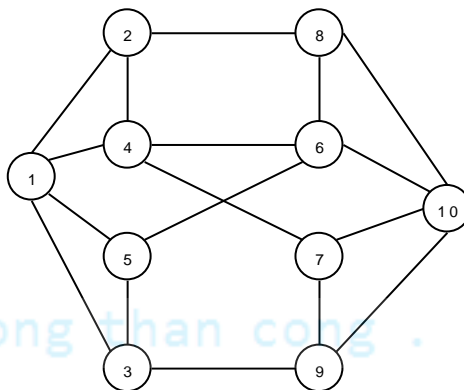
**Kết quả:** Ghi ra file văn bản THI.OUT

- Dòng đầu tiên: Ghi từ YES hay NO tùy theo có phương án đi thực hiện yêu cầu đề ra hay không
- Trong trường hợp có phương án đi về, 2k dòng tiếp theo cứ hai dòng ghi hành trình trong một ngày thi.
  - ♦ Dòng 1: Ghi danh sách các nút giao thông đi qua khi đi thi, theo đúng thứ tự: bắt đầu là nút 1, tiếp theo là các nút trung gian, kết thúc là nút n.
  - ♦ Dòng 2: Ghi danh sách các nút giao thông đi qua khi trở về, theo đúng thứ tự: bắt đầu là nút n, tiếp theo là các nút trung gian, kết thúc là nút 1.

**Các số trên một dòng của Input / Output file được ghi cách nhau ít nhất một dấu cách.**

**Ví dụ:**

THI . INP	THI . OUT
10 17 2	YES
1 2	1 2 8 10
1 3	10 9 3 1
1 4	1 4 7 10
1 5	10 6 5 1
2 4	
2 8	
3 5	
3 9	
4 6	
4 7	
5 6	
6 8	
6 10	
7 9	
7 10	
8 10	
9 10	





## 085. MÈO KIỂU ÚC

Gia súc ở một khu làng ven rừng hay bị một con hổ dữ đến bắt trộm, dân làng đã thử đào hố để bẫy nhưng vô hiệu bởi hổ có chỉ số IQ trên mức trung bình. Sau một thời gian bị hổ quấy phá, người ta phát hiện ra rằng con hổ này rất khoái Tiger Beer!!!, và khi hổ uống bia lúc đói thì rất có thể sẽ bị say và rơi xuống hố.

**Dân làng muốn tìm cách đặt các thùng bia và đào hố để trên mọi đường đi từ rừng tới làng, hổ sẽ phải gặp ít nhất một thùng bia và SAU ĐÓ gặp ít nhất một cái hố. Vì chi phí đào một cái hố lớn hơn rất nhiều so với tiền mua một thùng bia, nên dân làng muốn chọn trong các phương án phải đào ít hố nhất ra phương án phải dùng ít thùng bia nhất.**

Bản đồ được đánh dấu bằng n địa điểm mà tại những địa điểm này không thể đào hố cũng như đặt bia, rừng là địa điểm 1 và làng ở địa điểm n. Giữa hai địa điểm bất kỳ có nhiều nhất là một đường mòn nối chúng và trên đoạn đường đó chỉ được đào một hố hoặc chỉ đặt một thùng bia hoặc không đào hố cũng chẳng đặt bia.

**Dữ liệu:** Vào từ file văn bản TIGER.INP

- Dòng 1: Chứa số n ( $2 \leq n \leq 100$ )
- Các dòng tiếp, mỗi dòng ghi hai số nguyên dương u, v cho biết giữa hai địa điểm u và v có một đường mòn nối chúng.

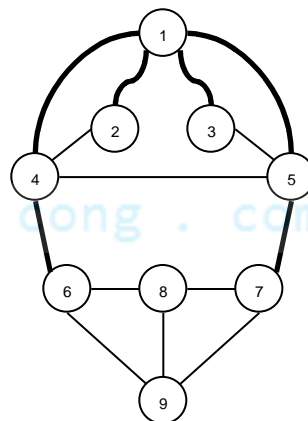
**Kết quả:** Ghi ra file văn bản TIGER.OUT

- Dòng 1: Ghi từ YES nếu có phương án ngăn không cho hổ vào làng, ghi NO trong trường hợp ngược lại
- Trong trường hợp có phương án:
  - ♦ Dòng tiếp theo ghi số hố phải đào (P)
  - ♦ P dòng tiếp theo, mỗi dòng ghi một cặp số x y cho biết phải đào hố trên con đường nối địa điểm x với địa điểm y.
  - ♦ Dòng tiếp theo ghi số thùng bia phải đặt (Q)
  - ♦ Q dòng tiếp theo, mỗi dòng ghi một cặp số z t cho biết phải đặt thùng bia trên con đường nối địa điểm z với địa điểm t.

**Các số trên một dòng của Input/Output file được ghi cách nhau ít nhất một dấu cách.**

**Ví dụ:**

TIGER.INP	TIGER.OUT
9	YES
1 2	2
1 3	4 6
1 4	5 7
1 5	4
2 4	2 1
3 5	3 1
4 5	4 1
4 6	5 1
5 7	
6 8	
6 9	
7 8	
7 9	
8 9	



## 086. THÀNH PHỐ TRÊN SAO HOẢ

Đầu thế kỷ 21, người ta thành lập một dự án xây dựng một thành phố trên sao Hoả để thế kỷ 22 con người có thể sống và sinh hoạt ở đó. Giả sử rằng trong thế kỷ 22, phương tiện giao thông chủ yếu sẽ là các phương tiện giao thông công cộng nên để đi lại giữa hai địa điểm bất kỳ trong thành phố, người ta có thể yên tâm chọn đường đi ngắn nhất mà không sợ bị trễ giờ do kẹt xe. Khi mô hình thành phố được chuyển lên Internet, có rất nhiều ý kiến phản nản về tính hợp lý của nó, đặc biệt, tất cả các ý kiến đều cho rằng hệ thống đường phố như vậy là quá nhiều, làm tăng chi phí xây dựng cũng như bảo trì.

**Hãy bỏ đi một số đường trong dự án xây dựng thành phố, thoả mãn:**

- Nếu giữa hai địa điểm bất kỳ trong dự án ban đầu có ít nhất một đường đi thì sự sửa đổi này không làm ảnh hưởng tới độ dài đường đi ngắn nhất giữa hai địa điểm đó.
- Tổng độ dài những đường phố được giữ lại là ngắn tới thiểu.

**Dữ liệu:** Vào từ file văn bản CITY.INP, chứa bản đồ dự án.

- Dòng thứ nhất: Ghi số địa điểm  $n$  và số đường phố  $m$  (Giữa hai địa điểm bất kỳ có nhiều nhất là một đường phố nối chúng);  $n \leq 200$ ;  $0 \leq m \leq n * (n - 1) / 2$ .
- $m$  dòng tiếp theo, mỗi dòng ghi ba số nguyên dương  $u, v, c$ : cho biết có đường hai chiều nối giữa hai địa điểm  $u, v$  và độ dài con đường đó là  $c$ .

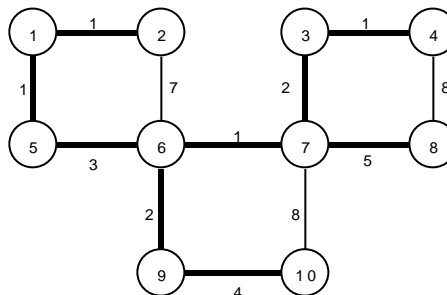
**Kết quả:** Ghi ra file văn bản CITY.OUT, chứa kết quả sau khi sửa đổi.

- Dòng thứ nhất ghi hai số  $k, d$ . Ở đây  $k$  là số đường phố còn lại còn  $d$  là tổng độ dài của các đường phố còn lại.
- $k$  dòng tiếp theo, mỗi dòng ghi hai số nguyên dương  $p, q$ : cho biết cần phải giữ lại con đường nối địa điểm  $p$  với địa điểm  $q$ .

**Các số trên một dòng của Input / Output File được ghi cách nhau ít nhất một dấu cách.**

**Ví dụ:**

CITY.INP	CITY.OUT
10 12	9 20
1 2 1	1 2
1 5 1	1 5
2 6 7	3 4
3 4 1	3 7
3 7 2	5 6
4 8 8	6 7
5 6 3	6 9
6 7 1	7 8
6 9 2	9 10
7 8 5	
7 10 8	
9 10 4	



## 087. RÔ BỐT XÂY NHÀ

Có một số con Rô-bốt xây nhà trên một mảnh đất hình vuông, mảnh đất đó được chia thành lưới ô vuông đơn vị kích thước  $n \times n$ . Vì Rô-bốt được lập trình xây nhà khá máy móc, nên hai ngôi nhà do cùng một con Rô-bốt xây nên sẽ có kích thước và hình dạng đáy giống hệt nhau (Có thể đặt chồng khít lên nhau qua một phép dời hình), hai ngôi nhà do hai con Rô-bốt khác nhau xây nên thì có ít nhất một ô khác nhau.

Khi công trình hoàn thành, các ngôi nhà được xây hoàn toàn tách biệt (không có hai ngôi nhà nào chung ô, chung tường, nhưng có thể chung góc tường). Bản đồ của khu đất đã được chụp ảnh và mã hoá dưới dạng một ma trận vuông  $A$  kích thước  $n \times n$ , trong đó  $a_{ij} = 1$  cho biết ô  $(i, j)$  của mảnh đất thuộc một ngôi nhà nào đó còn  $a_{ij} = 0$  cho biết ô  $(i, j)$  của mảnh đất vẫn còn để trống.

1	1	1	0	1	0	0	0	1
1	0	0	0	1	1	0	1	1
1	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0	0
0	0	1	1	0	1	0	0	0
1	0	0	0	0	1	0	0	1
1	0	1	0	0	1	1	0	1
1	1	1	1	1	0	0	1	1

**Vấn đề đặt ra là khi có bản đồ khu nhà trong tay, hãy xác định số con rô bốt tham gia xây nhà và chỉ rõ con rô bốt nào xây ngôi nhà nào.**

**Dữ liệu:** Vào từ file văn bản HOUSES.INP

- Dòng 1: Ghi số nguyên dương  $n$  ( $n \leq 100$ ).
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số, số thứ  $j$  là  $a_{ij}$

**Kết quả:** Ghi ra file văn bản HOUSES.OUT

- Dòng 1: Ghi số con rô-bốt tham gia xây nhà ( $k$ ).
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi  $n$  số, số thứ  $j$  là  $b_{ij}$ . Ở đây, nếu  $a_{ij} = 0$  thì  $b_{ij} = 0$ , nếu  $a_{ij} = 1$  thì  $b_{ij}$  là số hiệu con rô bốt xây ngôi nhà chứa ô  $(i, j)$ . Các con rô-bốt được đánh số từ 1 đến  $k$  theo thứ tự tùy thích.

**Các số trên một dòng của Input/Output file được ghi cách nhau ít nhất một dấu cách.**

**Ví dụ:**

HOUSES . INP	HOUSES . OUT
9	4
1 1 1 0 1 0 0 0 1	1 1 1 0 2 0 0 0 2
1 0 0 0 1 1 0 1 1	1 0 0 0 2 2 0 2 2
1 1 0 0 0 0 0 0 0	1 1 0 0 0 0 0 0 0
1 0 0 1 0 0 0 0 0	1 0 0 3 0 0 0 0 0
1 0 0 1 0 0 0 0 0	1 0 0 3 0 0 0 0 0
0 0 1 1 0 1 0 0 0	0 0 3 3 0 4 0 0 0
1 0 0 0 0 1 0 0 1	1 0 0 0 0 4 0 0 3
1 0 1 0 0 1 1 0 1	1 0 1 0 0 4 4 0 3
1 1 1 1 1 0 0 1 1	1 1 1 1 1 0 0 3 3

## 088. TƯ DUY KIỂU ÚC

Một phần mềm nhỏ đã được người phân tích thiết kế chia làm  $n$  công đoạn và giao cho hai lập trình viên thực hiện. Mỗi lập trình viên sẽ lần lượt viết các đoạn trình được giao một cách tuần tự, và tiến hành song song với lập trình viên còn lại. (Bởi phong cách lập trình này yêu cầu tuân thủ tuyệt đối thiết kế ban đầu, không được bắt người kia làm theo ý mình làm ảnh hưởng tới tiến độ).

Trong hai lập trình viên, có một người chuyên lập trình PASCAL và một người chuyên lập trình C++. Điều đó không gây khó khăn nhiều bởi họ sẽ dịch các đoạn trình dưới dạng các thư viện liên kết ngoài và sau đó chỉ cần lắp ráp lại là xong. Tuy nhiên, có thể có những công đoạn mà lập trình viên PASCAL viết nhanh hơn và cũng có thể có những công đoạn khác anh ta viết chậm hơn lập trình viên C++.

### ***Yêu cầu:***

***Cho biết thời gian dự kiến để lập trình viên PASCAL viết đoạn trình thứ  $i$  là  $p_i$  phút, thời gian dự kiến để lập trình viên C++ viết đoạn trình thứ  $j$  là  $c_j$  phút. Hãy phân mỗi công đoạn cho đúng một người viết để thời gian hoàn thành phần mềm là nhanh nhất.***

Ràng buộc:  $n, p_i, c_j$  ( $1 \leq i, j \leq n$ ) là các số nguyên dương không quá 100.

**Dữ liệu:** Vào từ file văn bản SOFTWARE.INP

- Dòng 1: Chứa số  $n$
- Các dòng tiếp theo, chứa các số từ  $p_1$  đến  $p_n$  rồi từ  $c_1$  đến  $c_n$  theo đúng thứ tự đó.

**Kết quả:** Ghi ra file văn bản SOFTWARE.OUT

- Dòng 1: Ghi thời gian cần để hoàn thành hết cả  $n$  công đoạn
- Dòng 2: Ghi số hiệu các công đoạn được giao cho lập trình viên PASCAL thực hiện
- Dòng 3: Ghi số hiệu các công đoạn được giao cho lập trình viên C++ thực hiện

***Các số trên một dòng của Input/Output file được ghi cách nhau ít nhất một dấu cách***

**Ví dụ:**

SOFTWARE . INP
6
10 100 30 50 50 80
100 30 40 40 60 90

SOFTWARE . OUT
130
1 3 6
2 4 5

cuu duong than cong . com

### 089. 8-3, TẶNG HOA KIỂU ÚC <sup>10test</sup>

Nhân ngày 8 - 3, một bạn nam trong lớp muốn tặng hoa cho một bạn nữ mà sở thích của bạn nữ này kỳ quặc đến mức chỉ có ... máy tính mới hiểu được. Chẳng hạn như bạn nữ này cho rằng trong bó hoa được tặng, đã có hoa hồng thì phải có hoa cúc, đã có hoa cúc thì phải có hoa phăng, mà đã có hoa phăng thì lại phải có ... hoa hồng. Và nếu như ai đem tặng cô ta một bó hoa không ưng ý thì thà không tặng còn hơn bởi hậu quả ra sao thì cũng chỉ có máy tính mới biết được.

#### ***Yêu cầu:***

***Hãy chọn một bó hoa gồm ít loại hoa nhất mà vẫn phù hợp với sở thích của bạn nữ khó tính đó.***

**Dữ liệu:** Vào từ file văn bản FLOWERS.INP

- Dòng 1: Ghi số n là số lượng các loại hoa ( $1 < n < 200$ )
- Các dòng tiếp theo, mỗi dòng ghi hai số u và v cho biết: Nếu đã tặng loại hoa u thì sẽ phải tặng luôn cả loại hoa v.

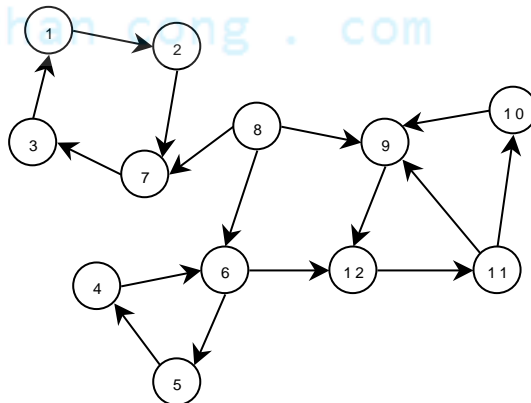
**Kết quả:** Ghi ra file văn bản FLOWERS.OUT

- Dòng 1: Ghi số nguyên dương k là số loại hoa chọn ra được
- Dòng 2: Ghi số hiệu của k loại hoa được chọn

***Các số trên một dòng của Input / Output file được ghi cách nhau ít nhất một dấu cách.***

**Ví dụ:**

FLOWERS . INP	FLOWERS . OUT
12	4
1 2	9 10 11 12
2 7	
3 1	
4 6	
5 4	
6 5	
6 12	
7 3	
8 6	
8 7	
8 9	
9 12	
10 9	
11 9	
11 10	
12 11	



**Về nhà:** Cho biết giá tiền mỗi loại hoa, hãy chọn một bó hoa rẻ tiền nhất!!!

## 090. MÃ HOÁ BURROWS WHEELER<sup>hard</sup>

Cho một từ  $W$  độ dài  $n$ , người ta có một cách mã hoá như sau: Ví dụ với từ BANANA.

Bước 1: Xét  $n$  hoán vị vòng quanh của  $W$ :

BANANA  
ANANAB  
NANABA  
ANABAN  
NABANA  
ABANAN

Bước 2: Sắp xếp  $n$  hoán vị vòng quanh đó theo thứ tự từ điển:

ABANAN  
ANABAN  
ANANAB  
BANANA (\*)  
NABANA  
NANABA

Bước 3:

Gọi  $k$  là vị trí của từ ban đầu trong dãy hoán vị vòng quanh sau khi đã sắp xếp (ở đây  $k$  là 4).

Lấy của mỗi hoán vị vòng quanh (theo đúng thứ tự sau khi đã sắp xếp theo thứ tự từ điển) một ký tự cuối và ghép thành một từ  $W'$  (ở đây  $W' = \text{'NNBAAA'}$ )

Ta gọi cặp  $(W', k)$  là mã công khai của từ  $W$ .

**Yêu cầu 1:**

Viết một chương trình đọc file văn bản *ENCODE.INP* gồm nhiều dòng, mỗi dòng chứa một từ. Tương ứng với mỗi từ  $W$  trên một dòng, hãy mã hoá và ghi vào file văn bản *ENCODE.OUT* hai dòng là mã công khai của từ đó: dòng 1 ghi từ  $W'$ , dòng 2 ghi số  $k$ .

**Yêu cầu 2:**

Viết một chương trình khác đọc file văn bản *DECODE.INP* gồm nhiều cặp dòng: Cứ hai dòng liên tiếp chứa một mã công khai: dòng 1 chứa từ  $W'$  và dòng 2 ghi số  $k$ . Tương ứng với mỗi cặp dòng đó, hãy giải mã và ghi vào file văn bản *DECODE.OUT* một dòng chứa từ  $W$  là từ đã giải mã ra được.

Hai yêu cầu trên phải được thực hiện độc lập trên hai file chương trình khác nhau.

Ràng buộc dữ liệu: Các từ được cho luôn khác rỗng, chỉ gồm các chữ cái in thường và có độ dài không quá 10000.

**Ví dụ:**

ENCODE . INP	ENCODE . OUT
qua	uaq
gi	2
ma	ig
to	1
to	ma
nhat	2
	to
	2
	to
	2
	hnta
	3

DECODE . INP	DECODE . OUT
xin	xin
3	thua
utah	rang
3	qua
rnag	dat
4	
uaq	
2	
dta	
2	

## 091. BAO LỒI

Trên mặt phẳng với hệ tọa độ Decartes vuông góc, cho  $n$  điểm không đồng thời thẳng hàng. Điểm thứ  $i$  có tọa độ là  $(x_i, y_i)$ .

(Số  $n$  và các tọa độ  $x_i, y_i$  đều là số nguyên:  $3 \leq n \leq 1000$ ;  $-300 \leq x_i \leq 300$ ;  $-200 \leq y_i \leq 200$ ).

Hãy tìm một đa giác lồi có diện tích nhỏ nhất mà miền đóng giới hạn bởi biên đa giác chứa tất cả những điểm đã cho.

**Dữ liệu:** Vào từ file văn bản BOUND.INP

- Dòng 1: Chứa số  $n$
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi hai số  $x_i, y_i$

**Kết quả:** Ghi ra file văn bản BOUND.OUT

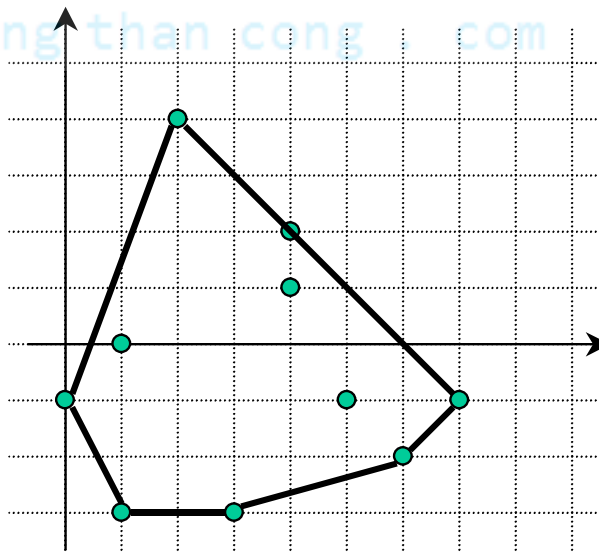
- Dòng 1: Ghi số  $m$  là số đỉnh của đa giác
- $m$  dòng tiếp theo, mỗi dòng ghi hai số nguyên theo thứ tự là hoành độ và tung độ của một đỉnh đa giác. Các đỉnh của đa giác không được phép có ba điểm thẳng hàng và chúng phải được liệt kê theo đúng thứ tự lập thành đa giác.

Vẽ hình mô tả kết quả tìm được trên màn hình đồ họa.

*Các số trên một dòng của Input / Output file ghi cách nhau ít nhất một dấu cách*

**Ví dụ:**

BOUND . INP	BOUND . OUT
10	6
0 -1	1 -3
1 0	3 -3
1 -3	6 -2
2 4	7 -1
3 -3	2 4
4 1	0 -1
4 2	
5 -1	
6 -2	
7 -1	



## 092. GIAI THỪA

Giai thừa của một số tự nhiên  $k$ , ký hiệu  $k!$  được định nghĩa quy nạp như sau:

- $0! = 1$
- $k! = (k - 1)! \cdot k \ (\forall k \geq 1)$

**Vấn đề đặt ra là cho trước hai số tự nhiên  $m, n$ . ( $1 \leq m \leq n \leq 10^6$ ). Hãy tìm hai số tự nhiên  $a$  và  $b$  để với mọi số tự nhiên  $k$  ( $[a, b]$  thì  $k!$  có không ít hơn  $m$  chữ số và không nhiều hơn  $n$  chữ số. Những số tự nhiên khác nằm ngoài đoạn  $[a, b]$  không có tính chất này.**

**Dữ liệu:** Vào từ file văn bản FDIGIT.INP gồm một dòng chứa hai số  $m, n$  cách nhau một dấu cách.

**Kết quả:** Ghi ra file văn bản FDIGIT.OUT gồm một dòng ghi hai số  $a, b$  cách nhau một dấu cách. Trong trường hợp không có số  $k$  nào thỏa mãn yêu cầu đề ra thì ghi hai giá trị bất kỳ  $a > b$ .

**Ví dụ:**

FDIGIT.INP	FDIGIT.OUT
2 4	4 7

FDIGIT.INP	FDIGIT.OUT
12 12	15 14

FDIGIT.INP	FDIGIT.OUT
3 9	5 12

cuu duong than cong . com

cuu duong than cong . com



## 093. PHỦ SÓNG

Dự kiến xây dựng mạng lưới phát thanh, truyền hình ở một địa phương nọ có một đài phát và n trạm tiếp sóng đánh số từ 1 tới n ( $n \leq 1000$ ). Trạm thứ i đã được xây dựng ở tọa độ  $(x_i, y_i)$ . (Các tọa độ là số thực,  $-10000 \leq x_i, y_i \leq 10000$ ). Để đảm bảo tính trung thực của các nguồn tin, các trạm tiếp sóng chỉ có thể nhận tín hiệu trực tiếp từ đài phát. Và như vậy có nghĩa là để phát sóng đến tất cả các trạm thu, bán kính phủ sóng của đài phát phải đủ lớn để phủ hết các trạm tiếp sóng. (Giả sử vùng phủ sóng là hình tròn có tâm là đài phát).

**Yêu cầu:**

**Hãy tìm vị trí đặt đài phát sao cho khoảng cách từ trạm xa nhất tới đài phát là ngắn nhất. Cho biết bán kính phủ sóng trong phương án tìm được tối thiểu phải là bao nhiêu.**

**Dữ liệu:** Vào từ file văn bản TELECOM.INP

- Dòng 1: Chứa số n
- n dòng tiếp theo, dòng thứ i chứa hai số  $x_i, y_i$  cách nhau ít nhất một dấu cách.

**Kết quả:** Ghi ra file văn bản TELECOM.OUT

Ghi ba số thực x, y, r. Ở đây (x, y) là tọa độ đặt đài phát và r là bán kính phủ sóng của đài phát (Đài phát có thể đặt trùng tọa độ với một trạm thu nào đó). Các số thực này phải được lấy tới 6 chữ số sau dấu chấm thập phân và phải ghi cách nhau ít nhất một dấu cách hoặc dấu xuống dòng

**Ví dụ**

TELECOM.INP	TELECOM.OUT
8	121.428571 135.714286
0 0	182.107840
200 300	
200 0	
200 200	
0 200	
100 300	
300 100	
100 0	

## 094. DÃY NGHỊCH THỂ

Cho  $x = (x_1, x_2, \dots, x_n)$  là một hoán vị của dãy số  $(1, 2, \dots, n)$ .

Dãy  $t = (t_1, t_2, \dots, t_n)$  được gọi là dãy nghịch thể của dãy hoán vị  $x$  nếu nó được xây dựng như sau:

$t_i :=$  số phần tử đứng trước giá trị  $i$  mà lớn hơn  $i$  trong dãy  $x$ . ( $1 \leq i \leq n$ ).

Ví dụ: Với  $n = 6$

- Dãy  $x = (3, 2, 1, 6, 4, 5)$  thì dãy nghịch thể của nó là  $(2, 1, 0, 1, 1, 0)$
- Dãy  $x = (1, 2, 3, 4, 5, 6)$  thì dãy nghịch thể của nó là  $(0, 0, 0, 0, 0, 0)$
- Dãy  $x = (6, 5, 4, 3, 2, 1)$  thì dãy nghịch thể của nó là  $(5, 4, 3, 2, 1, 0)$

**Vấn đề đặt ra là cho trước dãy  $t$ , hãy cho biết dãy hoán vị  $x$  nhận  $t$  làm dãy nghịch thể của nó.**

**Dữ liệu:** Vào từ file văn bản RECOVER.INP

- Dòng 1: Chứa số nguyên dương  $n$  ( $n \leq 5000$ ).
- Dòng 2: Chứa các số  $t_1, t_2, \dots, t_n$  theo đúng thứ tự đó cách nhau ít nhất một dấu cách.

**Kết quả:** Ghi ra file văn bản RECOVER.OUT

Chỉ gồm một dòng ghi các số  $x_1, x_2, \dots, x_n$  cách nhau ít nhất một dấu cách theo đúng thứ tự đó.

Dữ liệu vào được cho luôn luôn đúng đắn để có thể tìm ra nghiệm

**Ví dụ:**

RECOVER . INP	RECOVER . OUT
6	3 2 1 6 4 5
2 1 0 1 1 0	

cuu duong than cong . com

cuu duong than cong . com

## 095. MUA HÀNG

Một công ty muốn mua  $m$  máy tính. Sau khi lấy thông tin tại  $n$  cửa hàng ( $1 \leq n \leq 10000$ ), người ta biết được rằng cửa hàng thứ  $i$  có bán  $a_i$  máy tính và với giá mỗi máy tính là  $b_i$ . ( $a_i, b_i$  là những số nguyên dương:  $a_i \leq 100$ ;  $b_i \leq 2000$ ).

*Giả sử rằng các cửa hàng có đủ máy để bán cho công ty. Hãy tìm cách mua rẻ nhất.*

**Dữ liệu:** Vào từ file văn bản BUY.INP

- Dòng 1: Chứa hai số  $m, n$  cách nhau ít nhất một dấu cách.
- $n$  dòng tiếp theo, dòng thứ  $i$  chứa hai số  $a_i, b_i$  cách nhau ít nhất một dấu cách.

**Kết quả:** Ghi ra file văn bản BUY.OUT

- Dòng 1: Ghi tổng số tiền phải trả.
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi số máy tính mua ở cửa hàng thứ  $i$ .

**Ví dụ:**

BUY . INP	BUY . OUT
22 5	168
3 30	0
5 10	5
6 8	6
10 5	10
2 20	1

cuu duong than cong . com

cuu duong than cong . com

## 096. XÂU CON CHUNG DÀI NHẤT

Xâu ký tự X được gọi là xâu con của xâu ký tự Y nếu ta có thể xoá đi một số ký tự trong xâu Y để được xâu X.

*Cho biết hai xâu ký tự A và B, hãy tìm xâu ký tự C có độ dài lớn nhất và là con của cả A và B.*

**Dữ liệu:** Vào từ file văn bản STR.INP

- Dòng 1: chứa xâu A
- Dòng 2: chứa xâu B

**Kết quả:** Ghi ra file văn bản STR.OUT

- Chỉ gồm một dòng ghi xâu C tìm được

**Ví dụ:**

STR. INP
abc1def2ghi3
abcdefghi123

STR. OUT
abcdefghi3

cuu duong than cong . com

cuu duong than cong . com

## 097. DÃY CON NGẮN NHẤT

Cho số nguyên dương  $n \leq 1000$  và  $n$  số tự nhiên  $a_1, a_2, \dots, a_n$ . ( $\forall i: a_i \leq 10000$ ).

**Yêu cầu:**

**Cho số nguyên dương  $m \leq 10000$ , hãy cho biết một dãy con của dãy  $a$  có tổng bằng  $m$  chứa ít phần tử nhất.**

**Dữ liệu:** Vào từ file văn bản SUBSEQ.INP

- Dòng 1: Chứa hai số  $n, m$
- Dòng 2: Chứa  $n$  số  $a_1, a_2, \dots, a_n$  theo đúng thứ tự đó.

**Kết quả:** Ghi ra file văn bản SUBSEQ.OUT

- Dòng 1: Ghi số  $k$  là số phần tử của dãy con chọn ra được, nếu không tồn tại dãy con có tổng bằng  $m$  thì ghi số -1.
- Nếu có phương án chọn dãy con, thì dòng 2 ghi chỉ số của  $k$  phần tử được chọn (ghi theo thứ tự tùy thích).

**Các số trên một dòng của Input/Output file được ghi cách nhau ít nhất một dấu cách.**

**Ví dụ:**

SUBSEQ . INP
10 220
10 30 50 70 90 20 40 60 80 100

SUBSEQ . OUT
3
8 5 4

cuu duong than cong . com

cuu duong than cong . com

## 098. BIẾN ĐỔI DÃY SỐ

Cho dãy số nguyên dương  $a = (a_1, a_2, \dots, a_n)$  ( $1 \leq n \leq 100$ ; với  $\forall i: 1 \leq a_i \leq 100$ ). Xét hai loại phép biến đổi:

- Phép biến đổi  $+i$ : Tăng  $a_i$  lên 1 đơn vị
- Phép biến đổi  $-i$ : Giảm  $a_i$  đi 1 đơn vị.

**Yêu cầu:**

**Hãy tìm một cách sử dụng ít phép biến đổi nhất để biến dãy  $a$  trở thành dãy thoả mãn:**

$$1 \leq a_1 < a_2 < \dots < a_n \leq 100$$

**Dữ liệu:** Vào từ file văn bản SEQ.INP

- Dòng 1: Chứa số  $n$
- Dòng 2: Chứa  $n$  số  $a_1, a_2, \dots, a_n$  theo đúng thứ tự đó cách nhau ít nhất một dấu cách.

**Kết quả:** Ghi ra file văn bản SEQ.OUT

- Dòng 1: Ghi số  $m$  là số phép biến đổi tìm được
- $m$  dòng tiếp theo, mỗi dòng ghi một phép biến đổi

**Ví dụ:**

SEQ . INP	SEQ . OUT
5	8
4 1 6 7 4	+5
	+5
	+5
	+5
	+2
	-1
	-1
	-1

SEQ . INP	SEQ . OUT
4	7
98 99 100 96	+4
	+4
	+4
	+4
	-3
	-2
	-1

cuu duong than cong . com

## 099. GIÁ TRỊ NHỎ NHẤT

Một số nguyên dương  $x$  gọi là con của số nguyên dương  $y$  nếu ta có thể xoá bớt một số chữ số của  $y$  để được  $x$ .

**Cho hai số nguyên dương  $a$  và  $b$  hãy tìm số  $c$  nhận cả  $a$  và  $b$  là con, sao cho giá trị của  $c$  là lớn nhất có thể.**

Ràng buộc:  $1 \leq a, b \leq 10^{100}$ ;

**Dữ liệu:** Vào từ file văn bản NUMBER.INP

- Dòng thứ nhất chứa số  $a$
- Dòng thứ hai chứa số  $b$

**Kết quả:** Ghi ra file văn bản NUMBER.OUT

- Ghi ra trên một dòng số  $c$ .

**Ví dụ:**

NUMBER . INP	NUMBER . OUT
111999111 999111999	111999111999

NUMBER . INP	NUMBER . OUT
567812345678 123456781234	1234567812345678

cuu duong than cong . com

cuu duong than cong . com

## 100. NỐI DÂY

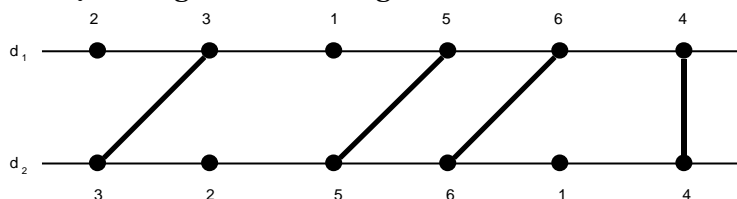
Cho hai đường thẳng song song nằm ngang  $d_1$  và  $d_2$ . Trên mỗi đường thẳng, người ta chọn lấy  $n$  điểm phân biệt và gán cho mỗi điểm một số nguyên dương là nhãn của điểm đó:

- Trên đường thẳng  $d_1$ , điểm thứ  $i$  (theo thứ tự từ trái qua phải) được gán nhãn là  $a_i$ .
- Trên đường thẳng  $d_2$ , điểm thứ  $j$  (theo thứ tự từ trái qua phải) được gán nhãn là  $b_j$ .

Ở đây  $(a_1, a_2, \dots, a_n)$  và  $(b_1, b_2, \dots, b_n)$  là những hoán vị của dãy số  $(1, 2, \dots, n)$

**Yêu cầu:** Hãy chỉ ra một số tối đa các đoạn thẳng thỏa mãn:

- Mỗi đoạn thẳng phải nối hai điểm có cùng một nhãn: một điểm trên đường thẳng  $d_1$  và một điểm trên đường thẳng  $d_2$ .
- Các đoạn thẳng đôi một không có điểm chung



**Dữ liệu:** Vào từ file văn bản LINES.INP

- Dòng 1: Chứa số nguyên dương  $n \leq 5000$
- Dòng 2: Chứa  $n$  số của dãy hoán vị  $a_1, a_2, \dots, a_n$ .
- Dòng 3: Chứa  $n$  số của dãy hoán vị  $b_1, b_2, \dots, b_n$ .

**Kết quả:** Ghi ra file văn bản LINES.OUT

- Dòng 1: Ghi số  $k$  là số đoạn thẳng nối được.
- Dòng 2: Ghi  $k$  nhãn của các đoạn thẳng được chọn (nhãn của mỗi đoạn thẳng là nhãn của điểm đầu mút)

**Các số trên một dòng của Input / Output file ghi cách nhau ít nhất một dấu cách.**

**Ví dụ:**

LINES . INP	LINES . OUT
6	4
2 3 1 5 6 4	4 6 5 3
3 2 5 6 1 4	

LINES . INP	LINES . OUT
7	5
1 2 3 4 5 6 7	1 2 3 4 5
1 2 6 7 3 4 5	

Cách cho điểm: Chấm theo 10 Test, điểm tối đa cho mỗi Test là 1.

Đối với mỗi một Test:

- Nếu chương trình chạy gặp lỗi, hoặc ghi sai khuôn dạng Output, hoặc cho phương án nối dây không hợp lệ (có hai đoạn thẳng cắt nhau), hoặc chạy quá 10 giây: 0 điểm.
- Nếu không, điểm cho test đó sẽ là:  $(\text{Số dây nối tìm được} / \text{số dây nối của đáp án})^2$ .